ISSN 2007-9737

Adaptation of Number of Filters in the Convolution Layer of a Convolutional Neural Network Using the Fuzzy Gravitational Search Algorithm Method and Type-1 Fuzzy Logic

Yutzil Poma[1], Patricia Melin[2]

[1] Tijuana Institute of Technology,
Mexico

[2] Tijuana Institute of Technology,
Computer Science in the Graduate Division,
Mexico

yutpoma@hotmail.com, pmelin@tectijuana.mx

Abstract. This paper presents a model of the search for adaptation of parameters and the creation of the membership functions of various fuzzy systems created using the fuzzy gravitational algorithm (FGSA). These fuzzy systems were created to find the optimal number of filters to enter a convolutional neural network (CNN) with an architecture of two convolution layers, as well as two pooling layers respectively and a classification layer, which is responsible for recognizing images. With this model, the results obtained by optimizing this CNN with the FGSA algorithm and the adaptation of parameters using this same algorithm are compared to form the membership functions of fuzzy systems. Both methods and their results are comparing with each other.

Keywords. CNN, FGSA, number of filters, fuzzy logic, fuzzy systems, adaptation of parameters, ORL database, Feret database, MNIST database.

1 Introduction

Artificial intelligence is an area that studies the way in which computers learn "naturally", as well as human beings based on examples that in turn form knowledge that is transformed into experiences, which learn to identify objects, images or signals reaching the recognition of each of these, based on their learning of the events [1, 2].

Deep learning is part of automatic learning, which is one of its greatest advantages, unlike traditional learning, which has a finite capacity for learning. In addition, deep learning expands our learning "skills" and accesses a greater amount of data, therefore "processes" or experiences the information, which in turn gets better and more accurate recognition [3, 4].

In recent times the rise of convolutional neural networks [5, 6] has been a success when using them in different fields, such as artificial vision [7], medicine [8, 9, 10], sign language [11, 12], in language recognition [13, 14], audio recognition [15, 16, 17], as well as face recognition [18, 19, 20], among other fields.

The recognition of human faces in recent times has increased potentially this is due to the demand for security as well as the regulation of the application of technology in commercial matters law [21]. Currently we have a variety of forms and methods to extract the characteristics of the images, whether we use convolutional neural networks in which by creating a hybridization of methods or with the help of bio-inspired algorithms we can obtain highly satisfactory results, such as in [22, 23, 24] or in [25, 26], in which [28] an image segmentation metaheuristic is used to detect pollen grains in images which, with the help of the gray logo algorithm, a classification of the pollen species is reached.

In addition, they have been used for the classification of COVID-19 images in order to detect the disease based on chest x-rays as well as lung tomography [29].

A widely used optimization method that has shown notable positive results is the Gravitational

Table 1. Rules used of fuzzy system

		Rules				
1	If	E is -REC	then	NF-1 is -REC	and	NF-2 is -REC
2	If	E is 1/2REC	then	NF-1 is 1/2REC	and	NF-2 is 1/2REC
3	If	E is +REC	then	NF-1 is +REC	and	NF-2 is +REC

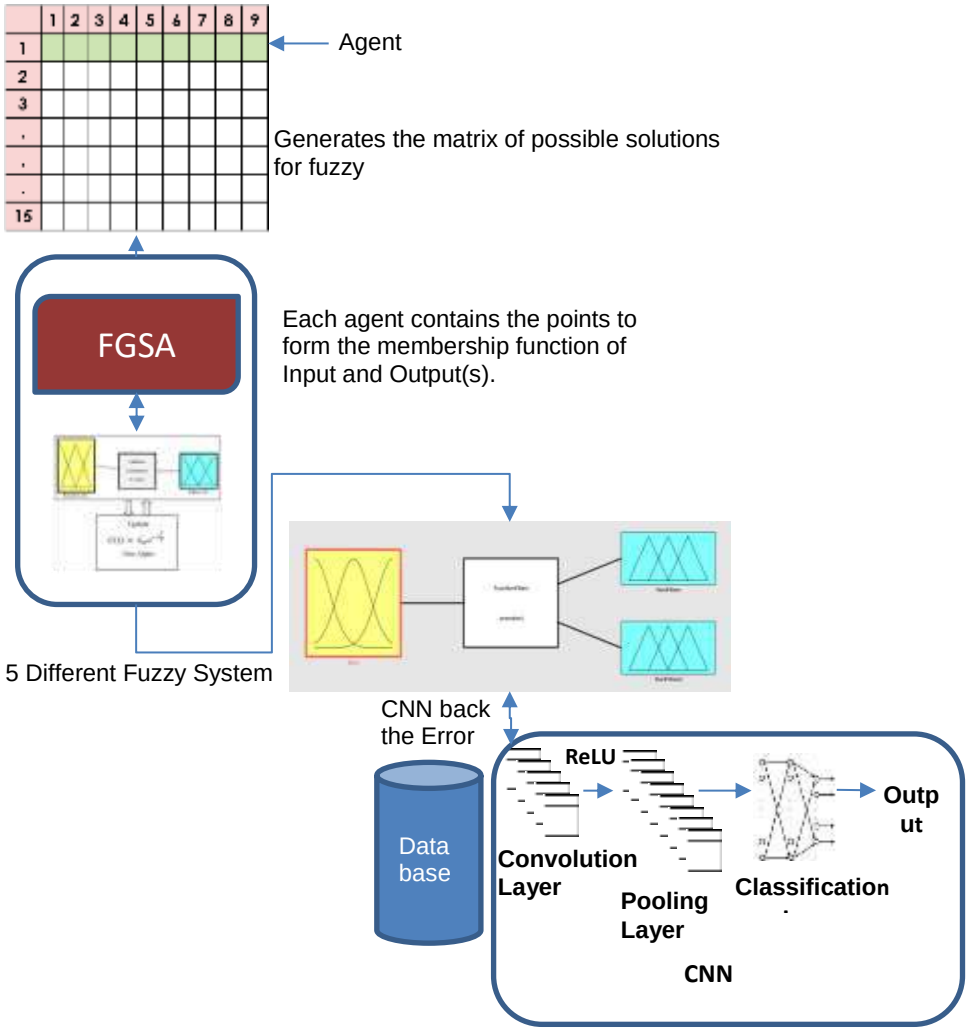


Fig. 1. General diagram for the adaptation of parameters using the FGSA method

Search algorithm which has been based on the law of gravity as well as the interactions of mass. This algorithm is based on search agents, which are a collection of masses that interact with each other based on gravity and the laws of motion proposed by Newton [30].

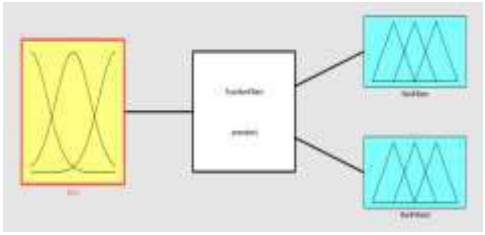


Fig. 2. Proposed Fuzzy system

Fig. 3. ORL Database examples

Table 2. Detail of experiment 1

Concept	Description
Number of Function of Membership	INPUT (3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Triangular Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filters of convolution layer 1): Triangular Membership Function	Static Points Ranges: -Rec: 0-0.5 ½ Rec: 0.25-0.75 +Rec: 0.5-1
Output 2 (Number of filters of convolution layer 2): Triangular Membership Function	Dynamic: Points generated by FGSA

A direct descendant of the GSA is the Fuzzy gravitational search algorithm (FGSA), which has been based on the same architecture of its predecessor with difference that the Alpha parameter is adaptive, thanks to the use of a type 1 fuzzy system, therefore the Acceleration and gravity are modified for each agent.

As in [31] where this method has been used in a modular neural network applied to echocardiogram recognition. Another outstanding work of this method is the adaptation of parameters dynamically using interval Type-2 fuzzy system presented in [32]. Some of the works where optimization metaheuristics have been applied are

Table 3. Results of experiment 1

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
10	93.12	25	27	90.35	1.08
20	94.37	25	30	91.66	1.12
30	95	25	26	92.25	1.15
40	94.37	50	50	92.37	1.00
50	95	50	50	92.52	1.15
60	96.25	25	21	92.83	1.10
70	**95.62**	**25**	**24**	**93.14**	**1.14**
80	95.62	50	50	93.02	1.26

Table 4. Detail of experiment 2

Concept	Description
Number of Function of Membership	INPUT (3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Triangular Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filters of convolution layer 1): Triangular Membership Function	Static Points Ranges: -Rec: 0-0.5 ½ Rec: 0.25-0.75 +Rec: 0.5-1
Output 2 (Number of filters of convolution layer 2): Triangular Membership Function	Static Points Ranges: -Rec: 0-0.5 ½ Rec: 0.25-0.75 +Rec: 0.5-1

Table 5. Results of experiment 2

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
40	94.37	25	25	92.62	1.13
50	95	25	25	92.58	1.06
60	95.62	25	25	92.66	1.21
70	**95.62**	**25**	**25**	**93**	**1.03**
80	95	25	25	92.60	1.10

[33] where the adaptation of parameters is used dynamically in bee colony optimization in which it is applied to control.

Some of the most used or databases in which different methodologies have been experimented are: FEI dataset [34], frontalized labeled faces in the wild (F_LFW) [35], GTAV face dataset [36], ORL dataset [37], Georgia Tech face [38], labeled faces in the wild (LFW) [39], GTAV face dataset [40], YouTube face dataset [41] Feret Database [42], and MNIST database [43].

The main contribution of this work is to find the best number of filters for each convolutional layer of the convolutional neural network, which, with the optimal number, will obtain better results in image recognition.

We have employed a combination of Type-1 fuzzy logic in conjunction with the FGSA method to find the best solutions as opposed to using only a CNN.

The content of the article in question is composed of the following form: in Section 2 we present the proposed method, in Section 3 we present the results obtained with the different case studies that we have used (ORL, FERET and MNIST Databases), in the Section 4 we find the conclusions and future work.

2 Proposed Method

The proposed method is the search for parameters for the number of filters (NF) of the convolution layers 1 and 2, respectively, of the convolutional neural network, which has the following architecture:

Conv1 (Number of filters) →ReLU→Pool1→Conv2 (Number of filters) →ReLU→Pool2→Clasif.

We propose 5 different fuzzy systems, which with the help of the Fuzzy gravitational search algorithm (FGSA) [44] we find the points of the membership functions, and these can vary their shape from triangular or Gaussian and also the points of the membership functions can be static or dynamic.

Various combinations of form and dynamic or static modification of the points of the membership functions were carry out and the results of each experiment were compared, replicating the two best ones of this methodology in other study cases.

In addition, the CNN was optimized to find the best number of filters with the same method (FGSA) and the results were compared using the two methodologies.

In Figure 1, we can find the general diagram for the adaptation of parameters using the FGSA method, it begins by means of this algorithm and its agents that will be the points that will form each membership function of the fuzzy system (this can be Gaussian or triangular).

Table 6. Detail of experiment 3

Concept	Description
Number of Function of Membership	INPUT (3), OUTPUT (3)
Input	Error (It is given by CNN)
Input: Gaussian Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filters of convolution layer 1): Triangular Membership Function	Dynamic: Points generated by FGSA
Output 2 (Number of filters of convolution layer 2): Triangular Membership Function	Dynamic: Points generated by FGSA

Table 7. Results of experiment 3

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
10	93.12	31	24	90.72	1.25
20	94.37	27	28	92.16	1.03
30	95	24	22	92.16	1.13
40	94.37	23	27	92.47	1.20
50	95	27	27	92.75	1.09
60	95	25	24	92.97	0.99
70	95.62	22	24	93.06	1.12
80	95	28	30	93.10	1.23

Table 8. Details of experiment 4

Concept	Description
Number of Function of Membership	INPUT (3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Gaussian Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filters of convolution layer 1): Triangular Membership Function	Static Points Ranges: -Rec: 0-0.5 ½ Rec: 0.25-0.75 +Rec: 0.5-1
Output 2 (Number of filters of convolution layer 2): Triangular Membership Function	Dynamic: Points generated by FGSA

Table 9. Results of experiment 4

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
40	95	25	24	92.37	1.11
50	95	26	23	92.60	1.18
60	**96.87**	**25**	**27**	**93.22**	**1.24**
70	95.62	27	22	92.79	1.38
80	95.62	25	29	93.22	0.99

Table 10. Details of experiment 5

Concept	Description
Number of Function of Membership	INPUT(3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Gaussian Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filter of convolution layer 1): Triangular Membership Function	Static Points Ranges: -Rec: 0-0.5 ½ Rec: 0.25-0.75 +Rec: 0.5-1
Output 2 (Number of filter of convolution layer 2): Triangular Membership Function	Static Points Ranges: -Rec: 0-0.5 ½ Rec: 0.25-0.75 +Rec: 0.5-1

Table 11. Results of experiment 5

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
40	95	26	26	92.35	1.35
50	94.37	25	25	92.41	1.04
60	94.37	25	25	92.54	0.98
70	**95.62**	**25**	**25**	**92.79**	**1.25**
80	94.37	25	25	92.41	0.96

Table 12. Details of experiment 6

Concept	Description
Number of Function of Membership	INPUT(3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Gaussian Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filter of convolution layer 1): Gaussian Membership Function	Dynamic: Points generated by FGSA
Output 2 (Number of filter of convolution layer 2): Gaussian Membership Function	Dynamic: Points generated by FGSA

Table 13. Results of experiment 6

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
40	95.62	14	15	93.12	0.95
50	95	15	14	93.27	0.90
60	96.87	32	31	93.52	1.31
70	**95.62**	**31**	**31**	**93.77**	**0.99**
80	95	28	16	93.29	1.01

Subsequently fuzzy systems are formed with the rules that we can observe in Table 1, where E=Error, NF1=Number of Filter 1 and NF2=Number of Filter 2.

Table 14. Details of experiment 7

Concept	Description
Number of Function of Membership	INPUT(3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Gaussian Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filter of convolution layer 1): Gaussian Membership Function	Static Points Ranges: -Rec: center=0.25, width= 0.11 ½ Rec: center=0.5, width= 0.11 +Rec: center=0.75, width= 0.11
Output 2 (Number of filter of convolution layer 2): Gaussian Membership Function	Dynamic: Points generated by FGSA

Table 15. Results of experiment 7

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
40	95	26	20	93	0.99
50	95.62	26	26	93.33	1.24
60	96.87	25	26	93.95	1.00
70	95.62	30	32	93.66	0.95
80	**96.25**	**26**	**27**	**94.02**	**0.89**

Table 16. Details of experiment 8

Concept	Description
Number of Function of Membership	INPUT(3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Gaussian Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filter of convolution layer 1): Gaussian Membership Function	Static Points Ranges: -Rec: center=0.25, width= 0.11 ½ Rec: center=0.5, width= 0.11 +Rec: center=0.75, width= 0.11
Output 2 (Number of filter of convolution layer 2): Gaussian Membership Function	Static Points Ranges: -Rec: center=0.25, width= 0.11 ½ Rec: center=0.5, width= 0.11 +Rec: center=0.75, width= 0.11

The CNN initially has an error of zero, the first time this network is executed, the error it obtains enters the fuzzy system that has previously formed both its shape (Gaussian or triangular) and its membership functions (static or dynamic).

It performs the parameter adaptation, thus obtaining the number of filters in convolution layer 1 and the number of filters in convolution layer 2 of the proposed CNN architecture. In Figure 2, we can see the details of the fuzzy system, which is of Mamdani type.

It has 1 input that corresponds to the Error returned by CNN and 2 outputs which correspond to the number of filters of convolution layers 1 (NF1) and 2 (NF2) respectively of the neural network.

Table 17. Results of experiment 8

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
40	95.62	29	29	92.39	1.33
50	95.62	25	25	92.77	1.28
60	95.62	27	27	92.75	1.09
70	**95.62**	**28**	**28**	**92.85**	**1.23**
80	95	29	29	93.02	1.12

Table 18. Details of experiment 9

Concept	Description
Number of Function of Membership	INPUT(3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Triangular Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filter of convolution layer 1): Triangular Membership Function	Static Points Ranges: -Rec: 0-0.5 ½ Rec: 0.25-0.75 +Rec: 0.5-1
Output 2 (Number of filter of convolution layer 2): Gaussian Membership Function	Dynamic: Points generated by FGSA

Table 19. Results of experiment 9

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
40	95	25	29	92.56	0.91
50	95.62	25	16	92.87	1.08
60	95.62	25	18	92.66	1.02
70	**96.25**	**25**	**27**	**93.41**	**1.30**
80	95.62	25	22	93.20	1.13

Table 20. Details of experiment 10

Concept	Description
Number of Function of Membership	INPUT(3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Triangular Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filter of convolution layer 1): Triangular Membership Function	Static Points Ranges: -Rec: 0-0.5 ½ Rec: 0.25-0.75 +Rec: 0.5-1
Output 2 (Number of filter of convolution layer 2): Gaussian Membership Function	Static Points Ranges: -Rec: center=0.25, width= 0.11 ½ Rec: center=0.5, width= 0.11 +Rec: center=0.75, width= 0.11

Table 21. Results of experiment 10

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
40	95	25	26	91.95	1.14
50	94.37	25	26	92.56	0.90
60	95.62	25	26	92.20	1.24
70	95	25	26	92.20	1.21
80	**94.37**	**50**	**50**	**92.70**	**0.88**

Table 22. Details of experiment 11

Concept	Description
Architecture	Conv1→ReLU→Pool1→Conv2→ReLU→Pool2→Clasif
Number of Function of Membership	INPUT (3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Triangular Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filter of convolution layer 1): Triangular Membership Function	Dynamic: Points generated by FGSA
Output 2 (Number of filter of convolution layer 2): Gaussian Membership Function	Static Points Ranges: -Rec: center=0.25, width= 0.11 ½ Rec: center=0.5, width= 0.11 +Rec: center=0.75, width= 0.11

Table 23. Results of experiment 11

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
40	94.37	23	25	92.27	1.08
50	95.62	26	26	92.43	1.19
60	95	26	26	92.72	1.15
70	96.25	27	25	92.47	1.24
80	96.25	28	28	92.93	1.22

3. Results and Discussions

3.1 Case of Study of the ORL Database

The first case study where all the possible combinations were applied in the form of membership functions (triangular and Gaussian), where they can be static or dynamic is the ORL database, which has 400 images of human faces; 40 different humans with 10 images of different angles each make it up of. These images have a size of 112 * 92 pixels each with in a .pgm format, below in Figure 3, we can see some examples of this database. These images have a size of 112 * 92 pixels each with in a .pgm format.

Below in Figure 3, we can see some examples of this database.

In the experimentation carried out with this case study, 16 experiments were carried out, which varied the type of membership functions and the points that make them up from static or dynamic. The epochs (EP) are varying.

Table 24. Details of experiment 11

Concept	Description
Number of Function of Membership	INPUT(3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Triangular Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filter of convolution layer 1): Triangular Membership Function	Dynamic: Points generated by FGSA
Output 2 (Number of filter of convolution layer 2): Gaussian Membership Function	Dynamic: Points generated by FGSA

Table 25. Results of experiment 11

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
40	95.62	27	29	92.77	1.02
50	96.25	50	50	92.83	1.16
60	96.25	26	30	93.31	1.21
70	**96.25**	**25**	**18**	**93.04**	**1.14**
80	95	22	24	93.04	0.93

Table 26. Details of experiment 13

Concept	Description
Number of Function of Membership	INPUT(3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Gaussian Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filter of convolution layer 1): Triangular Membership Function	Dynamic: Points generated by FGSA
Output 2 (Number of filter of convolution layer 2): Gaussian Membership Function	Static Points Ranges: -Rec: center=0.25, width= 0.11 ½ Rec: center=0.5, width= 0.11 +Rec: center=0.75, width= 0.11

Table. 27. Results of experiment 13

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
30	93.75	27	27	91.72	1.08
40	95.62	26	26	92.58	1.27
50	95.62	25	26	92.54	0.97
60	**95**	**24**	**26**	**92.87**	**1.19**
70	95.62	27	27	92.68	1.11
80	94.37	27	26	92.64	0.84

3.1.1. Experiment 1

In Table 2, we can find the details of the parameters used in experiment 1.

30 experiments were performed for each time of each experiment, we can find the best results in Table 3 and are shown in bold in each table.

Table 28. Details of experiment 14

Concept	Description
Number of Function of Membership	INPUT(3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Gaussian Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filter of convolution layer 1): Triangular Membership Function	Dynamic: Points generated by FGSA
Output 2 (Number of filter of convolution layer 2): Gaussian Membership Function	Dynamic: Points generated by FGSA

Table 29. Results of experiment 14

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
10	93.12	28	18	90.79	1.02
20	95	28	20	92.56	1.21
30	95.62	25	26	93.08	1.28
40	**95.62**	**26**	**16**	**93.52**	**0.93**
50	96.25	26	18	93.29	1.10
60	95.62	24	16	93.41	1.02
70	95.62	26	14	93.33	0.88
80	95	24	27	93.33	0.98

3.1.2. Experiment 2

In Table 4, we can find the details of the Parameters used in experiment 2.

In Table 5, we can find the results obtained using triangular and static outputs at the same time the input.

3.1.3. Experiment 3

In experiment number 3 we can see that both the input Gaussian type membership functions and the fuzzy system outputs are dynamic and the latter are triangular, we can see these results in Table 7 and the detail of this experiment in Table 6.

3.1.4. Experiment 4

In Table 8 we can note that the fuzzy system has Gaussian-type membership functions as input and they are dynamic, while the outputs are triangular and one is static and the other dynamic, as well as in Table 9 we find the results obtained from this experiment.

3.1.5. Experiment 5

In experiment 5, the input membership functions is Gaussian and dynamic while the outputs are triangular and static, we can see in Table 10 the details of this experiment, while in Table 11 the results obtained are shown.

3.1.6. Experiment 6

In Table 12, we can find the details of the parameters of experiment number 6, which has a Gaussian type membership function and is dynamic, like the outputs. In Table 13 we can see the results of this experiment where the best average of the 30 experiments made for each epoch is 93.77 with 70 epochs of network training.

3.1.7. Experiment 7

In this experiment, we can see that the input membership functions are Gaussian and the points that form them are dynamic, also the outputs, although they are also Gaussian, the first is static while the second is dynamic.

In Table 14, we can find the details of this experiment while in Table 15 its results.

Table 30. Details of experiment 15

Concept	Description
Number of Function of Membership	INPUT(3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Gaussian Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filter of convolution layer 1): Triangular Membership Function	Static Points Ranges: -Rec: 0-0.5 ½ Rec: 0.25-0.75 +Rec: 0.5-1
Output 2 (Number of filter of convolution layer 2): Gaussian Membership Function	Dynamic: Points generated by FGSA

Table 31. Results of experiment 15

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
40	95	36	37	93.08	0.98
50	96.25	25	29	93.29	1.18
60	96.87	25	16	93.54	1.38
70	95.62	28	23	93.18	0.96
80	**95.62**	**25**	**27**	**93.54**	**1.12**

Table 32. Details of experiment 16

Concept	Description
Number of Function of Membership	INPUT(3) OUTPUT (3)
Input	Error (It is given by CNN)
Input: Gaussian Membership Function	Dynamic: Points generated by FGSA
Output 1 (Number of filter of convolution layer 1): Triangular Membership Function	Static Points Ranges: -Rec: 0-0.5 ½ Rec: 0.25-0.75 +Rec: 0.5-1
Output 2 (Number of filter of convolution layer 2): Gaussian Membership Function	Static Points Ranges: -Rec: center=0.25, width= 0.11 ½ Rec: center=0.5, width= 0.11 +Rec: center=0.75, width= 0.11

Table 33. Results of experiment 16

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
40	95	25	25	92.18	1.03
50	94.37	26	27	92.41	1.09
60	95.62	25	25	92.68	1.16
70	95	25	26	92.70	0.98
80	**95**	**25**	**26**	**92.83**	**0.99**

Table 34. Compilation of the methods of all the experiments carried out for the ORL database

Experiment	Input (Error)	Output 1(NF1)	Output 2 (NF2)
1	Triangular (Dynamic)	Triangular Membership Functions (Static)	Triangular Membership Functions (Dynamic)
2	Triangular (Dynamic)	Triangular Membership Functions (Static)	Triangular Membership Functions (Static)
3	Gaussian (Dynamic)	Triangular Membership Functions (Dynamic)	Triangular Membership Functions (Dynamic)
4	Gaussian (Dynamic)	Triangular Membership Functions (Static)	Triangular Membership Functions (Dynamic)
5	Gaussian (Dynamic)	Triangular Membership Functions (Static)	Triangular Membership Functions (Static)
6	Gaussian (Dynamic)	Gaussian Membership Functions (Dynamic)	Gaussian Membership Functions (Dynamic)
7	Gaussian (Dynamic)	Gaussian Membership Functions (Static)	Gaussian Membership Functions (Dynamic)
8	Gaussian (Dynamic)	Gaussian Membership Functions (Static)	Gaussian Membership Functions (Static)
9	Triangular (Dynamic)	Triangular Membership Functions (Static)	Gaussian Membership Functions (Dynamic)
10	Triangular (Dynamic)	Triangular Membership Functions (Static)	Gaussian Membership Functions (Static)
11	Triangular (Dynamic)	Triangular Membership Functions (Dynamic)	Gaussian Membership Functions (Static)
12	Triangular (Dynamic)	Triangular Membership Functions (Dynamic)	Gaussian Membership Functions (Dynamic)
13	Gaussian (Dynamic)	Triangular Membership Functions (Dynamic)	Gaussian Membership Functions (Static)
14	Gaussian (Dynamic)	Triangular Membership Functions (Dynamic)	Gaussian Membership Functions (Dynamic)
15	Gaussian (Dynamic)	Triangular Membership Functions (Static)	Gaussian Membership Functions (Dynamic)
16	Gaussian (Dynamic)	Triangular Membership Functions (Static)	Gaussian Membership Functions (Static)

3.1.8. Experiment 8

In Table 16, we can find the details of the parameters used for the fuzzy system, where the input membership functions are Gaussian and dynamic while the outputs are of the same type but static. We can find the results in Table 17.

3.1.9. Experiment 9

In experiment 9 in Table 18, we can see the details of the parameters used for the fuzzy system and in Table 19 we can find the results.

3.1.10. Experiment 10

In Table 20, we can find the details of the parameters used for experiment 10.

In Table 20, we can find the details of the applied fuzzy system and its characteristics that make it up for this experiment, where the input of the membership functions is triangular and dynamic and output 1 is triangular and static as well as the second output is of type Gaussian and it is also static.

With the realization of this experiment, it has been found that with 80 periods of network training, the best result has been obtained in the recognition

Table 35. Comparison of the best results of the experimentation vs the optimization of CNN

Experiment	EP	Recognition Rate %	NF1	NF2	$\bar{X}$ %	σ	30 Experiment for each epoch for training
0	70	97.5	15	10	94.43	1.23	10,15,20,30,40,50,60,70
1	70	95.62	25	24	93.14	1.14	10,20,30,40,50,60,70,80
2	70	95.62	25	25	93	1.03	40,50,60,70,80
3	80	95	28	30	93.10	1.23	10,20,30,40,50,60,70,80
4	60	96.87	25	27	93.22	1.24	40,50,60,70,80
5	70	95.62	25	25	92.79	1.25	40,50,60,70,80
6	70	95.62	31	31	93.77	0.99	40,50,60,70,80
7	80	96.25	26	27	94.02	0.89	40,50,60,70,80
8	70	95.62	28	28	92.85	1.23	40,50,60,70,80
9	70	96.25	25	27	93.41	1.30	40,50,60,70,80
10	80	94.37	50	50	92.70	0.88	40,50,60,70,80
11	80	96.25	28	28	92.93	1.22	40,50,60,70,80
12	70	96.25	25	18	93.04	1.14	40,50,60,70,80
13	60	95	24	26	92.87	1.19	30,40,50,60,70,80
14	40	95.62	26	16	93.52	0.93	10,20,30,40,50,60,70,80
15	80	95.62	25	27	93.54	1.12	40,50,60,70,80
16	80	95	25	26	92.83	0.99	40,50,60,70,80

Table 36. Comparison of all experiments from highest to lowest of rate recognition average

Experiment	EP	Recognition Rate %	NF1	NF2	$\bar{X}$ %	σ	30 Experiment for each epoch for training
0	**70**	**97.5**	**15**	**10**	**94.43**	**1.23**	**10,15,20,30,40,50,60,70**
7	**80**	**96.25**	**26**	**27**	**94.02**	**0.89**	**40,50,60,70,80**
6	**70**	**95.62**	**31**	**31**	**93.77**	**0.99**	**40,50,60,70 80**
15	80	95.62	25	27	93.54	1.12	40,50,60,70,80
14	40	95.62	26	16	93.52	0.93	10,20,30,40,50,60,70,80
9	70	96.25	25	27	93.41	1.30	40,50,60,70,80
4	60	96.87	25	27	93.22	1.24	40,50,60,70,80
1	70	95.62	25	24	93.14	1.14	10,20,30,40,50,60,70,80
3	80	95	28	30	93.10	1.23	10,20,30,40,50,60,70,80
12	70	96.25	25	18	93.04	1.14	40,50,60,70,80
2	70	95.62	25	25	93	1.03	40,50,60,70,80
11	80	96.25	28	28	92.9375	1.22	40,50,60,70,80
13	60	95	24	26	92.875	1.19	30,40,50,60,70,80
8	70	95.62	28	28	92.85	1.23	40,50,60,70,80
16	80	95	25	26	92.83	0.99	40,50,60,70,80
5	70	95.62	25	25	92.79	1.25	40,50,60,70,80
10	80	94.37	50	50	92.70	0.88	40,50,60,70,80

of the images of 92.70% in their average. In Table 21 we can find these results.

3.1.11. Experiment 11

In Table 22, we can see the detail for these experiment, where the input is triangular and dynamic, while the output 1 is triangular and dinamic while the output 2 is Gaussian membership function, it has static values.

In Table 23, we can find all the results obtained in the experiment 11, where the best results are when the neural networks are trained 80 epochs with results of 92.93% average recognition rate.

3.1.12. Experiment 12

In Table 24, we can find the details for this experiment, where the input is triangular and dynamic, while the output 1 is triangular and the

Table 37. Details and comparison of all the experiments realized

Place	Experiment	EP	Recognition Rate %	NF1	NF2	$\bar{X}$ %	σ	30 Experiment for each epoch for training	Structure of experiment
1	1	70	97.5	15	10	94.43	1.23	10,15,20,30,40,50,60,70	FGSA optimizer-C: Conv1 (Opt (15))→ReLU→Pool1→Conv2(Opt (10)) →ReLU→Pool2→Clasif.
2	9	80	96.25	26	27	94.02	0.89	40,50,60,70,80	Input: Gaussian Dynamic Output1: Gaussian Static Output2: Gaussian Dynamic
3	8	70	95.62	31	31	93.77	0.99	40,50,60,70,80	Input: Gaussian Dynamic Output1: Gaussian Dynamic Output2: Gaussian Dynamic

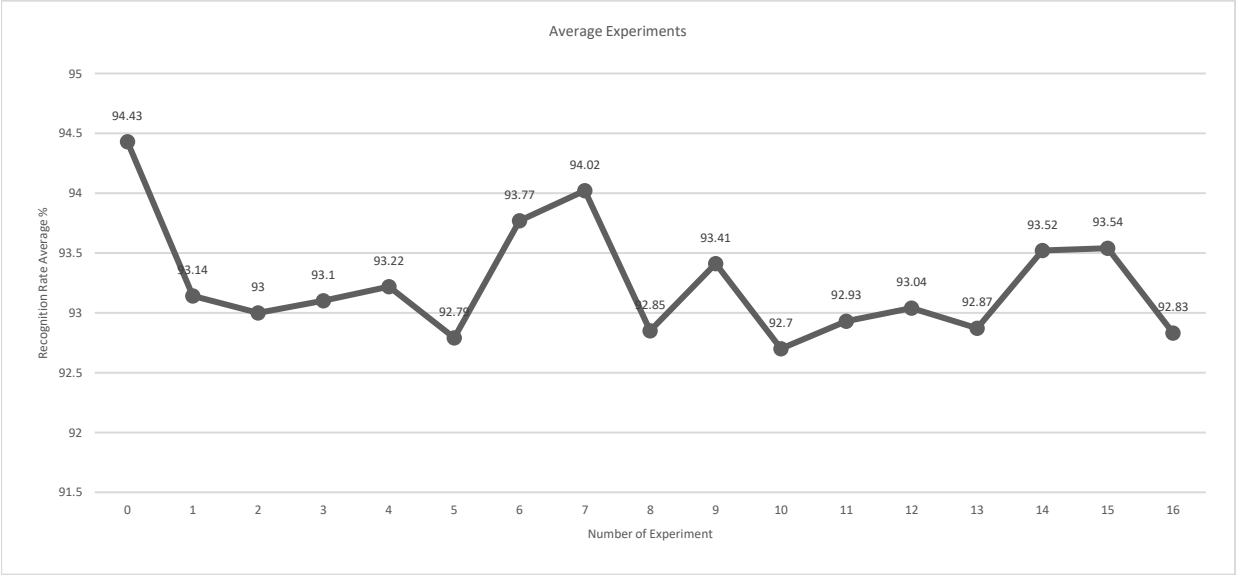


Fig. 4. Comparative graphic of recognition averages for the experiments

output 2 is Gaussian membership function, and both outputs have membership functions with dynamic values.

In Table 25, we can find all the results obtained in experiment 11, where the best results are when the neural network are trained 70 epochs with results of 93.04% average recognition rate.

3.1.13. Experiment 13

In this experiment, in Table 26, it can be seen that the fuzzy system has a Gaussian-type input, the membership function and its points are dynamic, while its different outputs, output 1 is triangular and dynamic, while the output 2 is of the Gaussian type and its points are static.

Table 27 shows the results obtained from the experiments carried out, where the 92.87% recognition average is the best value achieved for this experiment with a 60-epoch training of the convolutional neural network.

3.1.14. Experiment 14

In Table 28 we can find the details of experiment 14, where the input and its membership function is Gaussian type and its points are dynamic, while for output 1 the membership function is triangular and output 2 is Gaussian type, and both outputs are dynamic.

In Table 29 we can find the obtained results, it can been observed that the best result was 93.52% average in the recognition of the images, and these values were found training the network with 40 epochs.

Table 38. Comparison with other methods

*Preprocessing Method	Type of network	Optimization / Method	Integrator of response	Recognition rate (%) Max	Recognition rate (%)Max x̄	Data training	Data tester
IT1MGFLS [45]	Modular Neural Network (3 Modules)	No	Sugeno Integral	97.5	88.6	80%	20%
IT2MGFLS [45]	Modular Neural Network (3 Modules)	No	Sugeno Integral	93.75	85.98	80%	20%
IT1MGFLS [45]	Modular Neural Network (3 Modules)	No	Choquet Integral	97.5	92.59	80%	20%
IT2MGFLS [45]	Modular Neural Network (3 Modules	No	Choquet Integral	97.5	91.9	80%	20%
Gray and windowing method (4*4) [46]	Artificial neuronal network	No	Not apply	88.75	79.75	80%	20%
Gray and windowing method (8*8) [46]	Artificial neuronal network	No	Not apply	96.25	94.25	80%	20%
Not apply	**Convolutional Neural Network (70 EP)- Optimized with FGSA**	**FGSA**	**Not apply**	**97.5**	**94.43**	**60%**	**40%**
Not apply	**Convolutional Neural Network (80 EP) Optimized with Fuzzy Logic**	**Fuzzy Logic**	**Not apply**	**96.25**	**94.02**	**60%**	**40%**

3.1.15. Experiment 15

In this experiment, the input to the fuzzy system is Gaussian and dynamic, while output 1 is triangular and the points of the membership function are static. On the other hand, output 2 is Gaussian and the points are of dynamic type, and we can see in Table 30.

In Table 31, we can see that 93.54% is the highest average obtained, and this result was obtained by training the CNN with 80 epochs.

3.1.16. Experiment 16

This is the last experiment, in Table 32, that was carried out varying the membership functions of the fuzzy system from its type to its way of forming. The input was of Gaussian type and the points that form it are dynamic, output 1 is of triangular type, while output 2 is Gaussian type and both are static.

In the results obtained in Table 33 we can see that the best result was 92.83% on average in the recognition of the images, this result was achieved by training the neural network with 80 epochs.

Table 34 shows the compilation of all the best averages of all the experiments carried out, where the fuzzy system is applied with the variants where both the input and the outputs change their types (triangular and Gaussian) and the outputs of these membership functions can be static or dynamic.

In Table 35 we can see the comparison of the results obtained in the 16 experiments carried out with the experiment zero (0) that refers to the optimization of the filter numbers of the convolution layer 1 and the convolution layer 2 of the convolutional neural network.

In Table 36 we can observe the results of the 16 experiments carried out, these results have been ordered from the best average of the obtained recognition percentage to the lowest, also they have been compared with the results of the optimization of the convolutional neural network using the Fuzzy method Gravitational Search

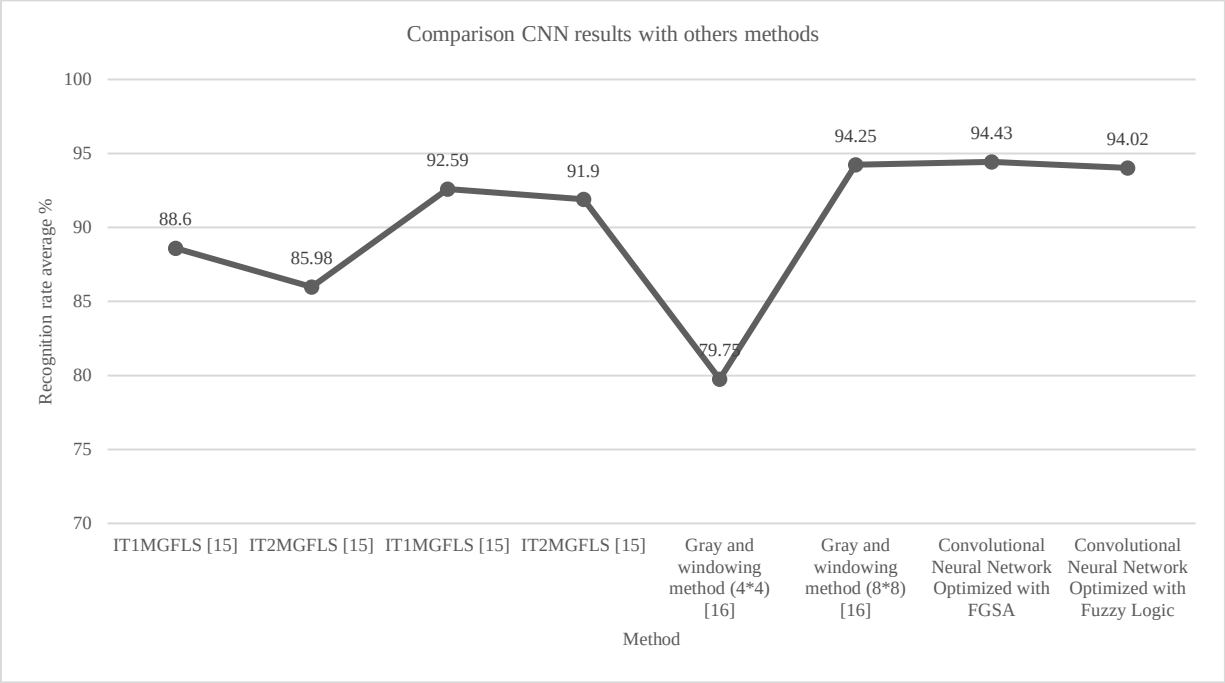


Fig. 5. Comparison with other methods

Algorithm to find the best number of filters and these results were compared with the best of the experimentation carried out, where a fuzzy system is used where both the inputs and the outputs and their membership functions are triangular or Gaussian and these can be static or dynamic.

Comparing the results we can decide that the best results are the experiment number zero (0) which is the optimization of the number of filters of the convolution layers 1 and 2 of the CNN and the second best result obtained (average) is for experiment 8 with a recognition percentage of 94.02% training the network at 80 epochs, using the fuzzy system where the input and the membership functions are Gaussian and dynamic, as well as the outputs that are also Gaussian but with the difference that the Output 1 points are static while Output 2 is not. The third best result of the experiments is experiment number 7, which obtained a recognition percentage of 93.77% with one input and two outputs of the Gaussian and dynamic type fuzzy system. We can see this comparison in detail in Table 37 and in Fig. 4.

In Table 38, we can see the comparison of the CNN optimization (best result obtained so far) with the adaptation of parameters using a fuzzy system where the membership functions change their type (triangular and Gaussian) and their points (dynamic or static).

We can see that although 60% of the images are used for training and 40% for tests and this percentage is much lower than what other methods use, better results are obtained than other works despite using a smaller percentage of images for training. It should be noted that a pre-processing is not being done to the images of the study database; in Fig. 5, we can find the comparison of the data.

Based on the experiments carried out in case study 1 with the ORL database, it was determined that the 3 best methodologies would be taken from all the experiments carried out and they were implemented in 2 more case studies to verify that metaheuristics can also be applied to other case studies.

3.2 Case of Study FERET Database

The FERET database is made up of 111,338 images of human faces, which consists of 994

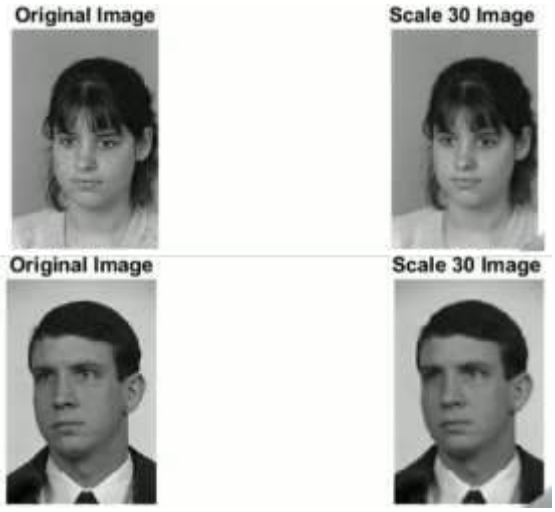


Fig. 6. Original image vs pre-processed image from the FERET database.

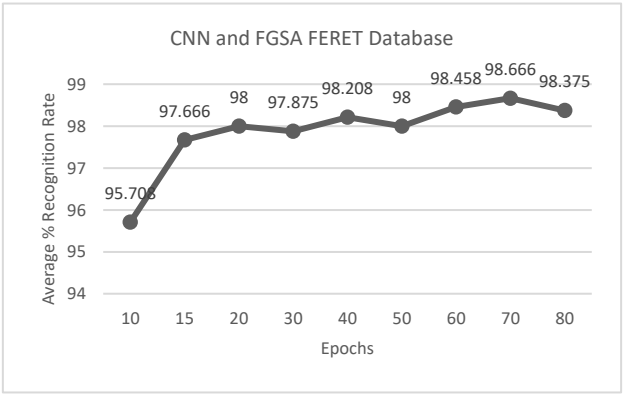


Fig. 7. Results of experiment # 0 represented graphically.

Table. 39. Results of first experiment (CNN optimized with FGSA)

EP	%RR	NF1	NF2	$\bar{X}$	σ
10	98.75	4	3	95.70	1.34
15	100	4	2	97.66	1.07
20	100	2	1	98	0.96
30	100	7	6	97.87	0.93
40	100	7	6	98.20	1.45
50	100	9	8	98	1.45
60	100	9	10	98.45	0.78
70	100	10	15	98.66	1.05
80	100	7	13	98.37	1.04

human faces taken from different angles, 200 images were used to train the neural network (10 images for each human), each image has a size of 256 * 384 pixels in their original size, but a preprocessing of each image was carried out to reduce its size, leaving a final size of 77 * 116 pixels for each image, each one of them is in a .jpg format.

Table 40. Results obtained using the methodology of experiment number 7 with the FERET database

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
40	**100**	**26**	**20**	**98.33**	**1.00**
50	100	25	17	98.33	1.05
60	100	30	32	98.25	1.01
70	100	26	30	98.20	0.90
80	100	25	18	98.12	0.97

Table 41. Results obtained using the methodology of experiment number 6 with the FERET database

EP	Recognition Rate	NF1	NF2	$\bar{X}$	σ
40	100	30	30	98.12	1.02
50	100	30	27	98.08	1.17
60	98.75	26	23	98	0.77
70	**100**	**32**	**20**	**98.20**	**0.84**
80	100	16	17	98	0.77

Table 42. Results comparative with experiments FERET database

Experiment	EP	Recognition Rate %	NF1	NF2	$\bar{X}$ %	σ	30 Experiment for each epoch for training	Structure of experiment
1	70	100	10	15	98.66	1.04	10,15,20,30,40,50,60,70,80	FGSA optimizer-C: Conv1 (Opt (10))→ReLU→Pool1→Conv2(Opt (15)) →ReLU→Pool2→Clasif.
9	40	100	26	20	98.33	1.00	40,50,60,70,80	Input: Gaussian Dynamic Output1: Gaussian Static Output2: Gaussian Dynamic
8	70	100	32	20	98.20	0.84	40,50,60,70,80	Input: Gaussian Dynamic Output1: Gaussian Dynamic Output2: Gaussian Dynamic

In the following Figure 6 we can see some examples of the original images vs the preprocessed image of the database used.

The first experiment carried out with this database was the optimization of the number of filters using the following architecture for CNN: Arq. Conv1 (Opt) → ReLU → Pool1 → Conv2 (Opt) → ReLU → Pool2 → Classification

Each experiment was performed 30 times using 15 agents and 3 dimensions in the FGSA, the neural network was trained 10,15, 20, 30, 40, 50, 60, 70 and 80 times, the results obtained show that the best result was 98.66% average in the recognition of the images.

The results can be seen in detail in Table 39, and the results are represented in Fig. 7.

The following test is experiment number 7, which is the second-best methodology in which the best results were obtained.

This experiment was carried out applying the proposed method, which, as already mentioned, consists of using the FGSA method for the creation of the membership functions, which are part of the fuzzy system created to find the parameters of the filter numbers of each convolution layer of the neural network.

The membership functions of the input are Gaussian and their values are dynamic, while output 1 has Gaussian and the values are static, unlike output 2, which are Gaussian, but the values are dynamic.

Each output represents the number of filters in each convolution layer of the CNN.

This experiment was carried out 30 times for each training season, the network was trained 40, 50, 60, 70 and 80 times, obtaining the best percentage of recognition of 98.33% with 40 training periods. The results can be seen in Table 40.

The third methodology with the best results was experiment #6, which uses membership functions of the input: Gaussian and Dynamic. In the same way, the 2 outputs of the fuzzy system are Gaussian and dynamic. This experiment was

Table 43. Results comparative with experiments FERET database with other methods and neural networks

Preprocessing Method	Type of network	Optimization	Integrator of response	Average recognition rate (%)	Recognition rate (%)	σ	Data training	Data tester
T1 FSs [47]	Monolithic neural network.	No	Sobel +T1 FSs	82.77	83.78	0.68	80%	20%
IT2 FSs [47]	monolithic neural network	No	Sobel + IT2 FSs	84.46	87.84	0.32	80%	20%
GT2 FSs [47]	monolithic neural network	No	Sobel + GT2 FSs	87.50	92.50	0.08	80%	20%
Not Applied	Canonical Correla3tion Analysis (CCA) [48]	No	Not applied	-	40%	-	-	
Not Applied	Linear Discriminant Analysis (LDA) [48]	No	Not applied	-	95%	-	-	
Viola-Jones algorithm, Resize 100*100 [49]	MNN	Grey Wolf Optimizer	-	92.63%	98%	4.05	up to 80%	20%
Not applied	**Convolutional Neural Network (70 EP) Optimized with FGSA**	**FGSA**	**Not applied**	**98.66**	**100%**	**1.04**	**60%**	**40%**
Not applied	**Convolutional Neural Network (40 EP) Optimized with Fuzzy Logic**	**Fuzzy Logic**	**Not applied**	**98.33**	**100%**	**1.00**	**60%**	**40%**

performed 30 times for each training season and the CNN was trained 40, 50, 60, 70 and 80 times, obtaining an average 98.20% as the best recognition result and 100% with its maximum percentage in the recognition of the images from the FERET database when the network is trained 70 times.

In Table 41, we can find the obtained results.

In Table 42, we have the compilation of the best results obtained, in which we can verify that the best result in the recognition of the images is when the CNN is optimized, following the proposed method with the methodology of experiment # 7 and finally in third place was experiment # 6.

The results obtained were compared with different methodologies, where we can see that the results obtained, both optimizing the network and introducing the proposed method, obtain better results than the rest of the compared methods, we can observe in Table 43.

3.3 Case of Study MNIST database

The third case study selected to use the proposed methodologies is the MNIST database, which consists of handwritten numbers, it consists of 60,000 images and has a set of 10,000 images for testing, we use the set of test images of 10,000 images which each image has a size of 28 * 28 pixels in black and white, 60% of images were used for training and 40% for tests.

For this case study we use 2 different architectures for the convolutional neural network, which are the following:

1- Arq. Conv1 (Opt) → ReLU → Pool1 → Conv2 (Opt) → ReLU → Pool2 → Classif.

Fig. 8. Examples of MNIST database

Table 44. Results obtained with the experiment #1 using the Architecture #1

EP	%RR	NF1	NF2	$\bar{X}$	σ
10	84.21	1	9	81.51	1.25
15	84.63	15	10	83.16	0.86
20	85.93	8	14	82.89	0.72
30	86.04	8	12	84.61	0.76
40	87.62	11	12	85.15	0.90
50	86.94	7	10	85.24	0.79
60	87.11	12	7	85.10	1.00
70	87.54	7	6	85.12	0.95
80	86.86	6	5	84.92	0.81
90	86.91	14	2	85.18	0.77
100	88.08	2	11	85.29	0.93
150	87.48	10	10	85.41	0.89
200	86.78	12	7	85.44	0.73
300	86.39	6	5	85.20	0.61
400	86.68	9	14	85.22	0.69
500	86.88	97	83	85.73	0.71
700	86.87	40	42	85.73	0.62
1000	88.04	46	40	85.82	0.77
3000	84.21	57	47	85.88	0.67
6000	88.84	19	17	86.14	0.92
10000	**87.47**	**57**	**73**	**86.19**	**0.63**

2- Arch. Conv1 (Opt) → ReLU → Pool1 → Conv2 (Opt) → ReLU → Pool2 → Conv3 (Opt) → ReLU → Pool3 → Classification

In the first architecture there are 2 convolution layers and two pooling layers, while in the second architecture 1 convolution layer was added, as well as a pooling layer to the architecture, thus deepening the neural network to improve the recognition percentage in the images.

In Figure 8 we can find some of the images that make up the MNIST database.

In experiment # 0, which deals with the optimization of the number of CNN filters, architecture # 1 was used, in which an exhaustive experimentation was carried out, carrying out 30 times each training period, until obtaining the best result for this architecture. In this case, the network was trained 10000 times resulting in an average

Table 45. Results obtained with experiment #7 using Architecture #1

EP	%RR	NF1	NF2	$\bar{X}$	σ
10000	**86.49**	**67**	**63**	**86.11**	**0.64**
6000	88.64	39	43	86.10	0.89
3000	86.68	44	47	85.89	0.63
1000	87.09	56	43	85.41	0.79

Table 46. Results obtained with experiment #6 using Architecture #1

EP	%RR	NF1	NF2	$\bar{X}$	σ
10000	86.88	45	73	83.19	0.63
6000	88.09	41	54	83.10	0.78
3000	86.76	15	87	82.78	0.63

Table 47. Results obtained with the experiment #7 using the Architecture #1

EP	Recognition Rate	NF1	NF2	NF3	$\bar{X}$	σ
100	91.33	61	66	18	89.72	0.60
500	91.29	61	66	18	89.80	0.59
1000	**91.18**	**61**	**66**	**18**	**89.82**	**0.57**
5000	90.70	10	4	33	89.78	0.54

Table 48. Comparative results for the MNIST database with architecture #1 using the experiment #0

Arq. Conv1 (Opt)→ReLU→Pool1→Conv2(Opt) →ReLU→Pool2→Clasif								
Experiment	**EP**	**%RR**	**NF1**	**NF2**	**$\bar{X}$ %**	**σ**	**30 Experiment for each epoch for training**	**Structure of experiment**
0	**10000**	**87.47**	**57**	**73**	**86.19**	**0.63**	**10000,6000,3000,1000,700,500,400,300,200, 150,100,90,80,70,60,50,40,30,20,15,10**	**FGSA optimizer-C: Conv1 (Opt (57))→ReLU→Pool1→Conv2(Opt (73)) →ReLU→Pool2→Clasif.**
7	10000	86.49	67	63	86.11	0.64	10000,6000,3000,1000	Input: Gaussian Dynamic Output1: Gaussian Static Output2: Gaussian Dynamic
6	10000	86.88	45	73	83.19	0.63	10000,6000,3000,1000	Input: Gaussian Dynamic Output1: Gaussian Dynamic Output2: Gaussian Dynamic

86.19% and a maximum of 87.47% in image recognition and the number of filters of each convolution layer was 57 and 73 respectively.

In Table 44 we can find a summary of the best results for each training season in which this case study was experimented.

The FGSA method with the following parameters were used for all experiments in which this method was used, 15 agents with 3 dimensions.

In experiment # 7, 30 iterations were carried out for each time and the network was trained 10,000, 6,000, 3,000 and 1,000 times, obtaining the best result when the CNN is trained with 10,000 times with a maximum of 86.49% and an average of 86.11. image recognition. In Table 45 we can find each result in detail.

Table 46 presents the results obtained in experiment # 6, which occupies the third place of the methodologies that yielded the best result based on the experimentation of case study # 1, this experiment consisted in training the neural network with 10000,6000 and 3000 epochs, 30 times for each case, and we observe that the maximum value is 86.88% and an average of 83.19% in the recognition of the images.

Based on previous experiments with the CNN architecture # 1, it was decided to deepen the network further, adding a convolution layer and a pooling layer to the architecture to find out if this new architecture could obtain better results.

Table 47 shows the results in which architecture # 2 was applied, where the network was trained 100, 500, 1000 and 5000 epochs each with 30 iterations respectively. In this experiment,

Table 49. The best results for the MNIST database with architecture #2 using the experiment #0

Arq. Conv1 (Opt)→ReLU→Pool1→Conv2(Opt) →ReLU→Pool2→ Conv3(Opt) →ReLU→Pool3→ Clasif								
EP	**Experiment Number**	**Recognition Rate**	**Number of agent**	**NF1**	**NF2**	**NF3**	$\bar{X}$	**σ**
1000	28	91.18	3	61	66	18	89.82	0.57

experiment # 0 was applied in which the network is optimized using the FGSA method, the results show that the best recognition value obtained was 91.18% and with an average of 89.82% in the recognition of the images with this case study with 1000 epochs training. The number of filters for the convolutional layer 3 is called NF3.

In Table 48, we can find the compilation of the best results obtained with architecture # 1 and the Table 49 the architecture # 2. Based on the previous experimentation, it was decided not to test the proposed method for architecture 2 since the increase in image recognition is minimal and it is more time and computing resource consuming.

4 Conclusions

As final conclusions of this experimentation, it has been observed that the optimization of the convolutional neural network with the number of filters using the FGSA method yields better results than with the proposed method using a fuzzy system, which is in charge of finding the best values for the parameters of the number of filters of each convolution of the network.

Another observation we have is that although the difference between the optimization and the proposed method is minimal, providing similar values, but nonetheless not better than using a bio-inspired algorithm to optimize the network as is the case of the Fuzzy Gravitational Search Algorithm method.

It was also observed that although the depth of the network is increased, better results are not always obtained and this depends on the database that is being used, the more complex it is, the deeper it is to extract more main characteristics; therefore, the simpler case study will be the CNN architecture and therefore the resources to use both in time and computing will also be less.

It was observed that the Gaussian-type membership functions produced better results than the triangular membership functions for the most part when these were dynamic rather than static.

As future work, more experimentation with the method will be carried out, modifying the depth of the convolutional neural network as well as making use of other more complicated databases to observe and analyze the data obtained. Based on the results obtained with type-1 fuzzy logic, it is intended as future work to implement type-2 fuzzy logic in the best architectures obtained in the work. It is expected to significantly improve the results obtained using this methodology and tested in new and more complicated databases.

References

1. **Russell, S.J. (2010).** Artificial intelligence: a modern approach. Upper Saddle River, N.J.: Prentice Hall.
2. **Roffel, S. (2020)** Introducing article numbering to Artificial Intelligence, Artificial Intelligence, Vol. 278, pp. 103210.
3. **Aggarwal, Ch.C. (2018).** Neural Networks and Deep Learning. Cham: Springer.
4. **Garain, A., Ray, B., Singh, P.K., Ahmadian, A., Senu, N., Sarkar, R. (2021).** GRA_Net: A deep learning model for classification of age and gender from facial images. IEEE Access 9: 85672–85689.
5. **Michelucci, U. (2019).** Advanced Applied Deep Learning: Convolutional Neural Networks and Object Detection. DOI: 10.1007/978-1-4842-4976-5.
6. **Li, J., Cao, F., Cheng, H., Qian, Y. (2021).** Learning the number of filters in convolutional neural networks. Int. J. Bio Inspired Comput., Vol. 17, No. 2, pp. 75–84.
7. **Browne, M., Ghidary, S.S. (2003).** Convolutional Neural Networks for Image Processing: An Application in Robot Vision. Australian Conference on Artificial Intelligence, pp. 641–652.

8. **Abdelrahman, L., Ghamdi, M.A., Collado-Mesa, F., Abdel-Mottaleb, M. (2021).** Convolutional neural networks for breast cancer detection in mammography: A survey. Computers in Biology and Medicine, Vol. 131, No. 104248.
9. **Lee, Y.W., Sheng Huang, C., Chung-Chih, S., Chang, R.F. (2021).** Axillary lymph node metastasis status prediction of early-stage breast cancer using convolutional neural networks. Computers in Biology and Medicine, Vol. 130.
10. **da Silva, B.C., Tam, G.R., Ferrari, R.J. (2021).** Detecting cells in intravital video microscopy using a deep convolutional neural network. Computers in Biology and Medicine, Vol. 129, No. 104133.
11. **Pias, P., Moh. Anwar-Ul-Azim Bhuiya, Md. Ayat Ullah, Molla Nazmus Saqib, Sifat-Momen, N.M. (2019).** A modern approach for sign language interpretation using convolutional neural network. **Nayak, A., Sharma, A. (eds)**. PRICAI´19: Trends in Artificial Intelligence. PRICAI´19. Lecture Notes in Computer Science, Vol. 11672.
12. **Ameen, S., Vadera, S. (2017).** A convolutional neural network to classify American sign language fingerspelling from depth and colour images. Expert Syst. J. Knowl. Eng. Vol. 34, No. 3.
13. **Gao, Y., Jia, C., Chen, H., Jiang, X. (2021).** Chinese fingerspelling sign language recognition using a nine-layer convolutional neural network. EAI Endorsed Trans. eLearn, Vol. 7, No. 20 e2.
14. **Ahuja, R., Jain, D., Sachdeva, D., Garg, A., Rajput, C. (2019).** Convolutional Neural Network Based American Sign Language Static Hand Gesture Recognition. Int. J. Ambient Comput. Intell. Vol. 10, No. 3, pp. 60–73.
15. **Cornejo, J.Y.R., Pedrini, H. (2019).** Bimodal emotion recognition based on audio and facial parts using deep convolutional neural networks. ICMLA, pp. 111–117.
16. **Han, Y., Kim, J.H., Lee, K. (2017).** Deep convolutional neural networks for predominant instrument recognition in polyphonic music. IEEE ACM Trans. Audio Speech Lang. Process, Vol. 25, No. 1, pp. 208–221.
17. **Abdel-Hamid, O., Mohamed, A.R., Jiang, H., Deng, L., Penn, G., Yu, D. (2014).** Convolutional neural networks for speech recognition. IEEE ACM Trans. Audio Speech Lang. Process., Vol. 22, No. 10, pp. 1533–1545.
18. **Liu, W., Zhou, L., Chen, J. (2021).** Face recognition based on lightweight convolutional neural networks. Inf. Vol. 12, No. 5, pp. 191.
19. **Rai, A.K., Senthilkumar, R., Kumar, A.R. (2020).** Combining pixel selection with covariance similarity approach in hyperspectral face recognition based on convolution neural network. Microprocess. Microsystems, Vol. 76, pp. 103096.
20. **Nasri, M.A., Hmani, M.A., Mtibaa, A., Petrovska-Delacrétaz, D., Slima, M.B., Hamida, A.B. (2020).** Face emotion recognition from static image based on convolution neural networks. ATSIP´20, pp. 1–6.
21. **Prasad, P.S., Pathak, R., Gunjan, V.K., Rao, H.V.R. (2019).** Deep learning based representation for face recognition. Springer Berlin, pp. 419–424.
22. **Mohammed, H.R., Hussain, Z.M. (2021).** Hybrid Mamdani fuzzy rules and convolutional neural networks for analysis and identification of animal images. Computation, Vol. 9, No. 3, pp. 35.
23. **Fregoso, J., González, C.I., Martínez, G.E. (2021).** Parameter optimization of a convolutional neural network using particle swarm optimization. Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms, pp. 149–169.
24. **Poma, Y., Melin, P. (2021).** Estimation of the number of filters in the convolution layers of a convolutional neural network using a fuzzy logic system. Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms, pp. 1–14.
25. **Rodriguez, R., González, C.I., Martinez, G.E., Melin, P. (2021).** An improved convolutional neural network based on a parameter modification of the convolution layer. Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms, pp. 125–147.
26. **Martín, A., Vargas, V.M., Gutiérrez, P.A., Camacho, D., Martínez. C.H. (2020).** Optimising convolutional neural networks using a hybrid statistically-driven coral reef optimisation algorithm. Appl. Soft Comput. Vol. 90, pp. 106144.
27. **Hekim, M., Cömert, O., Adem, K. (2020).** A hybrid model based on the convolutional neural network model and artificial bee colony or particle swarm optimization-based iterative thresholding for the detection of bruised apples. Turkish J. Electr. Eng. Comput. Sci., Vol. 28, No. 1, pp. 61–79.
28. **Menad, H., Ben-Naoum, F., Amine, A. (2020).** A hybrid grey wolves optimizer and convolutional neural network for pollen grain recognition. Int. J. Swarm Intell. Res., Vol. 11, No. 3, pp. 49–71.
29. **Varela-Santos, S., Melin, P. (2021).** A new approach for classifying coronavirus COVID-19 based on its manifestation on chest X-rays using texture features and neural networks. Inf. Sci., Vol. 545, pp. 403–414.
30. **Rashedi, E., Nezamabadi-pour, H., Saryazdi, S. (2009).** GSA: A Gravitational Search Algorithm. Information Sciences, Vol. 179, No. 13, pp. 2232–2248.

31. **González, B., Valdez, F., Melin, P., Prado-Arechiga, G. (2015)**. Fuzzy logic in the gravitational search algorithm enhanced using fuzzy logic with dynamic alpha parameter value adaptation for the optimization of modular neural networks in echocardiogram recognition. Appl. Soft Comput., Vol. 37, pp. 245–254.
32. **Olivas, F., Valdez, F., Melin, P., Sombra, A., Castillo, O. (2019).** Interval type-2 fuzzy logic for dynamic parameter adaptation in a modified gravitational search algorithm. Inf. Sci., Vol. 476, pp. 159–175.
33. **Castillo, O., Amador-Angulo, L. (2018).** A generalized type-2 fuzzy logic approach for dynamic parameter adaptation in bee colony optimization applied to fuzzy controller design. Inf. Sci., pp. 460–496.
34. **Thomaz, C.E. (2012).** https://fei.edu.br/~cet/facedatabase.html
35. **Frontalized Faces in the Wild (2016).** www.micc.unifi.it/resources/datasets/frontalized-faces-in-the-wild/.
36. **Tarres, F. (2011).** https://gtav.upc.edu/en/research-areas/face-database.
37. **ORL Face Database**. http://www.uk.research.att.com/facedatabase.html.
38. **Georgia Tech Face Database**. http://www.anefian.com/research/face_reco.html.
39. **Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, Labeled, E. (2007).** Faces in the wild: A database for studying face recognition in unconstrained environments. https://hal.inria.fr/inria-00321923.
40. **Tarres, F., Rama, A. (2011).** GTAV Face Database. https://gtav.upc.edu/en/research-areas/face-database.
41. **Wolf, L., Hassner, T., Maoz, I. (2011).** Face recognition in unconstrained videos with matched background similarity. Proceedings of the Computer Vision and Pattern Recognition (CVPR), Colorado Springs, pp. 529–534.
42. **Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J. (2000).** The FERET Evaluation Methodology for face recognition algorithms. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 22, pp. 1090–1104.
43. **Deng, L. (2012).** The MNIST database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine, Vol. 29, No. 6, pp. 141–142.
44. **Sombra, A., Valdez, F., Melin, P., Castillo, O. (2013).** A new gravitational search algorithm using fuzzy logic to parameter adaptation. IEEE Congress on Evolutionary Computation, pp. 1068–1074.
45. **Martínez, G.E., Mendoza, O., Castro, J.R., Rodríguez-Díaz, A., Melin, P., Castillo, O. (2017).** Comparison between Choquet and Sugeno integrals as aggregation operators for pattern recognition. Annual. Conf. North Am. Fuzzy Inf. Process. Soc., NAFIPS, pp. 2–7.
46. **Korkmaz, M., Yilmaz, N. (2015).** Face recognition by using back propagation artificial neural network and windowing method. Manuscript Received.
47. **González, C.I., Melin, P., Castro, J.R., Castillo, O. (2017).** Generalized type-2 fuzzy edge detection applied on a face recognition system. Edge Detection Methods Based on Generalized Type-2 Fuzzy Logic, Springer Briefs in Applied Sciences and Technology, Springer, Cham. DOI: 10.1007/978-3-319-53994-2_6.
48. **Jelšovka, D., Hudec, R., Brezňan, M. (2011).** Face recognition on FERET face database using LDA and CCA methods. 34th International Conference on Telecommunications and Signal Processing (TSP), pp. 570–574, DOI: 10.1109/TSP.2011.6043665.
49. **Sánchez, D., Melin, P., Castillo, O. (2017).** A grey wolf optimizer for modular granular neural networks for human recognition. Comput. Intell. Neurosci., No. 41805101-4180510, pp. 26.

Article received on 08/06/2021; accepted on 16/11/2021.
Corresponding author is Patricia Melin.

ISSN 2007-9737

Collaborative Recommender System Based on Improved Firefly Algorithm

Bharti Sharma[1], Adeel Hashmi[2], Charu Gupta[3], Amita Jain[4]

[1] Maharaja Surajmal Institute of Technology
Department of Information Technology
Delhi, India

[2] Maharaja Surajmal Institute of Technology,
Department of Computer Science and Engineering,
Delhi, India

[3] Bhagwan Parshuram Institute of Technology
Department of Computer Science and Engineering,
Delhi, India

[4] Netaji Subhas University of Technology,
Department of Computer Science and Engineering,
Delhi, India

charugupta0202@gmail.com, amita_jain_17@yahoo.com

Abstract. A recommendation system aims to capture the taste of the customer and predict relevant items which he/she may be interested in buying. There are many algorithms for generating recommendations in literature, however, most of them are non-optimal and do not have the capability to handle big data. In this paper, a collaborative recommendation system is proposed based on improved firefly algorithm. The firefly algorithm is used to generate optimal clusters which provide effective recommendations. The proposed algorithm works in two phases: Phase I which generates the clusters with firefly algorithm and Phase II gives real time recommendations. The firefly algorithm has been implemented in Apache Spark to give it the capability of handling big data. The combination of improved firefly-based clustering and Apache Spark makes it much faster and optimal than the state-of-the-art recommendation models. For experiments, movie-lens dataset has been utilized and different evaluation metrics have been used for performance analysis. The results show that the proposed method gives better results compared to existing methods.

Keywords. Clustering, collaborative filtering, firefly algorithm, recommender system, swarm intelligence

1 Introduction

Recommender systems aim to suggest the items a customer might like based on the information about his/her preferences and ratings. Recommendation system can be viewed as an extension to association/pattern mining. It has been observed if an item B is associated with item A then whenever any user buys item A, he is recommended item B and vice-versa [1, 2].

Recommendation systems are useful for both buyers and sellers since they reduce buyer's effort and increase sales. These systems are put to use in many fields like e-commerce websites, news filtering, web searches, online dating, social networking sites [3, 4, 5]. Movie recommendation or movie rating prediction is a popular use-case of recommender systems [6, 7]. It is analyzed that the state-of-the-art methods are slow, non-scalable and their achieved accuracy needs improvement. In this paper, a fast and scalable method to generate recommendations is proposed which is optimized by improved firefly algorithm.

The traditional methods of clustering like k-means algorithm are slow, so firefly optimization algorithm is used to create clusters. This firefly clustering algorithm is made scalable and parallelized by utilizing Apache Spark tool.

Firefly is a population based algorithm which has some additional advantages as compared to single point search algorithms. Some of the most important fields of its application are optimization of dynamic and noisy environment and constraints, combinatorial and multi-objective optimization. Apart from the field of optimization it is also capable of solving classification problems that we come across in the fields of neural network, data mining and machine learning. Clustering techniques are used to group similar items or objects together based on unsupervised learning. In this paper, the data set is divided based on random cluster heads and then the cluster-heads are re-calculated iteratively for optimal use in Firefly algorithm. The detailed working methodology and mathematical foundation is given in section 3.

The remainder of the paper is organized as follows. Section 2 surveys the various recommender algorithms. Section 3 introduces the vanilla version of firefly algorithm. Section 4 explains the working of proposed algorithm, improved firefly algorithm to generate optimal recommendations. Section 5 provides experimental results and analysis. Section 6 presents the conclusion along with future directions.

2 Recommender Algorithms

Recommender algorithms fall into three categories [8, 9]: content-based, collaborative and hybrid as shown in Figure-1. Collaborative filtering is based on the concept of user-ratings, where ratings given to the products by every user are stored, and for a user X the persons who have similar rating pattern are identified, and those products are recommended which were given high ratings by this identified group of people.

The recommender systems in addition to collaborative filtering also has approaches based on content-based methods on information retrieval, Bayesian inference, and case-based reasoning methods [10, 11]. These methods take the actual content or attributes of the items to make recommendation (instead of or in addition to patterns with user rating). Content-based algorithms recommend to a customer those items which are similar to items that the same customer has bought or searched in the past. Hybrid recommender systems [12] have also emerged as a recommendation technique combining content-based and collaborative algorithms into composite systems that build on the strengths of their algorithmic components.

Content-Based Filtering systems recommend an item based on the contents of that item. If a user has previously searched for, or looked at some items with attribute ‘A’ then more items with attribute A will be recommended. Thus recommendations are made by comparing the contents of an item with the profile of the target user. The profile of a user is built from his history of interaction with the system by modeling the user’s preferences. The attributes can be assigned automatically or manually. The attributes have to be represented such that the user profile and the items can be compared to extract meaningful relations. A learning algorithm which can create the user profile based on items bought/viewed is also needed [13].

Collaborative Filtering (CF) is a process in which ratings are obtained from the users and recommendations to a new user are given based on opinions of other users with similar taste. The items that are recommended to a user are based upon his/her similarity to other users. For example, if two users X and Y have shown similar preferences in the past then the items which are liked by X in the future will be recommended to Y and vice versa. So basically, this algorithm assumes that if some user A has the same view as user B on an issue then A is more likely to have the same view as B on any other issue as well [14]. Collaborative Filtering based approach can be further divided into two categories: Memory-based and Model-based.

Memory-Based approach is a simple approach which makes use of a similarity measure to find users/items which are related to the active user or

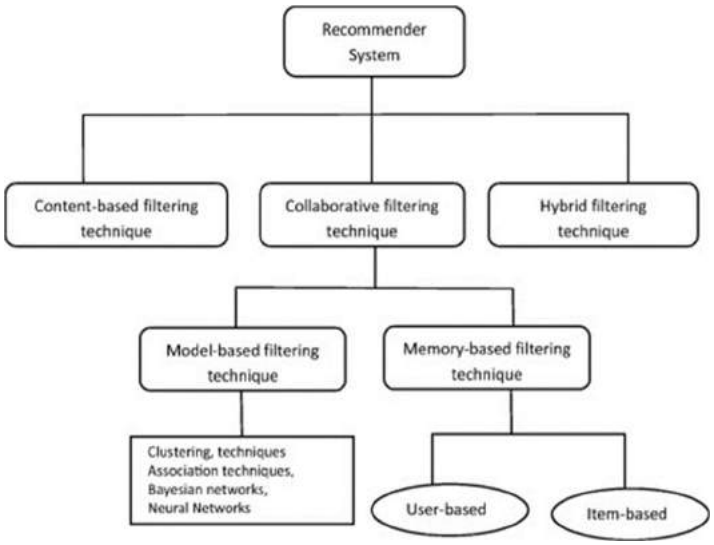


Fig. 1. Types of Recommender Systems

to the items bought/viewed by the active user. The methods that can be used to find out the similarity between two users are Euclidean distance, cosine similarity, correlation, etc.

i. User-based approach: User–user collaborative filtering was the first of the automated CF approach. It was first introduced in the GroupLens Usenet article recommender [15]. Then those items which have been highly rated by most of these users are identified and recommended to the active user [16, 17, 18].

ii. Item-based approach: Item-based collaborative filtering models the ratings item-wise and not user-wise. An item-item matrix is built to determine relationship between every pair of items. A similarity measure like correlation is used to build this matrix. When a customer rates an item A then this matrix is looked up to find the items which have highest similarity with A in the matrix. Slope [19, 20]

Model-Based approach is basically dependent upon machine learning, data mining algorithms to make predictions. In this approach the aim is not to find most similar users/items, but to develop a model to classify the user and recommend highly rated items of other users belonging to the same class. The machine learning algorithms used in model-based approach are clustering, Bayesian networks, singular value decomposition (SVD), etc. This model has advantage over memory-based approach in the sense that it provides faster recommendations, handles sparsity better than memory-based ones, scales with dataset and has better prediction performance.

Many recommender systems combine the memory-based and model-based collaborative filtering algorithms which can be called **hybrid collaborative filtering**. This type overcomes the limitations of both the other types but increases complexity and is expensive. The hybrid approach can be used to overcome some of the common problems that occur when either of the other two approaches is used independently. It has been observed that the hybrid approach provides more accurate recommendations than either of the two approaches [21]. A popular example of hybrid approach is content-boosted collaborative filtering [22]. Apart from these popular techniques, there are some other recommendation techniques as well.

Knowledge-Based Recommenders. These recommender systems area specific kind of

recommender systems that are based on prior knowledge about all the items that are available and also knowledge about user preferences [23].

Demographic recommenders. As the name suggests, these recommender systems provide recommendations based on a demographic profile of the user. The ratings given by users in a particular demographic section are used to provide recommendations to a user of that particular section. Here are also few problems which are encountered by recommender systems like cold start, sparsity, trust and privacy [24].

3 Firefly Algorithm

Firefly algorithm developed by Yang in 2008 [25] is a meta-heuristic algorithm used to solve optimization problems. Firefly algorithm is among those stochastic algorithms which follow randomization approach to search the solution in the data set. In this section, the biological, mathematical foundation and behavior of firefly is presented. It also explains the intuition and foundation of clustering with firefly algorithm.

3.1 Biological Foundation and Behavior

Fireflies are distinguished by their flashing light which is produced by a biochemical process also known as bioluminescence [26, 27, 28] . The rhythmic flashes are used as signals for mating [29, 30]. Apart from attracting the mating partners, these bright lights are used as warning signals from potential predators.

Firefly algorithm produced use the following assumptions [25]:

- The brightness of the firefly corresponds to the objective function.
- Each firefly is attracted to all other fireflies as they are unisex.
- A brighter firefly is more attractive, and a less bright firefly will move towards a firefly which is brighter. The attractiveness/brightness decreases as the distance between the fireflies' increases.

3.2 Mathematical Formulation

The light intensity I(r) varies according to the inverse square law:

$$I(r) = I_s / r^2 , \quad (1)$$

where, I_s is the light intensity at source.

The light intensity I varies with distance r for a stated medium with absorption coefficient λ, acc. to the equation:

$$I = I_0 e^{-\lambda r} , \quad (2)$$

where, I_0 is the actual light intensity.

The Eq-1 and Eq-2 can be combined to give the following equation:

$$I = I_0 e^{-\lambda r^2} , \quad (3)$$

Taking the above equations into consideration, the attractiveness β of a firefly can be defined as:

$$\beta = \beta_0 e^{-\lambda r^2} , \quad (4)$$

where β_0 is the attractiveness of the firefly at r=0.

In the real time environment, the attractiveness function of the firefly i.e. $\beta(r)$ can be any monotonically deceasing function described in the generalize form as:

$$\beta(r) = \beta_0 e^{-\lambda r^m} \ (m \geq 1) , \quad (5)$$

The Cartesian distance (*r*) is the distance between any two random fireflies *i* and *j* at location x_i and x_j, respectively:

$$r_{ij} = \left\| x_i - x_j \right\| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2} , \quad (6)$$

where, $x_{i,k}$ is the k^{th} dimension of the spatial coordinate x of the i^{th} firefly.

In 2-dimensional case, we have:

$$r_{ij} = \sqrt{(x_i - x_j)^2 - (y_i - y_j)^2}\,, \tag{7}$$

Firefly *i* is attracted to another more attractive firefly *j* according to equation (8):

$$x_i = x_i + \beta e^{-\lambda r^2}(x_j - x_i)\,, \tag{8}$$

The value of the parameters λ plays a significant role determining the speed of convergence and it follows the range of 0 to 10.

The pseudo-code for firefly algorithm is given below in Figure-2.

The above algorithm is only for exploitation part (finding the local best solution). For exploration part (to find global solution), we make use of the Levy flight instead of the traditional method. To make the process of exploitation faster, the less bright fireflies are moved towards brightest firefly only instead of all the brighter fireflies [31, 32].

3.3 Clustering Using Firefly Algorithm

In this section, the main aim is to calculate cluster heads by minimizing the sum of calculated distances of the patterns with their cluster heads [34, 35]. The function to be minimized during clustering process can be described as given in Eq-9:

$$J(k) = \sum_{k=1}^{M} \sum_{i \in c_k} (x_i - c_k)\,, \tag{9}$$

where M is the no. of clusters, c_k is the cluster head of k^{th} cluster, and x_i is a data point belonging to the cluster.

The cluster head of a cluster is the centroid of the cluster. The centroid of a cluster with n points can be calculated by Eq-10:

$$c_k = \sum_{i \in c_k} \frac{x_i}{n_k}\,, \tag{10}$$

where n_k is number of points in the k^{th} cluster.

By performing clustering, we can divide a dataset into different groups based on some similarity measures. Most widely used similarity measures are based on distance calculation between the dataset and the cluster heads [36].

The cluster heads are calculated by minimizing the Euclidean distance between each data instance X_i and the cluster center c_k. The cost function for the pattern *i* is given by Eq-11:

$$f_i = \frac{1}{D} \sum_{j=1}^{D} d(x_j, p_i^{C_q})\,, \tag{11}$$

where D is the count of data instances, and $p_i^{C_q}$ defines the class *q* to which the instance *i* belongs. The proposed pseudo-code for clustering through firefly algorithm is given in Figure-3.

4 Proposed Firefly Recommendation System (FRS)

The proposed Firefly Recommendation System i.e. (FRS) works in *two phases* which includes training phase and recommendation phase. **Phase I** is an offline process in which rating matrix is produced from the collected data and clusters are obtained using firefly clustering algorithm. **Phase II** is a real-time process in which the recommendations for current user are generated.

In this phase, the active user is assigned a recommendation cluster and recommendations are generated.

Phase I: Training Phase

The movie-lens dataset has 100,000 ratings of 943 users on 1682 movies. The movies are classified into 19 genres viz. action, comedy, horror, etc. The dataset is divided into two parts, 80% as training data and 20% as test data.

The data is converted into a 943X1682 matrix. The dataset need not be normalized as the ratings are in the scale of 1-5. However, the dataset is sparse (only 100,000 ratings out of possible 1,586,126 available), so we need to replace the missing values by 0.

The rating matrix is divided into K clusters using firefly clustering technique. N fireflies are initially generated, each having K cluster-heads.

Firefly Meta-heuristic Algorithm(FFA)

Begin
Initialize the algorithm parameters:
MaxGen: the maximal number of generations
λ : the light absorption coefficient
r : distance from the light source
d : the domain space
Identify the objective principle of f(x), where $x=(x_1,.....,x_d)^T$
Generate the initial population of fireflies x_i (i=1, 2 ,..., n)
Verify the light intensity of I_i at x_i via $f(x_i)$
While (t<MaxGen)
    For i = 1 to n (all n fireflies);
        For j=1 to n (n fireflies)
            If ($I_j > I_i$), move firefly i towards j by using equation (8);
            end if
            Attractiveness varies with distance r via $\exp[-\gamma r^2]$;
            Calculate new result and update light intensity;
            End for j;
        End for i;
        Rank the fireflies and find the current best;
    End while;
    Display results;
End procedure

Fig. 2. Firefly Meta-heuristic

FIREFLY_CLUSTERING (normalized_dataset, k_clusters, N_fireflies)

1. Generate fireflies
   - Create N fireflies, each having k cluster-heads (D-dimensional) with random normalized values.
   - Create clusters for each firefly (based on shortest Euclidean distance).
2. Calculate the fitness of each firefly. The fitness of a firefly is the average of Euclidean distance of all the points from their assigned cluster-head.
   Intensity of firefly (I) = Fitness of firefly
3. Rank the fireflies according to their intensity (higher intensity, higher rank).
4. Compare each firefly (starting with lowest ranked) with other fireflies. Move the firefly towards brighter firefly otherwise move it randomly. (Check that the new position lies in range 0-1).

$$\beta = \beta_0 e^{-\lambda r^2},$$

$$x_i = x_i + \beta e^{-\lambda r^2}(x_j - x_i).$$

5. Obtain the updated cluster-heads of each firefly after step-2. Re-form clusters of each firefly.
6. Re-calculate fitness/intensity of each firefly. Replace old firefly with new firefly if it has higher intensity. Repeat step-2.
7. Repeat Step 3-6 for MaxGenerations (e.g. 1000 times).
8. Display the cluster-heads of the brightest firefly.

Fig. 3. Proposed Firefly clustering algorithm

Each cluster-head has 1682 dimensions having values in the range 1-5, which are generated at random.

For each firefly, K clusters are generated by assigning each point in the dataset to the nearest cluster-head in the firefly (similarity measure used is Euclidean distance), and the WCSS (within-cluster sum of squares) is calculated.

The firefly with the lowest WCSS is considered to be the brightest firefly, and the less bright fireflies are moved towards the brightest fireflies.

The brightest firefly is also moved at random to a position which further increases the intensity of the brightest firefly.

This process is repeated to certain number of iterations, and the fittest firefly after all these

iterations is considered to be the final solution (clusters).

Phase-II: Process of recommendation for active users.

To generate recommendations for an active user, a cluster (among k-clusters) is to be selected. A simple approach is to select the cluster whose centroid has highest similarity with the active user e.g. the centroid with lowest Euclidean distance with the active user. If there are large numbers of clusters, then multiple clusters can also be used for better results. In such a case, the probability that a cluster is chosen for generating recommendations is given by Eq-12:

$$P_i = \frac{\rho_i * d_i}{\sum_{i=1}^{k} \rho_i * d_i}, \tag{12}$$

where, ρ_i is the density of the cluster, and d_i is the Euclidean distance between active user profile and centroid of cluster:

$$\rho_i = \frac{N_i}{\sum_{i=1}^{k} N_i}, \tag{13}$$

where N_i is the number of users in cluster i.

The recommendations are provided from the cluster with highest probability or from multiple clusters which lie in particular probability range. The latter approach may provide the active user recommendations which are different and make him interested in trying something new.

After selecting the clusters for recommendations, next step is to predict the ratings for un-rated items of the active user and recommending the items whose predicted value is high. If there is only one chosen cluster, then the values of unrated items is simply the average of the ratings given for corresponding item by all the users in the cluster.

But if multiple clusters have been selected then we also consider the quality of ratings in each chosen cluster. A criterion of the rating quality of a cluster is the number of ratings available to each item in the cluster, higher the density of ratings better the quality of the cluster:

$$Q_i = \frac{\sum_{p=1}^{t} r_{ip}}{n_i * t}, \tag{14}$$

where, Q_i is the quality of cluster i, n_i is the number of users in the cluster i, t is the number of items, and r_{ip} is the count of ratings available for item p in the cluster i.

5 Experimental Results and Analysis

For performance analysis of our recommendation system framework we calculate various metrics like MAE, SD, RMSE and t-value. Various graphs and tables of the calculated results are shown for better understanding of the framework.

MAE: Mean Absolute Error

We calculated mean absolute error on the dataset of movielens dataset by using Eq-15:

$$MAE = \frac{\sum |p_{ij} - t_{ij}|}{M}, \tag{15}$$

where, M is no. of movies in the dataset, p_{ij} is predicted value for *i* user on *j* items, and t_{ij} is true rating.

The results are shown in Table 4 for the calculated MAE for different values of K. The outcome as observed from this table is that as we increase the number of clusters, MAE values gradually decrease.

SD: Standard Deviation

By using Eq-16, we calculate SD on movie lens dataset:

$$SD = \frac{\sum_{i=1}^{k} \left\{ \sum_{j=1}^{n_i} \left(\sqrt{\frac{\sum_{l=1}^{D} l - \overline{l}}{D}} \right) \right\}}{n_i}, \tag{16}$$

The results of SD with different cluster count are shown in Table-4. The outcome of this calculated metrics is that as the number of cluster increases their SD value decreases.

Table 1. Snapshot of Movielens dataset

	Movie	Movie	…	Movie
	#1	**#2**		**#1682**
User#1	5	3	…	0
User#2	4	0	···	0
.	.	.	…	.
.	.	.		.
.	.	.		.
User#943	0	5	…	0

Table 2. Sample Snapshot of Fireflies (20 fireflies with 3 cluster-heads each)

		Movie	Movie	...	Movie
		#1	**#2**		**#1682**
Firefly#1	**Cluster-Head #1**	2	3	...	3
	Cluster-Head #2	3	1	...	5
	Cluster-Head #3	4	1	...	2
.	.. .	.. .	.. .	...	. .
Firefly#20	**Cluster-Head #1**	5	1	...	1
	Cluster-Head #2	1	2	...	2
	Cluster-Head #3	2	4	...	4

Table 3. Sample of cluster assigned to each user in fittest firefly (assuming that there are 3 clusters in each firefly)

	User	User	User	…	User
	#1	**#2**	**#3**		**#943**
Cluster#	3	1	3	…	2

RMSE: Root Mean Square Error

We calculated RMSE on movie lens dataset by using Eq-17:

$$RMSE = \sqrt{\frac{\sum (p - t)^2}{n}}, \tag{17}$$

where, p is the predicted value, t is the actual

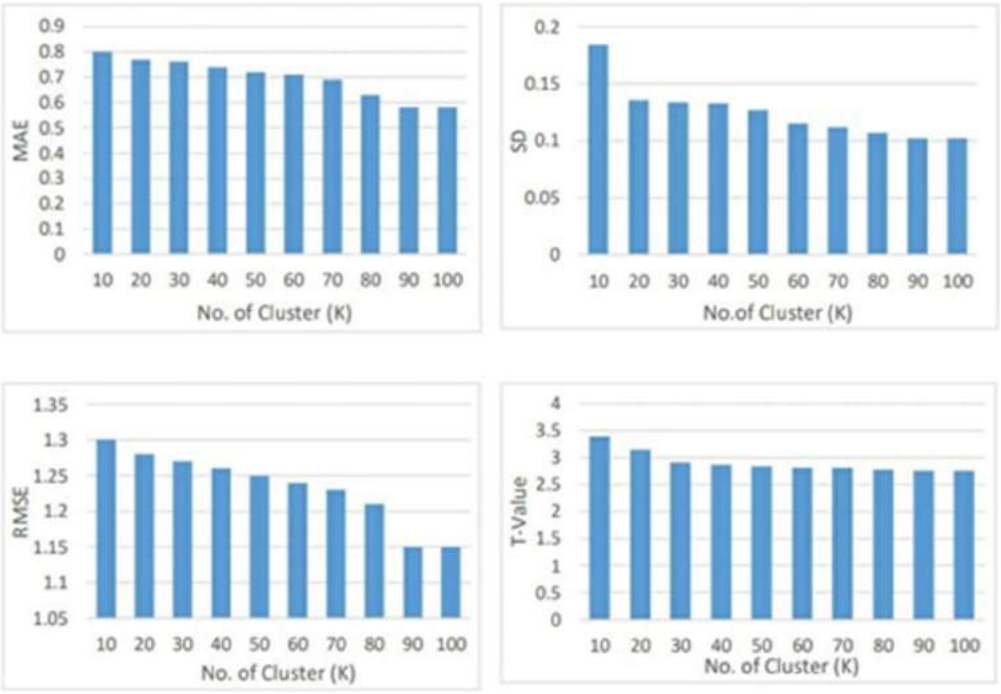


Fig. 4. Performance of proposed algorithm on changing cluster size

value, n is the number of predicted ratings.

The results after calculation of RMSE for different cluster count are represented in Table-4. It is observed that the RMSE value gradually decreases as we increase the number of clusters like other metrics like MAE and SD.

t-value

This t-value basically depends on the values of mean obtained for different clusters and their calculated SD values. We calculate t-value (for significance level of 5%) of the dataset by using Eq 18:

$$t-value = \sum_{i=1}^{k}\sum_{j=1}^{k}\left(\frac{\overline{X}_i - \overline{X}_j}{\sqrt{\frac{(SD_i)^2}{n_i} + \frac{(SD_j)^2}{n_j}}}\right) \quad (18)$$

.

Similar to the other matrices, t-value also decreases for the same reason as mentioned above. Results are shown in Table 4.

The performance of proposed firefly-based recommendation system was also compared with the other popular clustering-based techniques like k-means, PSO (Particle Swarm Optimization), and ACO (Ant Colony Optimization), Bat algorithm, Cuckoo search. All the algorithms were run for 100 iterations. The performance of firefly-based recommendation was slightly better than all other techniques as can be seen in the Table 5 and Figure-4.

6 Conclusion and Future Work

This paper proposed an improved firefly meta-heuristic based clustering approach for recommendation systems. A clustering based recommender system should be able to generate optimal clusters, hence firefly algorithm was

Table 4. Performance based on cluster size

No. of clusters (k)	MAE	SD	RMSE	t-value
10	0.8	0.184	1.3	3.39
20	0.77	0.136	1.28	3.15
30	0.76	0.134	1.27	2.91
40	0.74	0.133	1.26	2.87
50	0.72	0.127	1.25	2.84
60	0.71	0.115	1.24	2.81
70	0.69	0.112	1.23	2.81
80	0.63	0.107	1.21	2.77
90	0.58	0.102	1.15	2.75
100	0.58	0.102	1.15	2.75

Table 5. Performance comparison with other algorithms (k=90)

	k-means	PSO	ACO	Bat	Cuckoo	Firefly
MAE	0.69	0.7	0.7	0.67	0.71	0.58
SD	0.113	0.113	0.112	0.107	0.114	0.102
RMSE	1.23	1.23	1.22	1.19	1.23	1.15
t-value	2.81	2.81	2.81	2.76	2.81	2.75
Precision	0.53	0.52	0.51	0.54	0.52	0.58
Recall	0.43	0.41	0.41	0.44	0.41	0.47

utilized. The original firefly algorithm has been improved by making it faster by moving the less bright firefly towards only the brightest firefly instead of all the brighter fireflies.

For exploration, Levy flight has been used instead of random function. For fast results, the algorithm is parallelized using map-reduce to enable it to be executed in a scalable environment.

The performance of the proposed approach is evaluated using various metrics and the results indicate that the approach generates highly relevant recommendations. In the future work, other swarm optimization methods like whale

optimization, shark smell optimization, etc. can be utilized.

The optimization methods other than swarm optimization like neural networks can also be utilized. There are various ways in which the recommendations from optimal clusters can be generated, these also can be explored.

References

1. **Kidzinski, L. (2011).** Statistical foundations of recommender systems. Master Thesis, Faculty of Mathematics, Informatics and Mechanics, University of Warsaw.
2. **Deshpande, M., Karypis, G. (2004).** Item-based top-N recommendation algorithms. ACM Transactions on Information System, Vol. 22, No. 1, pp. 143–177. DOI: 10.1145/963770.963776.
3. **Li, P., Yamada, S. (2004).** A movie recommender system based on inductive learning. IEEE Conference on Cybernetics and Intelligent Systems, Vol. 1, pp. 318–323. DOI: 10.1109/ICCIS.2004.1460433.
4. **Wei, K., Huang, J., Fu, S. (2007).** A survey of e-commerce recommender systems. International Conference on Service Systems and Service Management, pp. 1–5. DOI: 10.1109/ICSSSM.2007.4280214.
5. **Porcel, C., Herrera-Viedma, E. (2010).** Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries. Knowledge-Based Systems, Vol. 23, No. 1, pp. 32–39. DOI: 10.1016/j.knosys.2009.07.007.
6. **Porcel, C., Moreno, J.M., Herrera-Viedma, E. (2009).** A multi-disciplinary recommender system to advice research resources in University Digital Libraries. Expert Systems with Applications, Vol. 36, No. 10, pp. 12520–12528. DOI: 10.1016/j.eswa.2009.04.038.
7. **Goldberg, D., Nichols, D., Oki, B.M., Terry, D. (1992).** Using collaborative filtering to weave an information tapestry. Communications of the ACM, Vol. 35, No. 12, pp. 61–70. DOI: 10.1145/138859.138867.
8. **Deshpande, P.K., Banchhor, C. (2014).** Survey on recommender systems. International Journal of Engineering Research and Development, Vol. 10, No. 6, pp. 49–54.
9. **Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., Reiterer, S., Stettinger, M. (2014).** Basic approaches in recommendation systems. Recommendation Systems in Software Engineering, Springer, pp. 15–37. DOI: 10.1007/978-3-642-45135-5_2.
10. **Schafer, J.B., Konstan, J.A., Riedl, J. (2001).** E-commerce recommendation applications. Data Mining and Knowledge Discovery, Vol. 5, pp. 115–153. DOI: 10.1023/A: 1009804230409.
11. **Pazzani, M., Billsus, D. (2007).** Content-based recommendation systems. The Adaptive Web, 325–341.
12. **Burke, R. (2002).** Hybrid recommender systems: Survey and experiments. User Modelig and User-Adapted Interaction, Vol. 12, No. 4, pp. 331–370. DOI: 10.1023/A: 1021240730564.
13. **Smyth, B. (2007).** Case-based recommendation. The Adaptive Web, Lecture Notes in Computer Science, Vol. 4321, pp. 342–376. DOI: 10.1007/978-3-540-72079-9_11.
14. **Hill, W., Stead, L., Rosenstein, M., Furnas, G. (1995).** Recommending and evaluating choices in a virtual community of use. Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 194–201.
15. **Resnick, P., Lacovou, N., Suchak, M., Bergstrom, P., Riedl, J. (1994).** GroupLens: an open architecture for collaborative filtering of netnews. ACM conference on Computer supported cooperative work, pp. 175–186. DOI: 10.1145/192844.192905.
16. **Vucetic, S., Obradovic, Z. (2000).** A regression-based approach for scaling-up personalized recommender systems in e-commerce. Workshop on Web Mining for E-Commerce (WEBKDD'00), pp. 1–9.
17. **Zhao, Z.D, Shang, M.S. (2010).** User-based collaborative-filtering recommendation algorithms on Hadoop. Third International Conference on Knowledge Discovery and

Data Mining, pp. 478–481. DOI: 10.1109/WKDD.2010.54.

18. **Zhu, X., Ye, H., Gong, S. (2009).** A personalized recommendation system combining case-based reasoning and user-based collaborative filtering. Chinese Control and Decision Conference, pp. 4026–4028. DOI: 10.1109/CCDC.2009.5192712.
19. **Sarwar, B., Karypis, G., Konstan, J., Riedl, J. (2001).** Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web, pp. 285–295.
20. **Sun, Z., Luo, N. (2010).** A new user-based collaborative filtering algorithm combining data-distribution. Information Science and Management Engineering (ISME), Vol. 2, pp. 19–23. DOI: 10.1109/ISME.2010.48.
21. **Murugasamy, K., Murugasamy, K. (2016).** Hybrid clustering using firefly optimization and fuzzy c-means algorithm. Circuits and Systems, Vol. 7, No. 9, p. 2339–2348. DOI: 10.4236/cs.2016.79204.
22. **Shardanand, U., Maes, P. (1995).** Social information filtering: algorithms for automating “word of mouth”. Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 210–217.
23. **Vozalis, E.G., Margaritis, K.G. (2003).** Recommender systems: An experimental comparison of two filtering algorithms. Proceedings of the 9th Panhellenic Conference in Informatics-PCI, pp. 152–166.
24. **Sarwar, B.M. (2001).** Sparsity, scalability, and distribution in recommender systems. Doctoral Dissertation, University of Minnesota.
25. **Yang, X.S. (2010).** Nature-inspired metaheuristic algorithms, 2nd ed. Luniver Press.
26. **Zang, H., Zhang, S., Hapeshi, K. (2010).** A review of nature-inspired algorithms. Journal of Bionic Engineering, Vol. 7, No. 4, pp. S232--S237. DOI: 10.1016/S1672-6529(09)60240-7.
27. **Yang, X.S. (2010).** Firefly algorithm, stochastic test functions and design optimisation. International Journal of Bio-Inspired Computation, Vol. 2, No. 2, pp. 78–84.
28. **Sajwan, M., Acharya, K., Bhargava, S. (2014).** Swarm intelligence based optimization for web usage mining in recommender system. International Journal of Computer Applications Technology and Research, Vol. 3, No. 2, pp. 119–124. DOI: 10.7753/IJCATR0302.1007.
29. **Yang, X.S., He, X. (2013).** Firefly algorithm: recent advances and applications. International Journal of Swarm Intelligence, Vol. 1, No. 1, pp. 36–50.
30. **Fister, I., Fister Jr., I., Yang, X.S., Brest, J. (2013).** A comprehensive review of firefly algorithms. Swarm and Evolutionary Computation, Vol. 13, pp. 34–46. DOI: 10.1016/j.swevo.2013.06.001.
31. **Khan, W.A., Hamadneh, N.N., Tilahun, S.L., Ngnotchouye, J.M. (2016).** A review and comparative study of firefly algorithm and its modified versions. Optimization Algorithms-Methods and Applications, IntechOpen. DOI: 10.5772/62472.
32. **Kumar, M.S., Prabhu, J. (2019).** Hybrid model for movie recommendation system using fireflies and fuzzy c-means. International Journal of Web Portals (IJWP), Vol. 11, No. 2, pp. 1–13. DOI: 10.4018/IJWP.2019070101.
33. **Yang, X.S. (2010).** Firefly algorithm, Lévy flights and global optimization. Research and development in intelligent systems XXVI, Springer, pp. 209–218. DOI: 10.1007/978-1-84882-983-1_15.
34. **Senthilnath, J., Omkar, S.N., Mani, V. (2011).** Clustering using firefly algorithm: performance study. Swarm and Evolutionary Computation, Vol. 1, No. 3, pp. 164–171. DOI: 10.1016/j.swevo.2011.06.003.
35. **Hassanzadeh, T., Meybodi, M.R. (2012).** A new hybrid approach for data clustering using firefly algorithm and K-means. 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP), pp. 7–11. DOI: 10.1109/AISP.2012.6313708.
36. **Mohammed, A.J., Yusof, Y., Husni, H. (2015).** Determining Number of Clusters Using Firefly Algorithm with Cluster Merging for Text Clustering. Advances in Visual Informatics (IVIC), Lecture Notes in Computer Science,

Vol. 9429, pp. 14–24. DOI: 10.1007/978-3-319-25939-0_2.

Article received on 15/06/2021; accepted on 17/11/2021.
Corresponding author is Bharti Sharma.

ISSN 2007-9737

Performance Evaluation of the Angle Modulated Particle Swarm Optimization Algorithm in a Heterogeneous Network in Shared Spectrum Access

Anabel Martínez-Vargas[1], Ángel G. Andrade[2]

[1] Universidad Politécnica de Pachuca,
Mexico

[2] Universidad Autónoma de Baja California,
Facultad de Ingeniería,
Mexico

anabel.martinez@upp.edu.mx, aandrade@uabc.edu.mx

Abstract. Spectrum assignment (SA) controls the interference among secondary and primary users using spectrum sharing (SS) access. SA assigns the appropriate frequency band to a secondary user according to a predefined criterion. This work proposes a SA approach that controls the allocation channels to minimize network interference. The angle modulated particle swarm optimization (AMPSO) algorithm is applied to maximize the heterogeneous network (HetNet) throughput when secondary users exploit a channel simultaneously with a primary user. The AMPSO results are compared with the memory binary particle swarm optimization (MBPSO), the socio-cognitive particle swarm optimization (SCPSO), and the modified version of binary particle swarm optimization (ModBPSO). Comparison results showed that AMPSO is suited for scenarios with high quality of service (QoS) requirements and many secondary and primary users deployed in an area. AMPSO presents the best performance by maximizing spectrum reuse. It selects the secondary and primary users to share a communication channel and maximizes the total throughput in the HetNet.

Keywords. Angle modulated particle swarm optimization, spectrum assignment, spectrum sharing, heterogeneous network.

1 Introduction

The continued development of radio technology and new services has increased the world's dependence on wireless communications, growing the demand and the cost of the radio spectrum finite resource [31]. By 2023, over 70 percent of the global population will have mobile connectivity [6]. To address this growing challenge, regulators will require policies, new approaches, and technological innovations to enable flexible and efficient access to the radio spectrum.

Today, the static spectrum allocation policy regulates wireless networks. Regulators decide on the usage of a spectrum band, providing a license to each user to transmit on a frequency over a specific area. This rigid spectrum regulation guarantees that destructive interference among wireless technologies does not occur [4]. However, it has led to the under-utilization of the radio spectrum, as studies have pointed out [11]. In this context, SS becomes a promising approach to improving spectrum usage efficiency [3].

SS enables mobile users to use a frequency band in a specific geographical area from different wireless communication technologies. An SS network has two kinds of users: the primary users (PUs) and the secondary users (SUs). PUs have guaranteed access since they are licensed users. Consequently, SUs access the licensed spectrum if they do not harm the operation of the PUs. Hence, the PUs do not experience service degradation due to interference caused by the SUs [24].

An important issue of SS strategy is the QoS requirements concerning the signal-to-interference-noise ratio (SINR) for both PUs and SUs during the concurrent spectrum access. By facing this issue, no network tiers undergo service degradation due to the interference, achieving a peaceful coexistence.

SA is a key task to accomplish the SS approach. SA limits the interference between SUs and PUs operating in the same geographical area by assigning the appropriate frequency band to an SU by one or more criteria: interference/power, throughput, fairness, delay, price, energy efficiency, risk, and network connectivity [32].

After that, a suitable technique is selected to solve the objective(s) such as heuristics, graph theory, linear programming, fuzzy logic [22], evolutionary algorithms [25], swarm algorithms [33], etc.

For conventional cellular networks, the SS approach enlarges the pool of available spectrum resources for mobile users through femtocells (small cells), overlaid on the existing macrocell. That mixture of different types of cells is known as a HetNet [1]. However, in a HetNet deployment, reusing radio resources leads to destructive interference for macro-users (PUs) and femto-users (SUs) [14]. The unplanned positions of femto-base stations lead to two kinds of interference: cross-tier (the aggressor and the victim of interference belong to different tiers) and intra-tier (the aggressor and the victim of interference belong to the same tier) [5].

This work considers the underlay SS paradigm in a HetNet to propose a solution to the SA problem. Then, we maximize network throughput when one or more SUs reuse a channel simultaneously with the PU, satisfying QoS requirements. The SA problem belongs to the class of the NP-Hard problems, i.e., no known algorithm generates a guaranteed optimal solution in an execution time expressed as a finite polynomial of the problem dimension [32]. Therefore metaheuristics are suitable to tackle the SA problem by discarding solutions in polynomial time [30]. This work determines the maximum HetNet throughput from identifying SUs and PUs that have access to the same spectral band. That solution also ensures a peaceful coexistence among PUs and SUs in terms of interference. We apply metaheuristics to solve the SA problem in HetNet.

In this case, the binary optimization algorithm represents each solution as a binary string. The number of vector elements equals the number of SUs in the HetNet. If n SUs are deployed in HetNet, then the vector solution size is n bits. Therefore, the size of the binary search space doubles with each element (SU) added to the binary string (solution). It is envisioned that deploy ultra-dense small cells in the coverage region of macrocells will be a solution to the exponentially increased traffic in the following years [1].

In light of this, we deal with a high-dimensionality problem that enlarges the search space, increasing the computational complexity. The motivation for applying AMPSO to solve the SA problem is its ability to handle higher-dimensional problems [23]. AMPSO reduces a particle to a four-dimensional particle defined in continuous space, with a direct mapping back to binary space.

In previous work, we reported an admission control and channel allocation algorithm [19], based on the underlay shared mode and the MBPSO algorithm. However, from the results obtained in [19], we observed that when the number of SUs in the network increased (high dimensionality), the MBPSO algorithm used did not converge to a good solution because the optimization complexity of the SA problem increased. The AMPSO algorithm offers a way to reduce the complexity of binary problems faster than conventional BPSO algorithms. Therefore, we consider applying the AMPSO technique to solve the SA problem in scenarios with a high density of SUs and QoS requirements in the wireless network. The purpose of our work is to evaluate the efficacy of the AMPSO algorithm to find a solution in those complex scenarios.

Other studies have addressed the throughput maximization in HetNets. For example, work in [29] considers an LTE HetNet composed of femtocells and macrocells. It proposes a centralized scheduling approach to mitigate interference and maximize the throughput of the HetNet. Then an optimization problem is formulated as a mixed-integer non-linear programming problem

(MINLP). Given that the MINLP is NP-Hard, it is transformed to be solved in polynomial time using a heuristic algorithm inspired by sociological theory. This transformation only applies to a scenario that authors call an apartment environment (OAE) with obstructive structures.

Work in [26] addresses resource allocation in a HetNet composed of one macrocell and several femtocells. It aims to maximize the femto-tier throughput. To reduce the complexity, the authors divide the maximization problem into two sub-problems: the clustering problem and the resource allocation problem. The first problem that forms the femtocell groups is solved by using an evolutionary game. In contrast, the second problem is posed as one of maximization of the throughput within a cluster. By doing this, it is possible to address it by the particle swarm optimization (PSO) technique.

In contrast, work in [27] addresses the SS with the primary objective of increasing the sum throughput system using QoS constraints for both SUs and PUs. They solve the SA problem by applying particle swarm optimization (PSO) in a homogeneous network (802.11), i.e., cells with the same characteristics. Then an optimal relay selection method is coupled. However, work in [15] envisions that the corresponding number of base stations in the network will increase as the number of users increases. So, it emphasizes that the design of the SS techniques must keep in view picocells, femtocells, small cells, etc., simultaneously in the network.

In [36], the authors maximize the D2D users' throughput with minimal interference to the cellular users. This is done in a multi-tier HetNet. Then, the authors propose an autonomous spectrum allocation scheme with distributed Q-learning. The D2D users can learn the wireless environment and select spectrum resources autonomously to achieve the objective through this strategy. The D2D users operate the underlay shared mode, i.e., they reuse spectrum used by cellular users. The authors simulated their scheme using the Monte Carlo technique by executing 10000 runs.

Finally, the study [17] proposes a numerical approach of coexisting LTE and WiFi networks to share an unlicensed spectrum. It maximizes total throughput in a HetNet if an access point (AP) achieves a throughput threshold level. Then, it applies decentralized and centralized traffic management schemes to show a maximum per-user link throughput of an AP and per-user network throughput. The authors characterize the statistical property of the cell load and channel access probability of each AP in a low-complexity form. The per-user link throughput and per-user network throughput are based on the derived mean spectrum efficiencies and maximize them applying Shannon transform to a non-negative random variable. The simulation results conclude in both schemes that offloading traffic from the LTE network to the WiFi network initially improves the per-user network throughput, but it finally leads to its reduction due to too much offloading.

Unlike works [26] and [29], we do not apply any transformation to the objective function to convert the problem into a deterministic problem. Through AMPSO, we handle candidate solutions with high dimensionality. As works in [36] and [27], we also consider QoS constraints in SUs and PUs, i.e., we guarantee successful communication to both kinds of users. Just as works [29] and [17], we assume centralized management in which our proposed approach is processed in the macro base station.

This paper has the following structure: Section 2 presents the system model and the problem statement. Section 3 describes AMPSO. Section 4 describes AMPSO to resolve the SA problem. Section 5 shows simulation results and consequently, Section 6 presents a discussion. Finally, Section 7 concludes the paper and addresses the implications for further research.

2 System Model and Problem Formulation

Fig. 1 is the down-link scenario considered in this work. It is a HetNet where femto-cells (the red dashed circles) exist within the coverage area A of a macrocell (the black dashed circle). The macrocell has a macro-base station (MBS) which communicates with its associated macro-users (PUs). Consequently, the union of a transmitter, i.e. an MBS, and a receiver (a macro-user) is referred to as a primary link. In Fig. 1, the primary links

are the black arrows; each primary link is identified by a number beside the link (the green numbers). Also, each primary link has a primary channel assigned (the number in brackets). The total number of primary links in A is Pl. The primary links have fixed locations. On the other hand, the femto-cell has a femto-base station (FBS) which communicates with its attached femto-user (SU). Then, a secondary link consists of the union of a transmitter (i.e. an FBS) and its corresponding receiver (a femto-user). Each secondary link is identified by a number beside the link (the blue numbers in Fig. 1). Then a primary channel is assigned to several secondary links. The primary channels assigned to each secondary link are the number in braces in Fig. 1. The total number of secondary links in A is Sl.

We assume that FBSs do not have channels to assign to their femto-users, so macro-users must share their primary channels. In the beginning, primary channels are assigned randomly to secondary links. In Fig. 1, we show the case when primary channel 1 is shared among secondary and primary links. The red number 1 means that primary channel 1 is being shared among secondary links 3, 4, 5, and primary link 2. This channel assignment will generate a level of interference between these links, and network capacity will be affected. In the worst case, if the interference exceeds a predefined QoS threshold, this channel assignment will not be valid. Then, it will be necessary to find another configuration to assign channel 1. Also in Fig. 1, other primary channels are being shared. For example, primary channel 4 is being shared between secondary link 1 and primary link 1. Another example is primary channel 3 that is being shared between secondary link 2 and primary link 4.

The SINR (in dB) is the instantaneous ratio of desired energy to interference. It is a metric on a receiver. In single-hop communications, the SINR must accomplish a minimum SINR threshold to indicate a successful reception [2]. Then, SINR relates to QoS. If primary links experience dropped calls or cannot connect because of the high interference due to the presence of the secondary links in the geographical region, the aim of SS is not achieved at all. Consequently, each service has a QoS or SINR threshold to achieve. For example, a voice service has a target QoS of 3 dB to be considered a successful communication between the transmitter and the receiver.

The SINR in a macro-user of a primary link v is given by [20]:

$$SINR_v = (P_v/ldp(v)^n)/(\sum_{k\in\varphi} P_k/dps(k,v)^n), \quad 1 \leq v \leq Pl, \quad (1)$$

where P_v is the transmit power of the primary link v. $ldp(v)$ is the link distance of the primary link v. n is the path loss exponent (a value between 2 and 4). Those parameters characterizes the desired signal. Consequently, φ is the set of the interferers, i.e. the active secondary links that have assigned the same primary channel as the primary link v. k is the index of interferers. P_k is the transmit power of the secondary link k. $dps(k,v)$ is the distance from the transmitter in secondary link k to the receiver in primary link v.

In Fig. 1, $SINR_v$ is computed in the macro-user of the primary link 2. There, the macro-user in primary link 2 has three interferers: secondary links 3, 4, and 5. The aforementioned is the aggregated cross-tier interference, i.e., the total interference from the secondary links that attempt to simultaneously exploit a channel with the primary link v.

Similarly, the SINR in a femto-user of a secondary link u is given by [20]:

$$SINR_u = (P_u/lds(u)^n)/(\sum_{k\in\varphi} P_k/dss(k,u)^n + P_v/dps(v,u)^n), 1 \leq u \leq Sl, \quad (2)$$

where P_u is the transmit power of the secondary link u. $lds(u)$ is the link distance of the secondary link u. n is the path loss exponent (a value between 2 and 4). Those variables characterize the desired signal. Meanwhile, φ is the set of the interferers, i.e., the active secondary links that have assigned the same primary channel as the secondary link u. k is the index of interferers. P_k is the transmit power of the transmitter of secondary link k. $dss(k,u)$ is the distance from the transmitter of secondary link k to the receiver of secondary

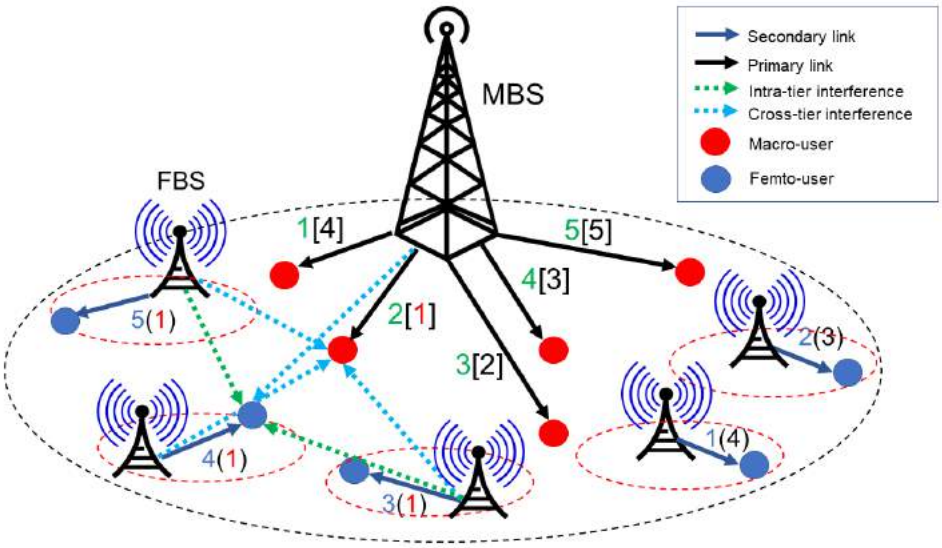


Fig. 1. HetNet scenario

link u. Therefore, those parameters represent the aggregate intra-tier interference.

For example, in Fig. 1, the intra-tier interference on the femto-user of the secondary link 4 comes from the secondary links 3 and 5. Likewise, P_v is the transmit power of the interferer primary link v (it has assigned the same primary channel as secondary link u). $dps(v,u)$ is the distance from the transmitter of the primary link v to the receiver of the secondary link u. Those parameters characterize the cross-tier interference perceived by a receiver of secondary link u. In Fig. 1, the cross-tier interference on the femto-user of the secondary link 4 comes from the primary link 2. In Fig. 1, $SINR_u$ is computed in secondary link 4.

Positive values of SINR indicate that the desired signal is greater than the interference. On the other hand, negative values of SINR refer to that the interference is greater than the desired signal.

Data rate (in Mbps) of the secondary link u and the primary link v are described in equations (3) and (4), respectively [20]:

$$C_u^{'} = Blog_2(1 + SINR_u), \tag{3}$$

$$C_v^{''} = Blog_2(1 + SINR_v), \tag{4}$$

where B is the channel bandwidth that secondary and primary links share. Positive values of SINR result in better throughput. In contrast, negative values of SINR lead to worse throughput.

We aim to optimize the sum throughput in the SS network. We formulate the optimization problem as [20]:

$$Maximize \sum_{u=1}^{Sl} c_u^{'} \cdot x_u + \sum_{v=1}^{Pl} c_v^{''}, \tag{5}$$

subject to

$$SINR_u \geq \gamma, \tag{6}$$

$$SINR_v \geq \alpha, \tag{7}$$

$$c_u^{'} > 0, u = 1, 2, ..., Sl, \tag{8}$$

$$c_v^{''} > 0, v = 1, 2, ..., Pl, \tag{9}$$

$$c_u^{'}, c_v^{''}, \in \mathrm{R}^+, \tag{10}$$

$$x_u = \begin{cases} 1, if\ SINR_u \geq \gamma\ and\ SINR_v \geq \alpha \\ 0, otherwise, \end{cases} \tag{11}$$

The task is to find a binary vector x_u=$(x_1, ..., x_{Sl})$ for which the objective function in equation (5) is maximum. Equation (5) represents the sum throughput of the SS network. It takes into account the selected secondary links, x_u, along with the

primary links that coexist in the same region and share the same spectrum. Equations (6) and (7) are the SINR requirements of the secondary links and primary links respectively.

A successful transmission in the primary link v is achieved if it reaches the SINR threshold α. Similarly, a successful transmission in the secondary link u is reached if its SINR is above the SINR threshold γ. Each position u in the binary vector x in equation (11) symbolizes if secondary link u is related to the primary link v ($x_u = 1$) or not ($x_u = 0$).

3 Angle Modulated Particle Swarm Optimization Algorithm

AMPSO [23] is an alternative version of binary particle swarm optimization (BPSO) [13] to address high dimensionality problems.

To do so, AMPSO employs standard PSO to optimize the coefficients of the following trigonometric function:

$$g(x) = sin[2\pi(x-a)b \cdot cos(2\pi(x-a)c)] + d]. \quad (12)$$

The function in (12) is called the generating function, and it is used as a bit string generator. To optimize the coefficients of the generating function, the position of a particle i is composed of a four-dimensional vector $X_i = (a_i, b_i, c_i, d_i)$. The coefficient a controls the horizontal shift of the entire function. The coefficient b influences the maximum frequency of the sine wave and controls the amplitude of the cosine wave. The coefficient c affects the frequency of the cosine wave (which changes the rate at which the frequency of the sine function changes), and d controls the vertical shift of the function.

For example, Fig. 2 shows the evaluation of $g(x)$ in the [-2, 2] interval for a set of default coefficient parameters: a=0, b=1, c=1, d=0. The coefficient parameters are substituted in equation (12) to generate the bit string.

Then the function $g(x)$ is sampled n_b times, where n_b is the number of bits required to represent the solution. If the $g(x)$ value is positive, the corresponding bit is set to 1. Otherwise it is set to 0. A bit is generated for each interval evaluated,

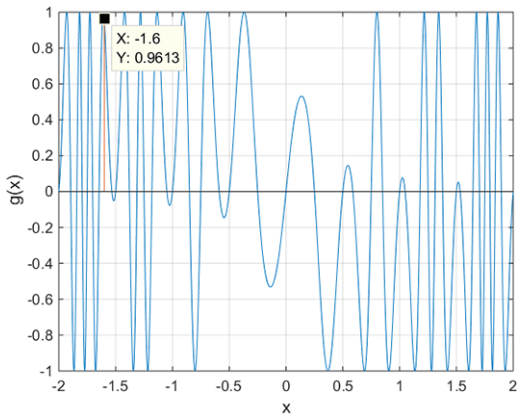


Fig. 2. Angle modulation function $g(x)$ for default parameters a=0, b=1, c=1, and d=0

so that each set of coefficient parameters that composes X_i has a Xb_i bit string with it.

For example, to generate 10 bits from Fig. 2, we define 10 equal separated intervals: x_j= (-1.6, -1.2, -0.8, -0.4, 0, 0.4, 0.8, 1.2, 1.6, 2). Then we evaluate $g(x)$ for x_1=-1.6, and as Fig. 2 shows, the $g(x)$ value is positive, so the first bit of the particle is set to 1. This process is repeated for all the remaining values of x.

Once we have sampled all the values, the whole bit string is generated Xb_i=(1 0 0 1 0 0 1 1 0 0). Xb_i represents one of the possible solutions to the discrete problem, then, it is evaluated to assign a fitness value.

AMPSO updates the velocity v_{id} using equation (13) and position x_{id} using equation (14) of X_i according to conventional PSO [12]:

$$v_{id} = wv_{id} + c_1 r_{1d}(p_{id} - x_{id}) + c_2 r_{2d}(p_{gd} - x_{id}), \quad (13)$$

$$x_{id} = x_{id} + v_{id}, \quad (14)$$

where c_1 and c_2 are positive constants used to scale the contribution of the cognitive and social components. r_1 and r_2 are vectors of random values in the range [0, 1] which are sampled from a uniform distribution and per dimension.

AMPSO reduces a high dimensional bit string to a four-dimensional vector. Algorithm 1 describes AMPSO for maximizing goodness.

Algorithm 1: AMPSO

Data: The equal intervals x_j to sample generating function, the swarm size S, the initial values of the four coefficients a, b, c and d

Result: The best solution Pb_g, evaluation of the best Pb_g in the fitness function $f(Pb_g)$

```
Initialize position vector X_i, intervals vector x_j, velocity vector V_i, memory vector P_i = X_i;
repeat
    for each particle i = 1 to number of particles in swarm S do
        Function Generate_bit_string_Xbi();
        Evaluate Xb_i using the objective function;
        Let f(X_i)=f(Xb_i);
        Function Generate_bit_string_Pbi();
        Evaluate Pb_i using the objective function;
        Let f(P_i)=f(Pb_i);
        if f(Xb_i) > f(Pb_i) then
            for d = 1 to 4 do
                p_id = x_id;
            end
        end
        g=i;
        for each particle j = 1 to number of particles in swarm S do
            Function Generate_bit_string_Pbj();
            Evaluate Pb_j using the objective function;
            Let f(P_j)=f(Pb_j);
            Function Generate_bit_string_Pbg();
            Evaluate Pb_g using the objective function;
            Let f(P_g)=f(Pb_g);
            if f(Pb_j) > f(Pb_g) then
                g=j;
            end
        end
        for d = 1 to 4 do
            Update velocity according to the equation (13);
            v_id ∈ (−V_max, V_max);
            Update position according to the equation (14);
        end
    end
until stopping condition(s) satisfied;
```

4 Angle Modulated Particle Swarm Optimization Algorithm to Resolve the Spectrum Assignment Problem

When a scenario (or snapshot) is analyzed using algorithm 3, Xb_i specifies a potential solution to solve the SA problem; that is, the set of secondary links that may coexist with the primary links in area A to achieve the maximum throughput, subject to QoS constraints for primary-secondary networks.

In algorithm 3, we include two new vectors: X_i' and P_i'. Each Xb_i has an X_i' which holds the candidate primary channels for the chosen secondary links. Take for example the snapshot illustrated in Fig. 3. Particle Xb_i indicates that secondary links 1, 3, 5, 6, 7, and 8 are

Algorithm 2: AMPSO, Functions

```
Function Generate_bit_string_Xbi()
    for j=1 to n_b do
        if g(X_i,x_j) ≥ 0 then
            xb_ij=1;
        else
            xb_ij=0;
        end
    end
End
Function Generate_bit_string_Pbi()
    for j=1 to n_b do
        if g(P_i,x_j) ≥ 0 then
            pb_ij=1;
        else
            pb_ij=0;
        end
    end
End
Function Generate_bit_string_Pbj()
    for k=1 to n_b do
        if g(P_j,x_k) ≥ 0 then
            pb_jk=1;
        else
            pb_jk=0;
        end
    end
End
Function Generate_bit_string_Pbg()
    for k=1 to n_b do
        if g(P_g,x_k) ≥ 0 then
            pb_gk=1;
        else
            pb_gk=0;
        end
    end
End
```

selected as a part of the solution; hence, X'_i is the channel allocation for those chosen secondary links. Consequently, P'_i keeps the best channels allocations find so far for Pb_i.

In contrast, the spectrum status vector holds the channel allocation for the primary links, so that, spectrum status vector is kept fixed through search. Mapping of X'_i and spectrum status

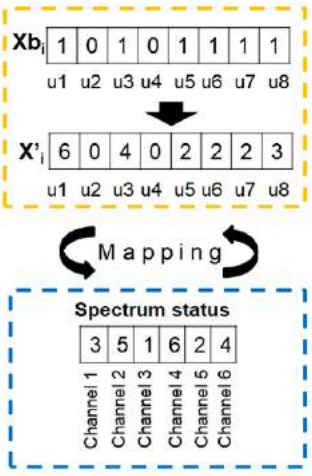


Fig. 3. Representation of particles for a given spectrum sharing access

provides the potential channels to share among the PUs and SUs. From the example in Fig. 3, channel 2 is exploited concurrently by primary link 5 and the secondary links 5, 6, and 7.

Once the bit strings Xb_i and Pb_i are generated as indicated in STEP 8 and STEP 9, the SINR levels for candidate SUs in Xb_i and PUs in the spectrum status vector are calculated. This is done by mapping each element of X'_i to its corresponding channel in the spectrum status vector. Also, the SINR levels for the best candidate SUs in Pb_i and PUs in the spectrum status vector are computed. Those SINR levels are calculated by using equations (1) and (2). In STEP 12, if SINR restrictions in equations (6) - (7) are achieved for SUs and PUs, Xb_i and Pb_i are feasible solutions. Then their fitness values are calculated as shown in equation (5). Otherwise, if SINR levels are not achieved, Xb_i and Pb_i are infeasible solutions that are penalized by setting their fitness values to zero.

From STEP 14 to STEP 19, the process of finding the best set of secondary links so far by the i-th particle is performed. Consequently, the best position and the best channel allocation are kept in those steps.

Another search process is performed from STEP 21 to STEP 31 to search for the best performer in the swarm.

Algorithm 3: AMPSO to resolve the spectrum assignment problem

Data: The total number of secondary links Sl, the total number of primary links Pl, the SINR thresholds γ = α, the swarm size S, the set of primary channels PC, the number of iterations T_{max}, the equal intervals x_j to sample generating function, the coefficients a, b, c and d

Result: The maximum data rate in the system $f(Pb_g)$, the set of selected secondary links Pb_g, the channel allocation for primary links in vector spectrum status, the best channel allocation for secondary links $P'g$

Locate randomly the total number of secondary links Sl and the total number of primary links Pl over the coverage area A;
Initialize randomly velocity vector V_i where $v_{id} \in (-V_{max}, V_{max})$. Set P_i = X_i;
Let coincide the personal best channel allocation vector P_i' and candidate channel allocation vector X_i';
Initialize randomly vector spectrum status with values from 1 to Pl;
Initialize position vector X_i where $x_{id} \in$ (-1, 1);
repeat
  for *each particle i = 1 to number of particles in swarm S* **do**
    Function *Generate_bit_string_Xbi()*;
    Function *Generate_bit_string_Pbi()*;
    Compute SINR at SUs and PUs by mapping X_i' and spectrum status;
    Compute SINR at SUs and PUs by mapping P_i' and spectrum status;
    Evaluate Xb_i and Pb_i at the fitness function at (5) and restrictions from (6) to (11);
    Let $f(P_i)$=$f(Pb_i)$ and $f(X_i)$=$f(Xb_i)$;
    if *f(Xb_i) > f(Pb_i)* **then**
      Perform from STEP 11 to STEP 13 from the algorithm (1);
      for *j=1 to* n_b **do**
        p'_{ij} = x'_{ij}
      end
    end
    g=i;
    for *each particle j = 1 to number of particles in swarm S* **do**
      Function *Generate_bit_string_Pbj()*;
      Function *Generate_bit_string_Pbg()*;
      Compute SINR at SUs and PUs by mapping P_j' and spectrum status;
      Compute SINR at SUs and PUs by mapping P_g' and spectrum status;
      Evaluate Pb_j and Pb_g at the fitness function at (5) and restrictions from (6) to (11);
      Let $f(P_g)$=$f(Pb_g)$ and $f(P_j)$=$f(Pb_j)$;
      if *f(Pb_j) > f(Pb_g)* **then**
        g=j;
      end
    end
    Perform from STEP 27 to STEP 31 from the algorithm (1);
    for *k=1 to* n_b **do**
      Allocate randomly a new channel to x'_{ik} from the set of primary channels PC;
    end
  end
until *number of iterations* $< T_{max}$;

The position and velocity of i-th particle are updated as shown in STEP 32.

The loop from STEP 33 to STEP 35, updates X_i'. The set of primary channels PC equals the channels in spectrum status vector. For example, from Fig. 3, they are five channels in spectrum status vector, so, $PC = \{1, 2, 3, 4, 5\}$.

Finally, in STEP 37, algorithm (3) repeats the above process until the maximum number of iterations T_{max} is met. Then, the solution of the problem is Pb_g which is the set of selected secondary links that maximize the throughput $f(Pb_g)$ with the primary links deployed in the area. Those selected secondary links also satisfy the QoS constraints, needed to keep the interference to a tolerable level for both secondary links and primary links.

5 Experimental Evaluation

In the following subsections, we present the scenario conditions to analyze AMPSO in the HetNet with SS approach. Then, we show the results obtained by the AMPSO for maximum throughput. For comparison, the SCPSO algorithm, the MBPSO algorithm [35], and the ModBPSO algorithm [34] are also included to solve the SA problem. Finally, we perform the Wilcoxon signed ranks and the sign test for multiple comparisons to confirm whether the AMPSO offers a significant improvement, or not, over the remaining BPSO variants.

5.1 Experimental Condition

We consider the downlink analysis of Fig. 1, characterized by a fixed deployment of primary links and a random deployment of secondary links in a 5000 m x 5000 m grid. An experiment is the combination of Pl, Sl, and SINR thresholds (γ = α). For each experiment, a BPSO variant is run for 500 independent instances (snapshots of random secondary links location and fixed primary links location). We incrementally vary the number of secondary links in the area, i.e., in step sizes of 10 secondary links. By doing that, interference rises gradually. The QoS requirement of 4 dB represents the less challenging scenario for the BPSO variants.

Therefore, most of the SUs deployed in the area will achieve that SINR threshold, i.e., most of them will be selected by the BPSO variants. On the other hand, the SINR threshold of 10 dB has a medium complexity for the BPSO variants. That means that some SUs may be able to be above the SINR threshold of 10 dB. Finally, the SINR threshold of 14 dB is the most challenging scenario for the BPSO variants due to the high QoS requirement. As more primary and secondary links are in the coverage area, the interference can rise to a harmful level. Then it is more challenging for the BPSO variants to leverage it up to a tolerable level. At this point, the task of selecting secondary users is vital since it is the strategy that the BPSO variants apply to cope with interference.

In regards to the HetNet, the femto-user is set to a maximum radius of 30 m away (for minimizing attenuation due to loss path) from the FBS; whereas, the macro-user is deployed 1000 m away from the MBS. We assume that secondary links and primary links employ unit transmission power and homogeneous traffic. Multipath and shadow fading are not considered for the SINR calculation. The number of channels to share depends on the number of primary links deployed in the area.

Table 1 and Table 2 show the parameters used for the BPSO variants and the experiments respectively. Increasing the number of primary and secondary links in the HetNet under different QoS requirements (4, 10, and 14 dB) result in increasing the complexity to find a good solution for every BPSO variant.

For instance, cognitive factor c_1, social factor c_2, socio-cognitive factor c_3, inertia weights w^1 and w which are parameters for SCPSO were set as suggested in the study [7]. The lower bound w_{min} and the upper bound w_{max} which are parameters of MBPSO were set as proposed in [35]. Concerning ModBPSO, the mutation rate r_{mu} is set as suggested in [34]. Consequently, we set the parameter maximum velocity V_{max} as suggested in [13].

The simulation methodology is in Fig. 4. Once we set the parameters for a BPSO variant and

experiment, the admission control and channel allocation algorithm based on a BPSO variant generates a snapshot of a HetNet scenario, and then the BPSO variant is run to solve equations (5) - (11). After the admission control and channel allocation algorithm based on a BPSO variant finishes its execution, it computes the maximum throughput for that snapshot. If the admission control and channel allocation algorithm based on a BPSO variant achieves the total sample snapshots to analyze, it selects the sample snapshot with the highest throughput.

Table 1. Parameters used for experiments

Parameters	Values
Number of secondary links Sl	10:100:10
Number of primary links Pl	6, 12, 24
Runs	500
SINR thresholds γ = α	4, 10, 14 dB
Channel bandwidth B	20 MHz

Table 2. Parameters used for the BPSO variants

Parameters	Values
Swarm size S	40
Maximum number of iterations T_{max}	100
Cognitive, social factors c_1, c_2	2, 2
Socio-cognitive factor c_3 (for SCPSO)	12
w_{max}, w_{min} (for MBPSO)	1.4, 0.1
$Iter_{max}$ (for MBPSO)	20
Inertia weight w^1 (for SCPSO)	0.9
Mutation rate r_{mu} (for ModBPSO)	0.02
Inertia weight w	0.721
Maximum velocity V_{max}	[-6, 6]

5.2 Experimental Results

In Figs. 5a, 5c, and 5e, we show the best solutions found by the BPSO variants for the HetNet when Pl = 6 at γ, α = [4, 10, 14] dB. The best solutions found by the AMPSO outperform the ones found by the remaining BPSO variants, in the range of 10-60 SUs. The ModBPSO comes next, however, for higher values of γ, α (high QoS requirements); it cannot find a solution.

In Figs. 5b, 5d, and 5f, we show the average maximum data rate, i.e., we average the results over all samples in the experiment. While the curves of ModBPSO, MBPSO and SCPSO come down in the range 30 - 40 SUs deployed in the area, AMPSO keeps almost a constant throughput with the highest data rates. When the other BPSO variants fail to find a solution, as in Fig. 5f which is the most challenging scenario, AMPSO is able not only in finding a solution but also in offering the one with the highest data rate.

Concerning the experiment when Pl = 12 at γ, α = [4, 10, 14] dB, the best solutions found and the average data rate are shown in Figs. 6a – 6f. In Figs. 6a, 6c, and 6e, AMPSO still outperforming the other BPSO variants. In Fig. 6e, while MBPSO, ModBPSO, and SCPSO could not find a solution in the range 80-100 SUs, AMPSO can find it. Figs. 6b, 6d, and 6f show that AMPSO produces, on average better solutions and it can find solutions even in the most challenging scenario as plotted in Fig. 6f.

Figs. 7a, 7c, and 7e are the best solutions found by the different versions of BPSO when Pl = 24 at γ, α = [4, 10, 14] dB. In this context, AMPSO performs better than the other BPSO variants, even when the QoS requirement is the highest, as in Fig. 7e. From that plot, we observe that AMPSO can find a solution when the others fail, especially in the range of 80 - 100 SUs. Figs. 7b, 7d, and 7f shows that AMPSO significantly outperforms the other BPSO methods in average throughput, finding solutions when the other BPSO methods cannot offer one as in Fig. 7f.

5.3 Use of Nonparametric Statistics for Comparing the Results

We perform the Wilcoxon signed ranks and the sign test for multiple comparisons to confirm whether AMPSO offers a significant improvement, or not, over the remaining BPSO variants for the HetNet with SS approach. Among the experiments, we are particularly interested in ones when Sl=100. We summarize the results obtained for each experiment and BPSO variant in Table 3. The performance measure is the average fitness (throughput).

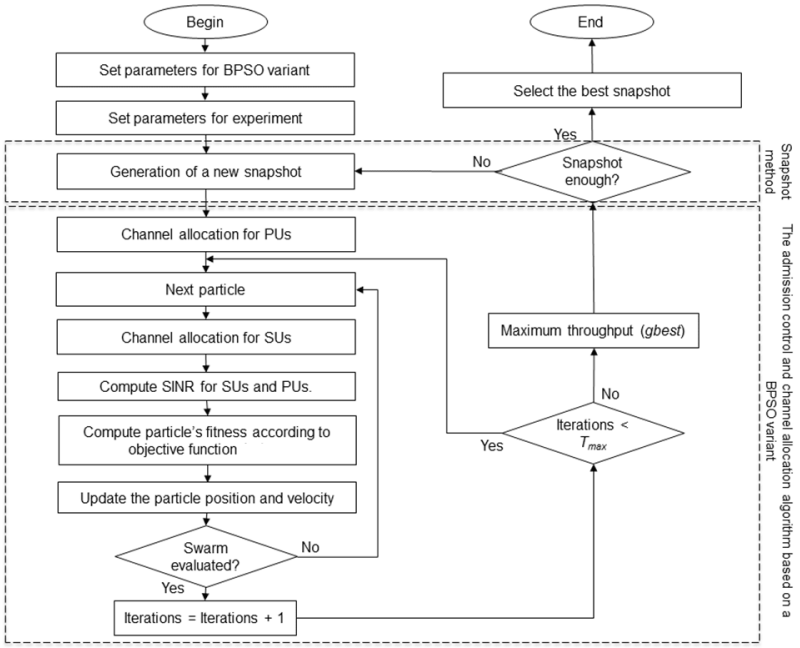


Fig. 4. Simulation methodology

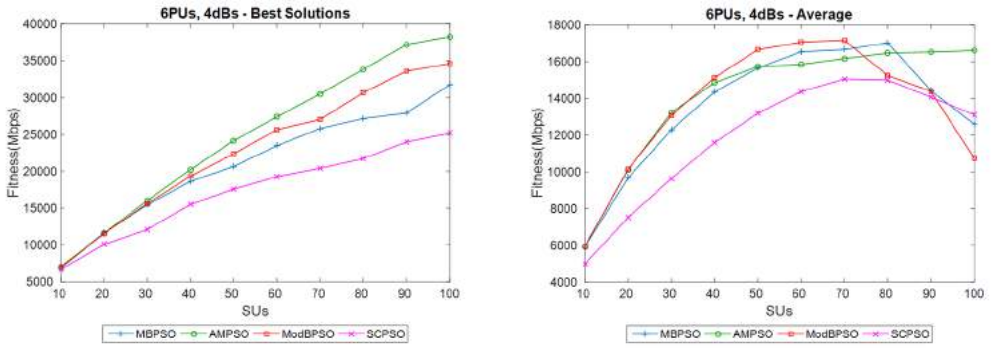
Table 3. Average fitness obtained in the different experiments

Experiment	AMPSO	MBPSO	ModBPSO	SCPSO
6 PUs, 100 SUs, 4 dB	16610.29	12599.9	10737.3	13109.61
6 PUs, 100 SUs, 10 dB	8913.5	440.2	617.41	832.44
6 PUs, 100 SUs, 14 dB	4992.05	80.9	0	114.26
12 PUs, 100 SUs, 4 dB	20519.99	19506.12	17721.69	18941.38
12 PUs, 100 SUs, 10 dB	10769.62	594.31	524.72	852.07
12 PUs, 100 SUs, 14 dB	5957.41	0	0	0
24 PUs, 100 SUs, 4 dB	24585.18	26138.38	24820.07	24787.52
24 PUs, 100 SUs, 10 dB	13151.15	947.02	1113.33	1409.48
24 PUs, 100 SUs, 14 dB	7302.51	0	0	0

Firstly, we present a comparative study on AMPSO performance and the remaining BPSO variants through pairwise comparisons. We apply the Wilcoxon signed ranks since it is a safe and robust nonparametric test for pairwise statistical comparisons. Also, the outliers (exceptionally good/bad performances) have less effect on it [8]. Table 4 summarizes the results of applying it, displaying the sum of rankings obtained in each comparison and the p-value associated.

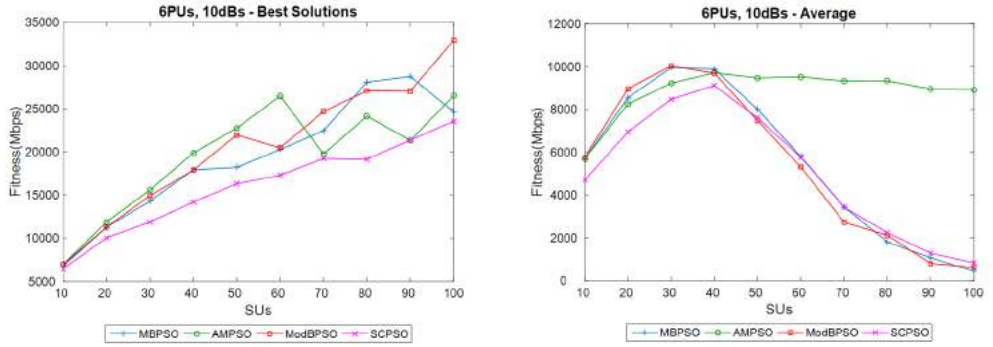
Table 4. AMPSO shows improvement over SCPSO, ModBPSO, and MBPSO, with a level of significance α=0.05

AMPSO vs.	R^+	R^-	p-value
SCPSO	44	1	0.011
ModBPSO	44	1	0.011
MBPSO	43	2	0.015



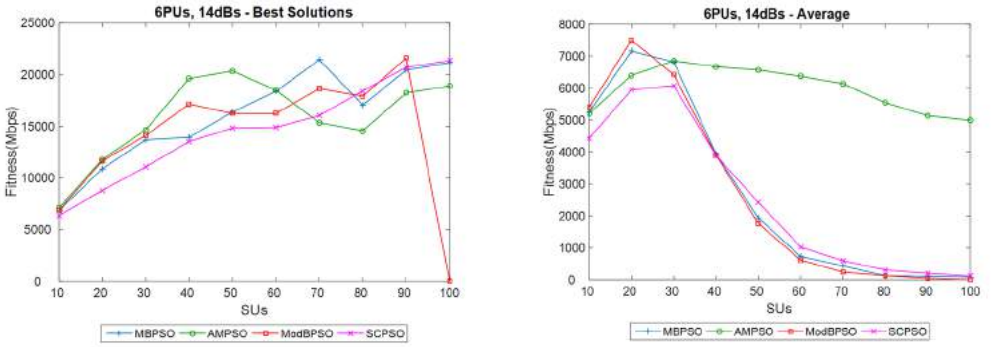
(a) best solutions γ, α = 4 dB

(b) average γ, α = 4 dB



(c) best solutions γ, α = 10 dB

(d) average γ, α = 10 dB



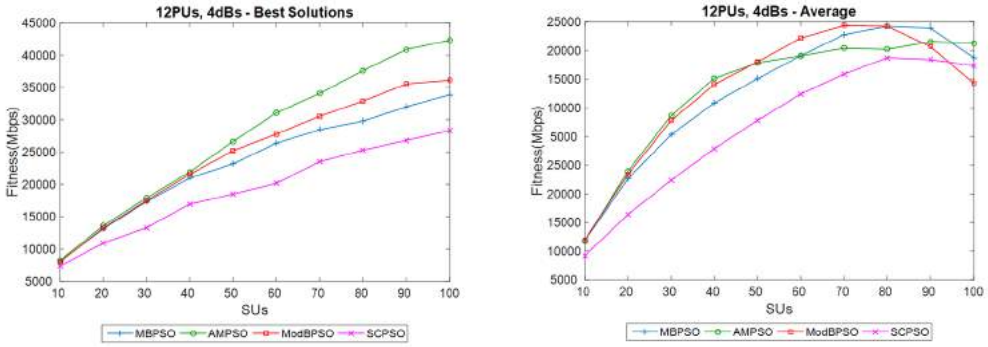
(e) best solutions γ, α = 14 dB

(f) average γ, α = 14 dB

Fig. 5. System throughput of the BPSO variants when Pl = 6 at γ, α = [4, 10, 14] dB

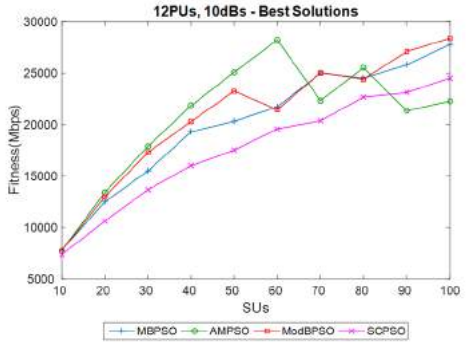
As Table 4 states, AMPSO shows a significant improvement over SCPSO, ModBPSO, and MBPSO, with a level of significance α=0.05. Since p-values are less than α=0.05, we reject the null hypothesis.

The null hypothesis (H_0) is stating no effect or no difference, whereas the alternative hypothesis (H_1) represents an effect or a difference (significant differences between algorithms) [8]. Labeling AMPSO as our control algorithm, the sign test

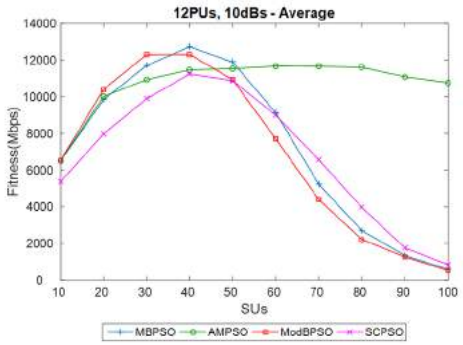


(a) best solutions γ, α = 4 dB

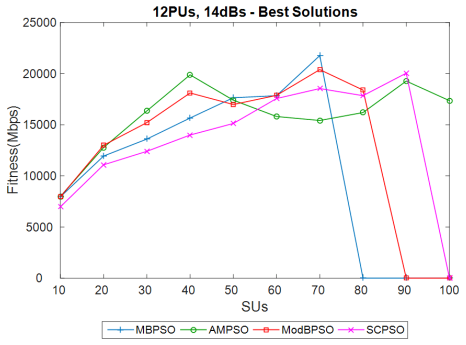
(b) average γ, α = 4 dB



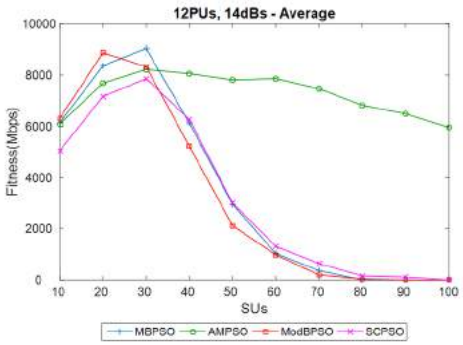
(c) best solutions γ, α = 10 dB



(d) average γ, α = 10 dB



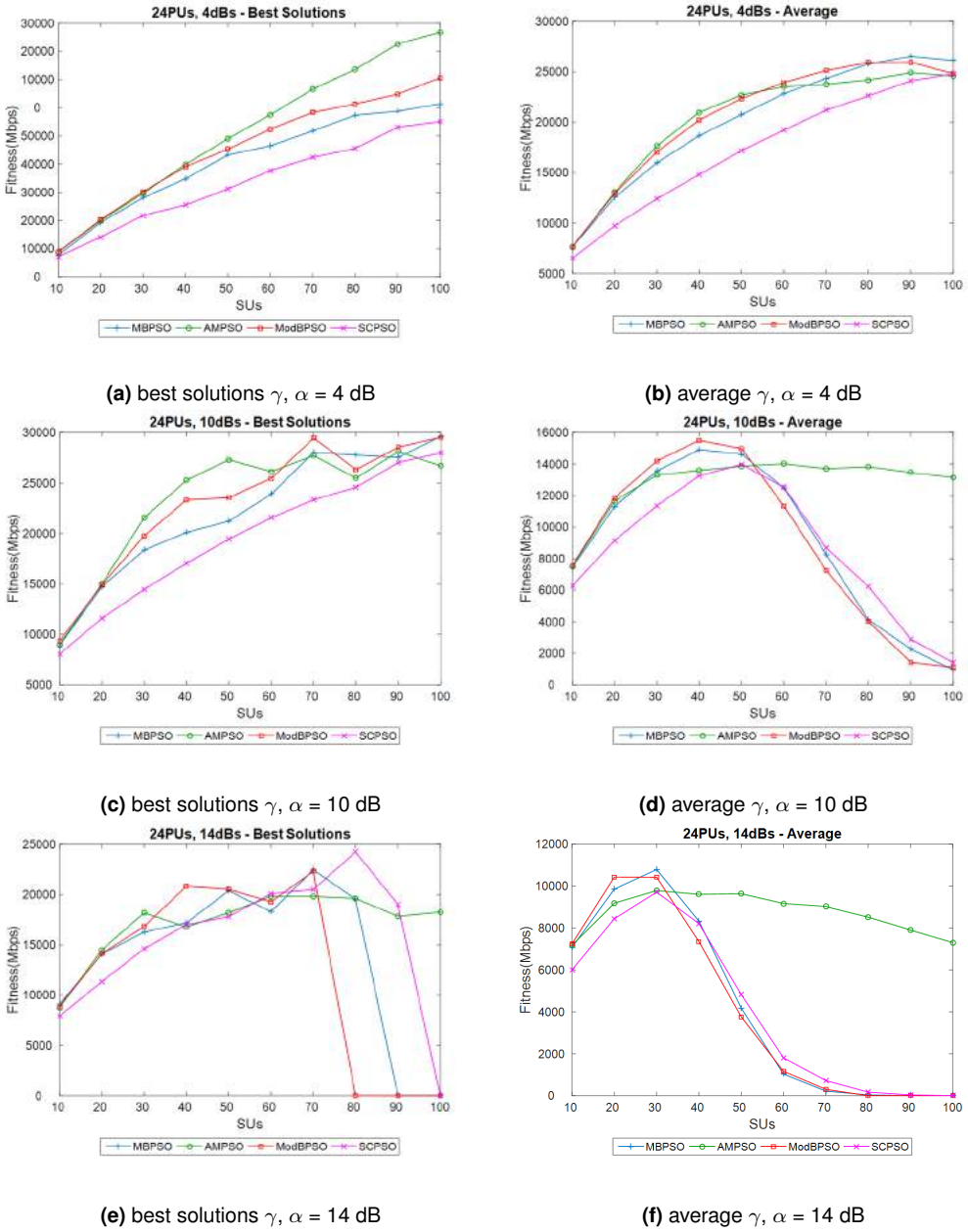
(e) best solutions γ, α = 14 dB



(f) average γ, α = 14 dB

Fig. 6. System throughput of the BPSO variants when Pl = 12 at γ, α = [4, 10, 14] dB

for multiple comparisons highlights those BPSO variants whose performances are statistically different when compared with the control algorithm. We apply the procedure described in [9]. Table 5 summarizes the results with two levels of significance α=0.1 and α=0.05. Let M_1 be the median response of a sample of results of the control method and M_j be the median response of a sample of results of the j-th algorithm. Let our hypotheses be H_0: $M_j \geq M_1$ and H_1: $M_j < M_1$;



(a) best solutions γ, α = 4 dB

(b) average γ, α = 4 dB

(c) best solutions γ, α = 10 dB

(d) average γ, α = 10 dB

(e) best solutions γ, α = 14 dB

(f) average γ, α = 14 dB

Fig. 7. System throughput of the BPSO variants when Pl = 24 at γ, α = [4, 10, 14] dB

that is, our control algorithm AMPSO is significantly better than the remaining algorithms.

Reference to Table A.1 from [9] for (k-1)=3 and n=9 reveals that the critical values are 1 (α=0.1) and 0 (α=0.05).

Then, since the number of pluses in MBPSO, ModBPSO and SCPSO is less than or equal to the critical values, the AMPSO has a better performance than them.

Table 5. Multiple sign test using AMPSO as the control algorithm

Experiment	AMPSO 1 (control)	MBPSO 2	ModBPSO 3	SCPSO 4
6 PUs, 100 SUs, 4 dB	16610.29	12599.9 (-)	10737.3 (-)	13109.61 (-)
6 PUs, 100 SUs, 10 dB	8913.5	440.2 (-)	617.41 (-)	832.44 (-)
6 PUs, 100 SUs, 14 dB	4992.05	80.9 (-)	0 (-)	114.26 (-)
12 PUs, 100 SUs, 4 dB	20519.99	19506.12 (-)	17721.69 (-)	18941.38 (-)
12 PUs, 100 SUs, 10 dB	10769.62	594.31 (-)	524.72 (-)	852.07 (-)
12 PUs, 100 SUs, 14 dB	5957.41	0 (-)	0 (-)	0 (-)
24 PUs, 100 SUs, 4 dB	24585.18	26138.38 (+)	24820.07 (+)	24787.52 (+)
24 PUs, 100 SUs, 10 dB	13151.15	947.02 (-)	1113.33 (-)	1409.48 (-)
24 PUs, 100 SUs, 14 dB	7302.51	0 (-)	0 (-)	0 (-)
Number of pluses		1	1	1
Number of minuses		8	8	8
Critical value at α=0.1		1	1	1
Critical value at α=0.05		0	0	0

6 Discussion

Contrasting the results among the BPSO variants in Sect. 5.2 for maximum throughput, AMPSO performed better. We applied statistical tests to confirm whether AMPSO offers a significant improvement over the BPSO variants for the given experiments. Firstly, we performed a pairwise statistical comparison using the Wilcoxon signed ranks test, confirming that AMPSO outperformed the remaining BPSO variants. In [10] is mentioned that the smaller the p-value, the stronger the evidence against H_0. In this context, we obtained small p-values (less than 0.05) when we applied the Wilcoxon signed ranks test, which indicated strong evidence against H_0. Secondly, we performed multiple comparisons with AMPSO as the control algorithm to determine which of the other algorithms exhibit a different performance. Multiple sign test, helped us to confirm that AMPSO outperformed the BPSO variants. We used significance levels α of 0.05 and 0.1 (95% and 90% certainty that there indeed is a significant difference).

The experimental results and the nonparametric tests, confirm that in the optimization problem posed in equations (5) - (11), AMPSO produces favorable results. AMPSO is suited for complex scenarios, i.e., scenarios with high QoS requirements and many SUs and PUs deployed in the service area. Then, the non-uniform frequency distribution of binary solutions in the AMPSO search space [16] is advantageous in the SA problem due to the generating function g. As described in [16], the generating function g leads to more than one permutation of the coefficients generating the same binary solution. They also mention that the most common solutions in the AMPSO search space are the ones that contain repetitive patterns. This trend is advantageous in problems whose optimal solutions include repetitive patterns because those solutions are common in the AMPSO search space [16]. Then in the context of the SA problem posed in equations (5) - (11), the repetitive patterns in the candidate solutions give an advantage to AMPSO.

For simple scenarios i.e., scenarios with low QoS requirements (γ, α = 4 dB), the SCPSO should be used. As more SUs are deployed in the area, it is more challenging for SCPSO, to select SUs to share a primary channel.

In contrast, MBPSO is unsuitable for complex scenarios, i.e., scenarios with high QoS requirements and many SUs. This is due to the decreasing inertia weight scheme that MBPSO uses to search for a solution. Through iterations, if fitness does not improve w increases, to stimulate exploration; otherwise, when fitness improves, w takes a small value to exploit a region where MBPSO has found a candidate solution. However, in the binary case, as the work in [18] suggests, a smaller inertia weight enhances the exploration

capability while a larger inertia weight encourages exploitation. In most cases, increasing inertia weight is favorably for the discrete PSO.

Also, from the simulation, ModBPSO had the worst performance. Although the v-shaped transfer function has been proved to have significant advantages [21], in the SA problem, it does not provide a high performance.

The objective function in (5) is the metric to measure how the HetNet efficiently uses the spectrum. Several methods exist to measure the efficiency of spectrum use, and no single measure works for all scenarios [28]. In this context from results in Sect. 5.2, maximum throughput is a well-suited metric to measure spectrum usage in scenarios with low dense cell deployments (macro and femtocells) at different QoS thresholds. Successful communications are ensured for PUs and those SUs that simultaneously exploit a channel through the QoS thresholds. Estimating the efficiency of a primary system (the set of PUs) will help to determine if it could be shared [28].

7 Conclusion and Future Work

We consider the SS paradigm in a HetNet to propose a solution to the SA problem, maximizing network throughput when one or more SUs exploit a channel simultaneously with the PU, satisfying QoS requirements on SINR. Assuming that SS will impact future next-generation cellular networks, we consider primary and secondary systems operating in that frequency band. We handle the SA problem in scenarios with high QoS requirements and many SUs and PUs deployed in an area. Under such scenarios, the candidate solutions are high-dimensional bit strings. The search for a good solution is challenging as the QoS requirements increase. To address this challenge, we apply the AMPSO metaheuristic due to its ability to handle higher-dimensional problems. Then the AMPSO results are compared with the MBPSO, the SCPSO, and the ModBPSO.

From the simulation, AMPSO is suited for complex scenarios i.e., scenarios with high QoS requirements and a large number of SUs and PUs deployed in the service area. Then, our results confirm the AMPSO's ability to handle problems defined in larger and more abstract dimensions by combining PSO with angle modulation.

For simple scenarios i.e., scenarios with low QoS requirements (γ, α = 4 dB), the SCPSO should be used. Whereas ModBPSO is not suitable for the SA problem.

In future work, we plan to address the fairness in SUs when a channel is shared among SUs and a PU, i.e., that the SUs have the same opportunity to access spectrum to perform a communication. Also, it is planned to pose the SA problem as a multi-objective approach to maximize the data rate and the number of selected secondary links. Those objectives conflict due to the interference. Finally, we will include other components of HetNet as microcells and picocells. Those types of small cells vary in deployment location (outdoor/indoor), coverage, transmit power, and deployment configuration (planned/unplanned).

References

1. **Adedoyin, M. A., Falowo, O. E. (2020).** Combination of ultra-dense networks and other 5g enabling technologies: A survey. IEEE Access, Vol. 8, pp. 22893–22932.
2. **Andrews, J. G., Ganti, R. K., Haenggi, M., Jindal, N., Weber, S. (2010).** A primer on spatial modeling and analysis in wireless networks. IEEE Communications Magazine, Vol. 48, No. 11, pp. 156–163.
3. **Beltran, F. (2017).** Accelerating the introduction of spectrum sharing using market-based mechanisms. IEEE Communications Standards Magazine, Vol. 1, No. 3, pp. 66–72.
4. **Cave, M., Doyle, C., Webb, W. (2012).** Essentials of Modern Spectrum Management. Cambridge University Press, New York, NY, USA.
5. **Cheng, S., Ao, W. C., Tseng, F., Chen, K. (2012).** Design and analysis of downlink spectrum sharing in two-tier cognitive femto networks. IEEE Transactions on Vehicular Technology, Vol. 61, No. 5, pp. 2194–2207.
6. **Cisco (2020).** Cisco annual internet report (2018–2023) white paper. Last accessed 12 May 2021.

7. **Deep, K., Bansal, J. C. (2008).** A socio-cognitive particle swarm optimization for multi-dimensional knapsack problem. 2008 First International Conference on Emerging Trends in Engineering and Technology, pp. 355–360.
8. **Derrac, J., García, S., Molina, D., Herrera, F. (2011).** A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation, Vol. 1, No. 1, pp. 3–18.
9. **García, S., Fernández, A., Luengo, J., Herrera, F. (2010).** Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences, Vol. 180, No. 10, pp. 2044–2064. Special Issue on Intelligent Distributed Information Systems.
10. **García, S., Molina, D., Lozano, M., Herrera, F. (2008).** A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec'2005 special session on real parameter optimization. Journal of Heuristics, Vol. 15, No. 6, pp. 617.
11. **Gupta, M. S., Kumar, K. (2019).** Progression on spectrum sensing for cognitive radio networks: A survey, classification, challenges and future research issues. Journal of Network and Computer Applications, Vol. 143, pp. 47–76.
12. **Kennedy, J., Eberhart, R. (1995).** Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, volume 4, pp. 1942–1948 vol.4.
13. **Kennedy, J., Eberhart, R. C. (1997).** A discrete binary version of the particle swarm algorithm. 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, volume 5, pp. 4104–4108 vol.5.
14. **Kibria, M. G., Villardi, G. P., Nguyen, K., Ishizu, K., Kojima, F. (2017).** Heterogeneous networks in shared spectrum access communications. IEEE Journal on Selected Areas in Communications, Vol. 35, No. 1, pp. 145–158.
15. **Kour, H., Jha, R. K., Jain, S. (2018).** A comprehensive survey on spectrum sharing: Architecture, energy efficiency and security issues. Journal of Network and Computer Applications, Vol. 103, pp. 29–57.
16. **Leonard, B. J., Engelbrecht, A. P. (2015).** Frequency distribution of candidate solutions in angle modulated particle swarms. 2015 IEEE Symposium Series on Computational Intelligence, pp. 251–258.
17. **Liu, C., Tsai, H. (2017).** Traffic management for heterogeneous networks with opportunistic unlicensed spectrum sharing. IEEE Transactions on Wireless Communications, Vol. 16, No. 9, pp. 5717–5731.
18. **Liu, J., Mei, Y., Li, X. (2016).** An analysis of the inertia weight parameter for binary particle swarm optimization. IEEE Transactions on Evolutionary Computation, Vol. 20, No. 5, pp. 666–681.
19. **Martinez, E., Andrade, A. G., Martínez-Vargas, A., Galaviz, G. (2016).** Optimización binaria por cúmulo de partículas con memoria (MBPSO) para resolver un problema de espectro compartido. Computación y Sistemas, Vol. 20, pp. 153–168.
20. **Martínez-Vargas, A., Andrade, Á. G., Sepúlveda, R., Montiel-Ross, O. (2014).** An admission control and channel allocation algorithm based on particle swarm optimization for cognitive cellular networks. In **Castillo, O., Melin, P., Pedrycz, W., Kacprzyk, J.**, editors, Recent Advances on Hybrid Approaches for Designing Intelligent Systems. Springer International Publishing, Cham, pp. 151–162.
21. **Mirjalili, S., Lewis, A. (2013).** S-shaped versus v-shaped transfer functions for binary particle swarm optimization. Swarm and Evolutionary Computation, Vol. 9, pp. 1–14.
22. **Olivas, F., Valdez, F., Melin, P., Sombra, A., Castillo, O. (2019).** Interval type-2 fuzzy logic for dynamic parameter adaptation in a modified gravitational search algorithm. Information Sciences, Vol. 476, pp. 159–175.
23. **Pampara, G., Franken, N., Engelbrecht, A. P. (2005).** Combining particle swarm optimisation with angle modulation to solve binary problems. 2005 IEEE Congress on Evolutionary Computation, volume 1, pp. 89–96 Vol.1.
24. **Peha, J. M. (2005).** Approaches to spectrum sharing. IEEE Communications Magazine, Vol. 43, No. 2, pp. 10–12.
25. **Roeva, O., Zoteva, D., Castillo, O. (2021).** Joint set-up of parameters in genetic algorithms and the artificial bee colony algorithm: an approach for

cultivation process modelling. Soft Comput., Vol. 25, No. 3, pp. 2015–2038.

26. **Rohoden, K., Estrada, R., Otrok, H., Dziong, Z. (2020).** Evolutionary game theoretical model for stable femtocells' clusters formation in hetnets. Computer Communications, Vol. 161, pp. 266–278.
27. **Ruby, D., Vijayalakshmi, M., Kannan, A. (2019).** Intelligent relay selection and spectrum sharing techniques for cognitive radio networks. Cluster Computing, Vol. 22, No. 5, pp. 10537–10548.
28. **Rysavy, P. (2014).** Challenges and considerations in defining spectrum efficiency. Proceedings of the IEEE, Vol. 102, No. 3, pp. 386–392.
29. **Sathya, V., Kala, S. M., Bhupeshra, S., Tamma, B. R. (2021).** Raptap: a socio-inspired approach to resource allocation and interference management in dense small cells. Wireless Networks, Vol. 27, No. 1, pp. 441–464.
30. **Talbi, E.-G. (2009).** Metaheuristics: From Design to Implementation. Wiley Publishing.
31. **Talukdar, B., Kumar, D., Hoque, S., Arif, W. (2021).** Cooperative spectrum sensing in energy harvesting cognitive radio networks under diverse distribution models. In **Mandloi, M., Gurjar, D., Pattanayak, P., Nguyen, H.**, editors, 5G and Beyond Wireless Systems: PHY Layer Perspective. Springer Singapore, Singapore, pp. 245–272.
32. **Tragos, E. Z., Zeadally, S., Fragkiadakis, A. G., Siris, V. A. (2013).** Spectrum assignment in cognitive radio networks: A comprehensive survey. IEEE Communications Surveys Tutorials, Vol. 15, No. 3, pp. 1108–1135.
33. **Valdez, F., Vazquez, J. C., Melin, P., Castillo, O. (2017).** Comparative study of the use of fuzzy logic in improving particle swarm optimization variants for mathematical functions using co-evolution. Applied Soft Computing, Vol. 52, pp. 1070–1083.
34. **Yang, J., Zhang, H., Ling, Y., Pan, C., Sun, W. (2014).** Task allocation for wireless sensor network using modified binary particle swarm optimization. IEEE Sensors Journal, Vol. 14, No. 3, pp. 882–892.
35. **Zhen Ji, Tao Tian, Shan He, Zexuan Zhu (2012).** A memory binary particle swarm optimization. 2012 IEEE Congress on Evolutionary Computation, pp. 1–5.
36. **Zia, K., Javed, N., Sial, M. N., Ahmed, S., Pirzada, A. A., Pervez, F. (2019).** A distributed multi-agent rl-based autonomous spectrum allocation scheme in d2d enabled multi-tier hetnets. IEEE Access, Vol. 7, pp. 6733–6745.

Article received on 29/06/2021; accepted on 16/11/2021.
Corresponding author is Anabel Martínez-Vargas.

ISSN 2007-9737

A Mamdani Type-Fuzzy Inference - Alignment Matrix Method for Evaluation of Competencies Acquired by Students Enrolling at the Mexican Higher Middle Education System I: Formulation and Explanation Based on Simulation, and a Real but Incomplete Data Set

Cecilia Leal-Ramírez[1], Héctor Alonso Echavarría-Heras[1]
Hajasya Maraay Romero-Escobar[2]

[1] Centro de Investigación Científica y de Estudios Superiores de Ensenada,
Departamento de Ecología,
Mexico

[2] Universidad Tecnológica de Tijuana,
Facultad de Energías Renovables y Ambiental,
Mexico

cleal@cicese.mx, heheras@icloud.com, hajasya@hotmail.com

Abstract. The last reforms at the Mexican educational system attempt to modify graduates' traditional profiles to enable them to investigate, develop and propose solutions to various problems, think more assertively, form opinions, interact with multidisciplinary groups of people, and ultimately assume a proactive role in the community. The implementation of curricula promoting the development of competencies becomes essential in achieving envisioned goals. The alignment of competencies to study plans and the evaluation of the development of related skills becomes critical at the scheme's implementation. Currently, relating duties mainly depends on the teacher's will, but the lack of a systematic protocol to achieve involved endeavors promotes designs based on subjectivity. On top of this, concurring student's performance grading scheme relies on a thorough alphabetic character rating approach, thereby masking objective scoring. Assessment at the Mexican educational system realm conceives upon criteria designed abroad. Thereby linking particularities are left behind at performing evaluation tasks. This contribution proposes a method based on a fuzzy inference approach that aims to provide a formal structure that allows teachers to achieve the alignment of competencies and evaluate their development by students relying on a quantitative paradigm. We offer a comprehensive display of the formalities of the proposed fuzzy method. Explanation of its functionality relies on both simulated and real data. The aim here is on addressing the student-teacher aspect of alignment and evaluation procedures. Extension to other precincts within the whole Mexican educational system also contemplates and intends to appear in further contributions.

Keywords. Alignment matrix, competencies, mathematical relation.

1 Introduction

Reforms the Mexican educational system that initiated in the last decade in Mexico address all levels and have in common a curriculum design promoting the development of competencies [1]. The ensuing strategy aims to transform education so that graduates investigate, develop and propose solutions to various problems, think more assertively, form opinions, interact with multidisciplinary teams and ultimately assume a proactive role in their communities [2]. In what follows the composite Mexican educational system could be also referred simply as educational system for short. So far, the educational system's protocol for the evaluation of student's performance has mainly mimicked international criteria. Subsequent assessments have so far revealed unfavorable results.

For example, the values reported by the International Program for the Evaluation of Students [3-6] demonstrate that since 2006 the output of the Mexican educational endeavors exhibited severe deficiencies, mainly in mathematics, science, and reading. Therefore, the procedures involved in the whole educational system should be revised, particularly those contributing to students' development of competencies and the concurring evaluation of their acquisition.

A competency describes as a composite of a coordinated and integrated knowledge set along with procedures and attitudes so that the individual can display the "know-how to do" and the "know-how to be" while performing professional practice [7]. From this, it follows that when individuals become competent, they can select, coordinate and efficiently mobilize a set of articulated and interrelated knowledge to solve problems in a specific context. Given this interpretation, competencies could conceive as complex skills that are difficult to conceptualize, especially concerning what it means to integrate them into practice because there is no unified theory for their implementation and evaluation. For example, the definition of attitudes and values, by their very nature, implies a noticeable subjective burden on the evaluating entities since they are intangible and dependent on human interpretation.

This aspect adds ambiguity or vagueness to the evaluation because it is a function of the experiences. There is an uncertainty load when a context conforms to vagueness, imprecision, and lack of data. Fortunately, techniques and instruments proposed by a Fuzzy Logic approach have efficiently contributed to modeling uncertain phenomena in educational sciences. Applications cover assessing student-centered learning [8]. Besides, fuzzy paradigms contributed classifying the student academic profile [9]. And fuzzy logic applications also appear for the cognitive diagnosis of the student [10], as well as, at evaluating student's attitude [11].

Currently, the Mexican education system structures a study program based on units, learning outcomes, and complementary activities, assigning each one of them a specific value relative to the whole program [12].

Ensuing percentage values express without specifying any mathematical relationship that allows evaluating the development of competencies in each activity and despite this, teachers must maintain indicators of skills acquired by each of the students. Besides, teachers do not have at hand a specified protocol that allows them to align the competencies with the unit's objectives, learning outcomes, and learning activities.

A lack of a systematized approach to achieve the alignment of competencies also makes it difficult to integrate their acquisition into the student’s assessment protocol. In addition, the evaluation scheme implemented at the technological branch of the Mexican higher middle education system constitutes impairing the implementation and evaluation of competencies. For example, the use of alphabetic characters to label grades describes an effort to deter discrimination scenarios [12]. But such an approach could mask the student's objective evaluation.

Also, in the current evaluation approach, a student does not "approve" or "fail" in the literal interpretation of the words but is instead declared “Competent" or "Not competent" [12]. Such an evaluation scheme makes it difficult to compare results on a quantitative basis. For a student, it is enough to obtain "Competent," which achieves when 70% of academic activities evaluate at least with the letter "S" (Sufficient) [12]. Since students know how the system operates, so they efficiently manage to know when they have reached 70% of their evaluation. This way, many students resolve not to participate more actively in in their educational task. In addition, students assume that even though a lack of commitment at the end of the period, they will not get a “failed," This characteristic of the educational system undermines all the efforts of teachers. And the fact that there is still no methodology based on a mathematical formalism to establish an alignment of the student's competencies with the learning activities reduces the effectiveness of the teacher's competencies own skills.

In the literature, it is possible to find models on the alignment and evaluation of acquired competencies. For instance, Biggs in [13] points out that the way to carry out a constructive alignment is by elaborating learning objectives,

which can be created by establishing the subject centered on the action. This identifies the task or product or, more generally, the accomplishment to be carried out. Vargas in [14] proposed a meta-model to assess the grade of acquisition of competencies. They established a relationship between competencies, a learning outcome, and an evaluation tool through a tree. The children's weight of the internal nodes has an arbitrary value between 0 and 1, and their sum is 1. Romero-Escobar in [12] proposed a model that establishes an alignment of the objectives of the learning activities with the competencies of the teacher and the student's competencies. The approach offers a procedure to construct a relevance matrix to indicate which competencies will be considered in the design of learning activities. However, the resulting alignment does not express the mathematical relationship between the competencies and the objectives of the learning activities mathematically. The most outstanding aspect of Romero-Escobar's work is conceiving the relevance matrix and the outline from which its formal structure can be arranged.

The approach that we offer in this work involves the development of a system composed of an alignment method, whereby means of a matrix on its mathematical conception, the teacher can make an alignment indicating the competencies and attributes formally integrated at the process of evaluation of learning activities.

The alignment is strictly a task consigned to the teachers; their criterion traces the student's path to achieve full development of competencies. Based on this integration, the system creates weight values rating the relative importance for the learning units, and activities contemplated in the study program. It is worth emphasizing that our approach undertakes using a mathematical formalism. A weight value will represent the teacher's criteria applied in said alignment, totally transparent and easy to accomplish. Proposed alignment formalization will also allow knowing the grade of development of the competencies acquired by the student or by the group.

In addition, our construct integrates a Mamdani type fuzzy inference model to evaluate the efficiency with which the student performs their activities through three elements, such as the knowledge acquired, the procedure developed and the attitude shown during the process, that is, extracted evidence when developing the learning activities. The teacher's mastery of the subject must create learning activities aligned with the acquisition of competencies [15-17] and provide the techniques to evaluate them.

Fuzzy systems based in fuzzy logic [18] are used in many scientific applications, they are classified into two types. Type 1 fuzzy systems can handle linguistic variables and expert reasoning and reproduce the knowledge of the systems to be controlled (e.g., [19]).

Type 2 fuzzy systems can model complex non-linear systems, achieving better performance from controllers designed under this approach (e.g., [18, 20, 21]).

The fuzzy logic systems of Type 2 by interval are a representation of reduced complexity of the fuzzy systems Type 2, it is the most used now (e.g., [22]). Uncertainty is inherent in the information regardless of the type of technique, treatment or any other factor relying in its use. Regarding the evaluation of competencies, fuzzy logic has been used to assess student responses (e.g., [23-26]) or to construct fuzzy rubrics (e.g., [27]).

Our Mamdani-typo-1 fuzzy inference system integrates three elements: knowledge, procedure and attitude, defined as linguistic variables to evaluate efficiency, which has not been done so far. To begin with, talking about attitude implies identifying the kind, we refer to [28]. It would be very difficult to integrate all kinds of attitudes in a single Type 1 fuzzy system, considering that elements such as knowledge and the procedure are to be also integrated in the evaluation of the efficiency. The system that we develop is also relevant because it attempts to adapt a first step by mathematically formalizing the alignment of competencies with learning activities in accordance with the objectives established by the study program.

Once the alignment is made, our approach provides a way to evaluate the grade of development of the competencies with transparency and with the certainty that the competences acquired by the student are those established in said alignment. In addition, the Mamdani type-1 fuzzy inference system evaluates the efficiency with which the competencies were developed in each of the learning activities,

integrating three fundamental elements that define the set of articulated and interrelated knowledge to solve problems in a specific context.

What remains of the paper organizes as follows. In Section 2, we present basic concepts needed to comprehend addressed matters: included are outlines of the Mamdani Type 2 fuzzy inference system, the alignment method and mathematical formulism involved at evaluating acquired competences. In Section 3, we present the results of a simulated data study case and a counterpart based on real data. In Section 4, we present the discussion of the results linking a detailed description of the problem. Conclusions are presented in Section 5. Finally, in an Appendix we present some concepts and definitions of fuzzy set theory that are relevant to build the addressed fuzzy inference system.

2 Methods

2.1 Competencies

The paradigm of acquisition of competencies recently adopted by the Mexican Higher Middle Educational System aims to the development of generic, disciplinary and professional competencies by the students (Figure 1). Pertaining explanations of terms appear in the document entitled Integral Reform of the Higher Middle Education (RIEMS in Spanish) [2].

The development of generic competencies requires to all graduates and is pertinent to all the subjects taught. The disciplinary competencies comprise two categories. They could be basic disciplinary competencies that are intended to be achieved in every subject composing the curricular design at a whole system level. There could be also extended disciplinary competencies whose accomplishment is required to be achieved by the enrolment in the educational scheme pertaining to constituting subsystems. Professional competencies also split into basic and extended categories and with similar associations to the whole system and subsystem levels. Finally, generic competencies comprise 11 different dimensions.

In the present study, we center on generic competencies because they ought to be acquired

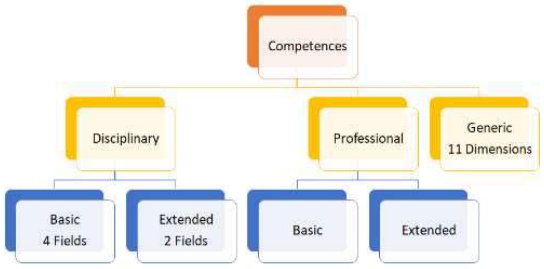


Fig 1. Competencies of the student of higher middle education level defined by the RIEMS [2]

by every student. Certainly, their development could entail a successful performance in any professional field, thereby increasing hiring chances. Generic competencies compose 11 dimensions and each one integrates a set of attributes that every student must develop (Table 1).

The process of evaluating learning by competencies requires a comprehensive evaluation of the student's attainment of affective, cognitive and psychomotor characteristics. In the affective aspect, the know how to be at coexistence is promoted, the individual is encouraged to be a good citizen so that he assumes his rights and duties with responsibility and builds her (his) personal identity; in the cognitive aspect, the known to know is stimulated so the student could seek to develop the ability to use a set of tools that allow processing and interpreting information, and provide solutions to situations that arise; in the psychomotor aspect, the know how to do is heartened by focusing on the individual's performance in real life situations, it induces the use of knowledge in a systemic and reflective way to achieve goals; know-how implies an individual's performance in a proficient way in the completion of an activity or in the solution of a problem based on planning and understanding the domain of the problem [27].

An evaluation criterion should aim to assess the development of the student's acquisition of characteristics associating with the cognitive, affective, and psychomotor domains [29-30]. Using linguistic appraisals of performance conditions, assessment procedures must allow establishing the level of development of competences. Tobón

Table 1. Description of the generic competencies and number of attributes they compose. The letter C symbolizes the word "Competence" and the subscript indicates the competence framed in [2]. Also, a detailed description of the attributes is presented in [2]

Competence	Description	No. of attributes
	Self-determines and takes care of itself	
C_1	Knows and self-values, and addresses problems and challenges, considering the pursued objectives.	6
C_2	Sensitive to art and participates in the appreciation and interpretation of its expressions in different genres.	3
C_3	Chooses and practices healthy lifestyles.	3
	Expresses oneself and communicates with others	
C_4	Listens to, interprets and delivers relevant messages in different contexts using appropriate media, codes and tools.	5
	Thinks critically and reflectively	
C_5	Develops innovations and proposes solutions to problems based on established methods	6
C_6	Holds a personal stance on topics of general interest and relevance, considering other points of view critically and reflectively.	4
	Learns autonomously	
C_7	Learns by own initiative and self-interests throughout life.	3
	Works collaboratively	
C_8	Participates and collaborates effectively on diverse teams.	3
	Participates responsibly in society	
C_9	Participates with a civic and ethical conscience at community, region, Mexico and the world levels.	6
C_{10}	Maintains a respectful attitude towards interculturality and diversity of beliefs, values, ideas and social practices.	3
C_{11}	Contributes to sustainable development in a critical way, with responsible actions.	3

[30] proposed as evaluation elements those corresponding to the affective, cognitive and psychomotor characteristics (Figure 2).

Concisely, an evaluation of acquisition of competencies amounts to collecting evidence through the analysis of learning activities and application of techniques aimed to their assessment. Gonczi and Athanasoub [31] establish that the development of competencies must be evaluated integrally by selecting the most

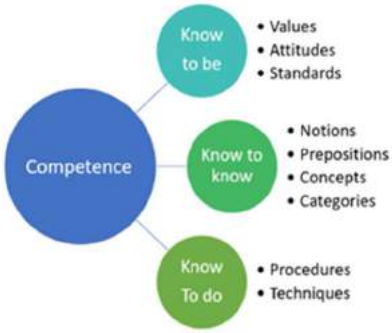


Fig. 2. Elements to evaluate the student's attainment of affective, cognitive and psychomotor characteristics as proposed by [30]

suitable methods, for example, written tests, observation, problem-solving, or a combination of techniques depending on the skill or competency to be evaluated, more examples are presented in Table 2.

Regardless of the evaluation technique used by the teacher, for present aims we have considered that evidence of success in a learning activity should base on the display of following elements: knowledge (notions, concepts, prepositions or/and categories), the procedure or technique carried out, and the attitude shown by the student during achieving a certain process. In other words, according to our approach, the evaluation of the effectiveness of each learning activity should consider the formation of listed elements.

The traditional procedures for evaluating learning, for the most part, centers on the proficiency attained by the student at a specific moment and addresses only the cognitive aspect as the basis of the assessment.

The reason underneath adopting a fuzzy inference system is that it offers a scheme that allows evaluation according to a comprehensive perspective, wherein competence assessment considers affective, cognitive, and psychomotor aspects.

All affective aspects are essential; however, we consider that the student's attitude turns out to be a key feature since it sets; the type of emotion displayed while carrying out a given activity, the way of interacting, or the acting manner in the face of a specific situation or stimulus [32-33]. Castillero-Mimenza [30] classifies different types of attitudes according to various criteria (Table 3).

We consider attitude as evidence according to its practical value, how the subject values the environment or situation, positive or negative. The effects of a positive attitude favor optimistic interpretation regardless of the difficulties that may arise, bringing the subject closer to stimulation or action and to pursue achieving goals through a healthy, confident, and generally disciplined way [30].

The effects of a negative attitude maximize the aversive experience, the subject could minimize the relevance of a circumstance, or directly fail to acknowledge the positive aspects of the situation.

A negative attitude can also induce them to withdraw from acting or acquire complaining behavior beyond what is rational, making it difficult to achieve goals [30].

To acquire competencies, the student must achieve a certain number of learning activities. These activities are guided by the teacher, who should also aim to obtain sufficient evidence to determine how efficiently a student develops them.

The teacher can choose any technique (Table 2) to evaluate a learning activity, as long as the criteria include the ranks that the teacher assigns to knowledge, procedure, and attitude, such that the way in which these elements combine, coordinate, and integrate into determining the student's performance can be appraised [7].

2.2 Fuzzy Inference System

We use the notation convention described in the Appendix to construct the fuzzy inference system. We begin by defining the membership functions that characterize the fuzzy sets inherent to the fuzzy system's antecedents and consequents.

Based in an expert system knowledge base, we consider three input variables, $\text{Knowledge} = \text{X}_1$, $\text{Procedure} = \text{X}_2$ and $\text{Attitude} = \text{X}_3$, to be the antecedents of the fuzzy system and an output variable, $\text{Efficiency} = \text{Y}$, entailing the consequent. To the input variable X_1 we associate the linguistic terms "Little", "Enough" and "Much", characterized by z-shaped (Equation (A6)), Gaussian (Equation (A4)) and sigmoidal (Equation (A5)) membership functions, respectively (see Figure 3).

Similarly, the input variable X_2 describes by the linguistic terms "Not attempted", "Incomplete" and "Achieved" and these characterize by means of the

Table 2. Assessment techniques and their characteristics: (1) conceptual content, (a) facts and data, (b) principles and concepts, (2) procedural content, (3) attitudes and values, (4) thinking skills and (5) auxiliary techniques [12]

Techniques	1		2	3	4	5
	a	b				
Mental maps	X	X	X		X	Checklist.
Problem solving	X	X	X	X	X	Interview, checklist, rubrics and ranks.
Case method	X	X	X	X	X	Interview, checklist, rubrics and ranks.
Projects	X	X	X	X	X	Interview, checklist, rubrics and ranks.
Daily	X	X	X	X	X	Interview
Debate	X	X	X	X	X	Checklist and rubrics.
Techniques based in questions	X	X	X	X	X	Interview and checklist.
Essays	X	X	X	X	X	Interview, checklist, rubrics and ranks.
Briefcase	X	X	X	X	X	Interview, checklist, rubrics and ranks.

Table 3. Classification of the types of student's attitude according with Castillero-Memenza in [30]

Classification	Type	Attitude
1	**According to an effective valence** How they allow to assess the environment and the situation.	• Positive • Negative • Neutral
2	**According to activity orientation** In what way do they generate a concrete approach or orientation towards the idea of carrying out a behavior or activity?	• Proactive • Reactive
3	**According to the motivation to act** What motivates them to carry out a behavior or activity?	• Interested • Selfless
4	**Depending on the relationship with others** How do they get along with others?	• Collaborator • Manipulative • Assertive • Permissive • Passive • Aggressive
5	**According to the type of elements used to assess the stimuli** How do you process reality?	• Emotional • Rational

z-shaped (Equation (A6)), Gaussian (Equation (A4)) and sigmoidal (Equation (A5)) membership functions, respectively (see Figure 3).

In turn, the input variable X_3 characterizes by a trapezoidal membership function defined in the Equation (A7) (see Figure 3).In the same way, the linguistic terms "Terrible", "Very Bad', "Bad", "Regular", "Good", "Very Good", "Excellent" and "Outstanding" associate to the output variable Y and characterize by a triangular membership function defined by Equation (A4) (see Figure 3).

The process by which the IF-THEN rules composing the addressed fuzzy inference system are built must consider the relationship among the variables knowledge, procedure, attitude and efficiency (Table 4).

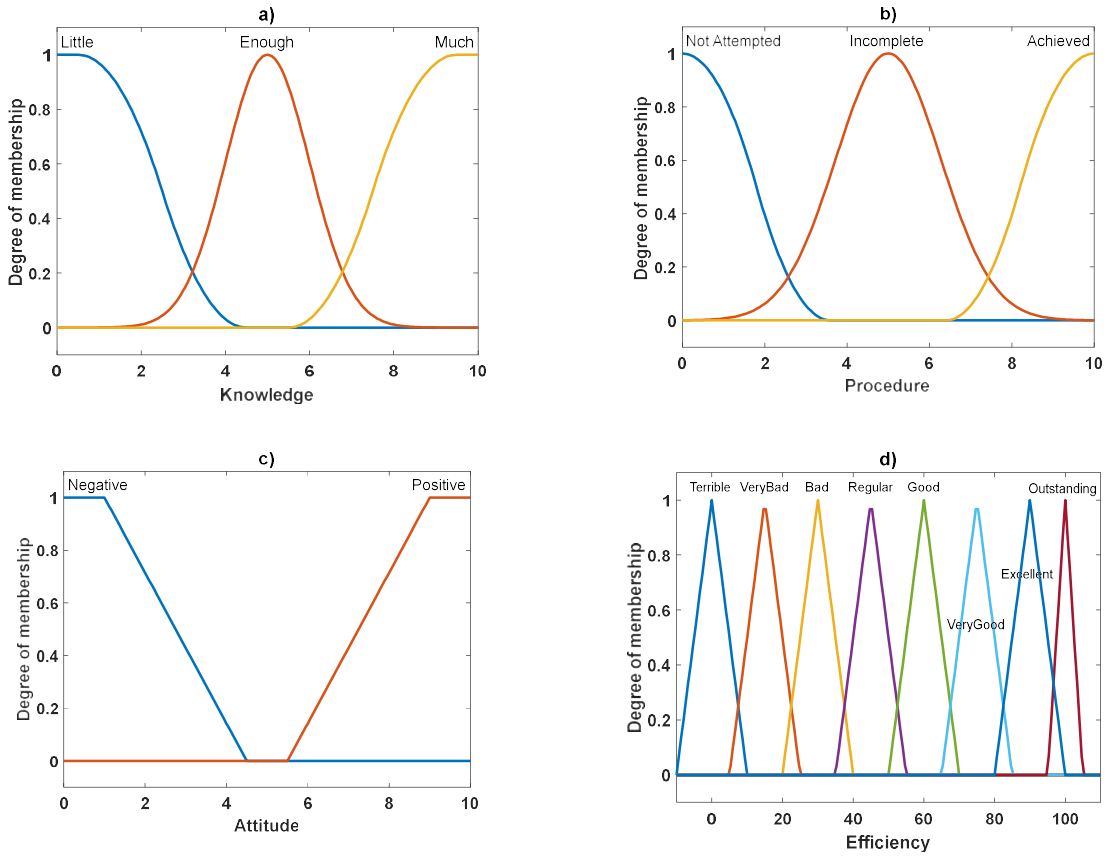


Fig. 3. Membership functions that characterize the fuzzy sets inherent to the fuzzy system's antecedents and consequent. (a) $\text{Little} = zmf(x,[0.5,4.5])$, $\text{Enough} = gaussmf(x,[1,5])$ and $\text{Much} = smf(x,[5.5,9.5])$ (Equations (A6), (A4) and (A5) respectively). (b) $\text{Not Attempted} = zmf(x,[0,3.6])$, $\text{Incomplete} = gaussmf(x,[1.274,5])$ and $\text{Achieved} = smf(x,[6.4,10])$ (Equations (A6), (A4) and (A5) respectively). (c) $\text{Negative} = tmf(x,[0,0,1,4.5])$ and $\text{Possitive} = tmf(x,[5,5.5,10,10])$ (Equation (A3)). (d) $\text{Terrible} = tmf(x,[-10,0,10])$, $\text{Very Bad} = tmf(x,[5,15,25])$, $\text{Bad} = tmf(x,[20,30,40])$, $\text{Regular} = tmf(x,[35,45,55])$, $\text{Good} = tmf(x,[50,60,70])$, $\text{Very Good} = tmf(x,[65,75,85])$, $\text{Excellent} = tmf(x,[80,90,100])$ and $\text{Outstanding} = tmf(x,[95,100,105])$ (Equation (A3))

The fuzzy inference system implements through Matlab's fuzzy logic toolbox (version 2016b). This aims to evaluate the efficiency given the input values x_1, x_2 and x_3 (see Appendix and Figure 4).

Building the fuzzy inference system relied on the Mamdani-Type method [34] (see Appendix). The minimum operator modeled the AND expression (fuzzy intersection in Equation (A9)), and correspondingly the OR expression by the maximum operator (fuzzy union in Equation (A9)).

In turn, the implication defined by the THEN expression modeled by the minimum operator and the aggregation by the maximum operator. The defuzzification step relied on the center of gravity procedure [35] defined in Equation (A14).

2.3 Alignment of Competencies Method

To line up the competencies with the objectives of the learning activities, an alignment matrix is initially constructed. This formalizes the mathematical relationship of the attributes of each competency with each one of the learning activities

Table 4. Relationship among variables knowledge, procedure and attitude and efficiency

	IF						
	Knowledge is		**Procedure** is		**Attitude** is		**Efficiency** is
1	Little	AND	Not attempted	AND	Negative	THEN	Terrible
2	Little	AND	Incomplete	AND	Negative	THEN	Bad
3	Little	AND	Achieved	AND	Negative	THEN	Good
4	Enough	AND	Not attempted	AND	Negative	THEN	Terrible
5	Enough	AND	Incomplete	AND	Negative	THEN	Regular
6	Enough	AND	Achieved	AND	Negative	THEN	Very Good
7	Much	AND	Not attempted	AND	Negative	THEN	Terrible
8	Much	AND	Incomplete	AND	Negative	THEN	Good
9	Much	AND	Achieved	AND	Negative	THEN	Excellent
10	Little	AND	Not attempted	AND	Positive	THEN	Terrible
11	Little	AND	Incomplete	AND	Positive	THEN	Regular
12	Little	AND	Achieved	AND	Positive	THEN	Very Good
13	Enough	AND	Not attempted	AND	Positive	THEN	Very Bad
14	Enough	AND	Incomplete	AND	Positive	THEN	Good
15	Enough	AND	Achieved	AND	Positive	THEN	Excellent
16	Much	AND	Not attempted	AND	Positive	THEN	Bad
17	Much	AND	Incomplete	AND	Positive	THEN	Very Good
18	Much	AND	Achieved	AND	Positive	THEN	Outstanding

included in each unit of the study program. Relationship is made by indicating which attributes will be integrated in the evaluation of each learning activity, this relationship synthetically represents the alignment process.

From left to right, the attributes of the competencies are placed in the first column and from the second column the corresponding learning activities are placed in each unit. When an attribute is involved in a learning activity, the cell corresponding to the intersection of the attribute row with the column of the learning activity is marked, and this is done with the rest of the activities.

Once the alignment matrix is completed, it is easy to know which and how many attributes and competencies are integrated in a learning activity. This allows to calculate the value of the weight of each learning activity and each unit included in the study program. The grade of development of the competencies acquired by the student or by a group will be evaluated using named weight values. The grade is the percentage value of the development of the acquired competencies.

A study program can be structured by a number n of units and each one of them by another number m of learning activities (Figure 5). That is, let $\mathrm{U_i}$, for $i = 1 \ldots n$, be the ith unit and let $\mathrm{H_{ij}}$, for $j = 1 \ldots m$, be the jth learning activity of the unit $\mathrm{U_i}$.

To illustrate how the alignment method works, we will consider as an example a study program containing three units; unit $\mathrm{U_1}$ involving learning

activities $\mathrm{H_{1j}}$, for $j = 1,2,3$, unit $\mathrm{U_2}$ with learning activities $\mathrm{H_{2j}}$, for $j = 1,2$, and unit $\mathrm{U_3}$ that integrates learning activities $\mathrm{H_{3j}}$, for $j = 1,2,3$. Similarly, the $h-th$ competence denotes by $\mathrm{C_h}$, for $h = 1,2,\ldots,11$ (Table 1) and the $k-th$ attributes of a competence $\mathrm{C_h}$ represents through $\mathrm{C_{hk}}$, that is, the index k designates the kth attribute of the competence $\mathrm{C_h}$ (Table 5), we create the alignment matrix and make the alignment indicating with the value of 1 that attribute $\mathrm{C_{hk}}$ is integrated in the learning activity $\mathrm{H_{ij}}$, (Table 5).

So, let $V^{(\mathrm{H_{ij}})}$ be the number of competence attributes integrated in the learning activity $\mathrm{H_{ij}}$, which can be expressed as:

$$V^{(\mathrm{H_{ij}})} = \sum_{h=1}^{11} \sum_{k=1}^{p} \mathrm{C}_{\mathrm{hk}}^{(\mathrm{H_{ij}})}, \tag{1}$$

where the symbol $\mathrm{C}_{\mathrm{hk}}^{(\mathrm{H_{ij}})}$ denotes the attribute k of the competence $\mathrm{C_h}$ integrated in the learning activity $\mathrm{H_{ij}}$ (Table 5) and p represent the total number of attributes of the competence $\mathrm{C_h}$. Likewise, $V^{(\mathrm{U_i})}$ represent the total number of attributes integrated to the unit $\mathrm{U_i}$, which is obtained from the expression:

$$V^{(\mathrm{U_i})} = \sum_{j=1}^{m} V^{(\mathrm{H_{ij}})}, \tag{2}$$

and the total sum of attributes V integrated in a study program will be calculated by:

$$V = \sum_{i=1}^{n} V^{(\mathrm{U_i})}, \tag{3}$$

Considering the information presented in Table 5, we take Equations (1) and (2) to obtain the total number of competence attributes integrated in $\mathrm{H_{ij}}$ and $\mathrm{U_i}$ respectively (Table 6).

To finish the description of the alignment method, we calculate the values of the weights, v_{ij} and u_i, corresponding to the alignment matrix. The weight value v_{ij} is obtained through:

$$v_{ij} = \frac{100}{\sum_{j=1}^{m} V^{(\mathrm{H_{ij}})}} \times V^{(\mathrm{H_{ij}})}, \tag{4}$$

And the weight value u_i is obtained through:

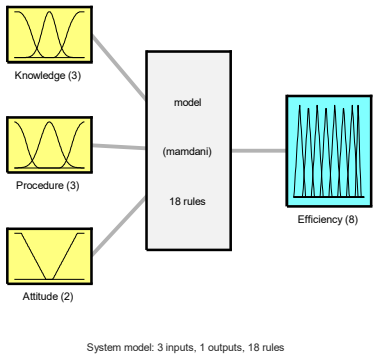


Fig. 4. Fuzzy inference system composing three antecedents: Knowledge, Procedure and Attitude, and a consequent, Efficiency, which are characterized by membership functions parameterized as describing in Figure 3. The 18 fuzzy rules integrated into conceived fuzzy inference system appear in Table 4

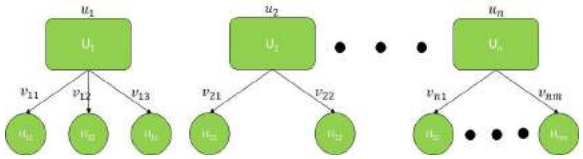


Fig. 5. Units ($\mathrm{U_i}$ for $i = 1,\ldots,n$) and learning activity ($\mathrm{H_{ij}}$ for $i = 1,\ldots,n$ and $j = 1,\ldots,m$) that comprise a study program. v_{ij} is the weight value corresponding to the jth learning activity of ith unit and u_i is the weight values corresponding to the unit ith

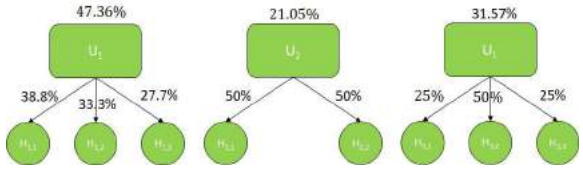


Fig. 6. Weight values obtained through the Equations (4) and (5) and corresponding to the alignment matrix presented in Table 5

$$u_i = \frac{100}{\sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij}} \times V^{(\mathrm{U_i})}, \tag{5}$$

For the addressed example, we calculate the weight values, v_{ij} and u_{ij}, that correspond to the alignment matrix (Table 5) through the Equations (4) and (5) and considering the total number of competence attributes integrated in the learning activities $\mathrm{H_{ij}}$ and in the unit $\mathrm{U_i}$ from Table 6 (Figure 6).

Table 5. Alignment matrix: attribute C_{hk} with the learning activity H_{ij} included in a hypothetical study program taken as example

		U_1				**U_2**				**U_3**	
C_{hk}	H_{11}	H_{12}	H_{13}	**C_{hk}**	H_{21}	H_{22}	**C_{hk}**	H_{31}	H_{32}	H_{33}
C_{31}	1	1		**C_{31}**		1	**C_{31}**	1		
C_{32}		1	1	**C_{32}**			**C_{32}**		1	1
C_{33}	1			**C_{33}**	1		**C_{33}**		1	
C_{51}	1		1	**C_{51}**			**C_{51}**	1	1	
C_{51}		1		**C_{52}**		1	**C_{52}**		1	
C_{53}			1	**C_{53}**			**C_{53}**			1
C_{81}		1	1	**C_{81}**			**C_{81}**		1	
C_{82}	1			**C_{82}**	1		**C_{82}**	1		1
C_{83}		1		**C_{83}**		1	**C_{83}**		1	
$C_{11\,1}$	1			**$C_{11\,1}$**	1					
$C_{11\,2}$	1	1		**$C_{11\,2}$**		1				
$C_{11\,3}$	1		1	**$C_{11\,3}$**	1					

Table 6. Total number of attributes integrated in H_{ij} and U_i corresponding to the information presented in Table 5

Unit	$V^{(H_{i1})}$	$V^{(H_{i2})}$	$V^{(H_{i3})}$	$V^{(U_i)}$
1	7	6	5	**18**
2	4	4	0	**8**
3	3	6	3	**12**

The alignment method consists of establishing an implicit relationship between the attribute C_{hk} to be evaluated and the learning activity H_{ij} where it will be evaluated.

The teacher could select the most appropriate technique (Table 2) to evaluate the learning activity and could establish the criteria to collect the evidence (knowledge, procedure and attitude) required in our fuzzy system (Figure 4) to evaluate the efficiency with which the student developed the activity. Our method requires that the teacher indicates which competence attributes (into the alignment matrix, e.g., Table 5) are considered in each learning activity of the study program, then the educational evaluation system calculates the weights, v_{ij} and u_i.

Keep in mind that in no way will it be possible to change the weights values, v_{ij} and u_i, manually.

These values only can change when the alignment modifies. Also, it must be considered that the same competence attribute could be

evaluated in one or more learning activities of one or more units.

Therefore, the student's progress, or in other words, the grade of development of the competencies acquired by the student is determined in a percentage way considering the tree forming by the resulting weight values (e.g., Figure 6) of the alignment matrix (e.g., Table 6).

2.4 Grade of Development of Acquired Competencies

The evaluation of the efficiency E_{ij} bases on the quantitative and synthetic conception of the components explaining the performance displayed by the student in the courses of a learning activity $\mathrm{H_{ij}}$.

Then, through the resulting weight values (e.g., Figure 6) of the alignment matrix (e.g., Table 6) and the efficiency E_{ij}, the educational evaluation system determines the grade of development of acquired competencies over a period within the study program. This is as follows: The grade of development of the competencies acquired by a student at the end of the learning activity $\mathrm{H_{ij}}$ is denoted by $\mathrm{D}_s^{(\mathrm{H_{ij}})}$ and calculated through:

$$\mathrm{D}_s^{(\mathrm{H_{ij}})} = E_{ij} \times v_{ij}, \tag{6}$$

where s denotes the sth student, the efficiency value, E_{ij}, is provided by the fuzzy system and v_{ij} is the weight value corresponding to the learning activity $\mathrm{H_{ij}}$. Likewise, $\mathrm{D}_s^{(\mathrm{U_i})}$ denotes the grade of development of the competencies acquired by the sth student while completing unit $\mathrm{U_i}$ and obtained by:

$$\mathrm{D}_s^{(\mathrm{U_i})} = \sum_{j=1}^{m} E_{ij} \times v_{ij} \times u_i, \tag{7}$$

where u_i is the weight value corresponding to the unit $\mathrm{U_i}$. In addition, the grade of development of the competencies acquired by the sth student while completing the study program, D_s, is obtained by:

$$\mathrm{D}_s = \sum_{i=1}^{n} \mathrm{D}_s^{(\mathrm{U_i})}. \tag{8}$$

Taking Equations (6-8) as a reference, it is possible to calculate a series of values that represent statistical parameters of the educational system that are relevant at obtaining information about students or groups.

For example, the number $\mathrm{N^+}$ of students who acquired their competencies at the end of the study program with a grade of development D_s greater than P or less than P, and correspondingly the number $\mathrm{N^-}$ of those achieving a developmental grade smaller that P, can be calculated by:

$$\mathrm{N^+} = \sum_{s=1}^{n_s} 1 \quad \forall\, \mathrm{D}_s > \mathrm{P} \text{or } \mathrm{N^-} = \sum_{s=1}^{n_s} 1 \quad \forall\, \mathrm{D}_s < \mathrm{P}, \tag{9}$$

where P is a perceptual value and n_s is the number of students to be evaluated.

But if what we want is to calculate the percentage value associating to $\mathrm{N_{\%}^+}$ of $\mathrm{N^+}$, or that one $\mathrm{N_{\%}^-}$ corresponding to $\mathrm{N^-}$, then we can use the following expressions:

$$\mathrm{N_{\%}^+} = \frac{\mathrm{N^+}}{n_s} \times 100 \text{ or } \mathrm{N_{\%}^-} = \frac{\mathrm{N^-}}{n_s} \times 100. \tag{10}$$

In addition, we can obtain the average percentage of the grade of development of the competencies acquired by a group is obtained through:

$$\mathrm{D} = \frac{\sum_{s=1}^{n_s} \mathrm{D}_s}{n_s}. \tag{11}$$

Even more, it is possible to modify the Equation (9) if what we want is calculating the number $\mathrm{N_H^+}$ of students who acquired their competencies at the end of a learning activity $\mathrm{H_{ij}}$ with a grade of development greater than P, namely:

$$\mathrm{N_H^+} = \sum_{s=1}^{n_s} 1 \quad \forall\, \mathrm{D}_s^{(\mathrm{H_{ij}})} > \mathrm{P}. \tag{12}$$

Similarly, if we want is calculating the number $\mathrm{N_H^-}$ of students who acquired their competencies at the end of a learning activity $\mathrm{H_{ij}}$ with a grade of development less than P, then:

$$\mathrm{N_H^-} = \sum_{s=1}^{n_s} 1 \quad \forall\, \mathrm{D}_s^{(\mathrm{H_{ij}})} < \mathrm{P}. \tag{13}$$

Respectively. Likewise, if what we want is calculating the number $\mathrm{N_U^+}$ of students who

Table 7. Alignment matrix: attributes integrated within the learning activities that comprise the study program corresponding to the simulated data study case

	U1				**U2**				**U3**			
$\mathbf{C_{hk}}$	H11	H12	H13	$\mathbf{C_{hk}}$	H21	H22		$\mathbf{C_{hk}}$	H31	H32	H33	
C11		1		**C22**	1			**C14**	1		1	
C16	1			**C33**		1		**C32**	1			
C44		1		**C61**	1			**C53**			1	
C55		1		**C73**		1		**C95**		1		
C93	1		1	**C82**		1		**C10 3**	1			
C11 1	1			**C91**		1		**C11 3**			1	

Table 8. Total number of attributes integrated in the learning activities and units included in the alignment matrix presented in Table 7

Unit	$V^{(\mathrm{H}_{i1})}$	$V^{(\mathrm{H}_{i2})}$	$V^{(\mathrm{H}_{i3})}$	V^{U_i}
1	3	3	1	**7**
2	2	4		**6**
3	3	1	3	**7**

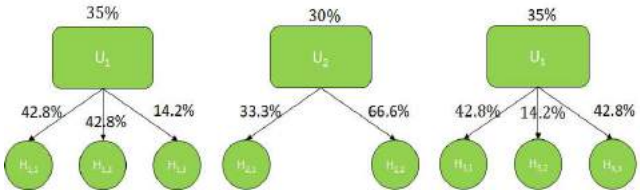


Fig. 7. Weights v_i and u_{ij} obtained through Equations (4) and (5) associating to the alignment matrix presented in Table 7

acquired their competencies at the end of the unit U_i with a grade of development greater than P, then:

$$\mathrm{N}_{\mathrm{U}}^{+} = \sum_{s=1}^{n_s} 1 \ \forall \ \mathrm{D}_s^{(\mathrm{U_i})} > \mathrm{P}, \tag{14}$$

or less than P:

$$\mathrm{N}_{\mathrm{U}}^{-} = \sum_{s=1}^{n_s} 1 \ \forall \ \mathrm{D}_s^{(\mathrm{U_i})} < \mathrm{P}, \tag{15}$$

Table 9. Simulated values $(n = 50)$ for x_1, x_2 and x_3 for each one of the $\mathrm{H_{ij}}$ learning activities corresponding to the unit $\mathrm{U_1}$

$\mathrm{U_1}$																		
$\mathrm{H_{11}}$ (1st to 25th)			$\mathrm{H_{12}}$ (1st to 25th)			$\mathrm{H_{13}}$ (1st to 25th)			$\mathrm{H_{11}}$ (25th to 50th)			$\mathrm{H_{12}}$ (25th to 50th)			$\mathrm{H_{13}}$ (25th to 50th)		
x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3
9.49	3.45	5.92	2.51	9.99	8.01	4.29	8.43	8.38	9.68	2.54	6.92	2.60	9.62	8.36	3.67	6.55	8.03
9.72	3.21	7.70	2.82	9.56	9.44	3.62	7.87	7.75	9.70	3.01	6.89	2.84	9.25	9.70	4.13	7.98	8.30
9.61	2.63	6.42	2.53	9.60	9.41	4.01	7.35	8.42	9.90	2.59	7.76	2.92	8.72	9.06	4.14	7.48	8.01
9.46	2.83	6.97	2.29	8.60	9.07	4.22	8.12	8.43	9.02	2.87	7.47	2.88	7.78	8.42	3.89	7.52	9.46
9.83	2.53	7.68	2.56	7.90	9.93	4.97	6.17	9.86	9.35	2.45	6.57	2.99	9.55	9.75	4.96	8.37	8.13
9.67	2.71	7.12	2.30	8.44	9.29	3.73	6.54	9.31	9.44	2.36	6.05	2.53	9.78	9.93	3.97	6.33	9.82
9.86	2.88	7.83	2.99	9.16	9.63	4.58	7.37	9.68	9.59	2.66	7.99	2.71	8.93	9.72	4.05	7.97	9.72
9.81	2.65	6.72	2.79	8.80	8.58	3.99	7.92	9.87	9.63	2.50	7.37	2.88	8.33	10.0	4.35	7.43	8.97
9.33	2.92	7.28	2.55	9.80	9.58	4.08	7.58	8.31	9.52	2.30	6.78	2.67	8.36	9.00	3.86	7.76	8.25
9.56	2.79	7.28	2.75	9.91	9.33	4.09	7.01	8.03	9.30	2.83	6.98	2.56	9.54	10.8	4.00	7.37	8.39
9.02	2.27	6.46	2.72	8.91	9.25	4.94	6.73	9.52	9.81	2.47	7.60	2.66	8.50	9.93	4.85	6.80	8.24
9.26	3.15	6.97	2.78	9.50	9.76	3.98	8.05	9.09	9.98	2.67	7.29	2.75	10.0	9.37	3.66	6.90	8.23
9.49	2.52	7.54	2.43	8.77	9.23	3.62	8.90	8.69	9.44	2.88	7.43	2.54	9.17	10.0	4.30	8.09	10.0
9.92	3.01	6.70	2.42	9.37	9.70	3.55	6.39	8.53	9.75	2.67	6.53	2.60	9.68	10.0	3.79	6.29	8.61
9.02	2.83	6.34	2.46	8.79	9.91	3.64	6.67	8.28	10.0	2.63	7.89	2.64	10.0	8.99	4.15	7.23	8.84
9.83	2.87	7.31	2.59	8.80	9.67	4.09	7.28	8.65	9.70	2.89	7.05	2.75	8.58	10.0	3.74	6.90	9.03
9.51	3.16	7.43	2.63	7.80	8.63	3.60	8.29	9.15	9.67	2.74	7.77	2.82	8.89	9.69	4.03	7.57	8.60
9.75	2.67	7.43	2.60	9.58	9.94	4.27	6.42	9.44	9.46	2.44	7.79	2.42	8.43	9.21	3.83	6.61	8.72
9.33	2.61	7.52	2.61	9.07	9.70	3.76	7.46	8.95	9.97	2.39	7.71	2.20	9.54	10.0	4.09	7.32	7.66
9.07	2.40	7.86	2.91	8.25	9.51	3.61	8.58	7.74	9.34	3.36	6.54	2.84	9.54	10.0	4.47	6.25	9.96
9.09	2.36	6.76	2.75	7.77	9.57	4.42	6.39	7.98	9.72	2.82	6.65	3.36	7.97	10.0	3.59	6.84	9.62
9.62	2.86	7.61	2.50	7.90	9.69	4.38	7.87	9.26	9.22	2.74	7.65	3.09	9.32	9.44	3.87	6.48	8.35
9.30	2.33	7.06	2.36	8.43	9.29	4.20	7.61	8.63	9.15	2.59	7.91	2.71	8.56	10.0	4.03	7.58	9.17
9.15	2.37	6.86	2.66	8.28	8.85	4.62	7.33	8.91	9.25	2.50	7.44	2.57	8.27	10.0	4.48	8.11	9.10
9.95	2.80	6.72	2.76	8.20	9.70	4.10	7.14	9.48	10.0	3.00	7.35	2.48	8.61	9.26	4.45	7.12	10.0

3 Results

3.1 Study Case Based on Simulated Data

To illustrate the performance of offered evaluation scheme, we first consider a try based on a simulated data set. We conceive a hypothetical group of 50 students, assumed to have enrolled in a study program structured in three units: unit $\mathrm{U_1}$ involving learning activities $\mathrm{H_{1j}}$ for $j = 1,2,3$, unit $\mathrm{U_2}$ composing learning activities $\mathrm{H_{2j}}$ for $j = 1,2$ and $\mathrm{U_3}$ with $\mathrm{H_{3j}}$ for $j = 1,2,3$ learning activities.

Following the procedure in Section 2.3, we create the associating alignment matrix, but first,

Table 10. Simulated values $(n = 50)$ for x_1, x_2 and x_3 for each one of the $\mathrm{H_{ij}}$ learning activities corresponding to the unit $\mathrm{U_2}$

$\mathbf{U_2}$												
H_{21} (1st to 25th)			**H_{22} (1st to 25th)**			**H_{21} (25th to 50th)**			**H_{22} (25th to 50th)**		
x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3
2.68	4.57	9.68	5.53	7.72	3.23	3.24	4.78	8.33	4.95	8.20	2.88
2.64	5.33	9.60	5.18	7.41	2.75	2.94	5.16	9.50	4.55	8.03	2.77
2.68	4.34	9.87	4.68	7.59	2.50	2.88	5.06	10.0	5.47	8.46	3.20
2.28	4.63	9.48	4.92	8.49	2.57	2.61	4.31	10.0	5.04	8.04	2.87
2.78	4.77	9.20	4.25	8.08	2.84	3.26	5.22	10.0	5.50	7.29	3.17
2.66	5.64	9.74	6.02	7.29	2.49	2.96	5.60	8.98	5.44	8.12	2.49
3.06	4.52	9.17	4.91	7.66	3.09	2.48	4.77	10.0	5.40	8.87	3.03
2.42	4.62	8.99	5.12	8.36	2.62	2.36	6.07	11.78	5.55	8.63	3.03
2.78	4.36	9.18	5.73	7.88	2.57	2.42	5.08	10.0	4.65	8.28	2.69
2.60	4.89	9.49	4.31	8.60	2.81	2.86	5.22	10.0	5.52	8.54	3.26
2.65	4.78	9.65	5.01	7.05	2.28	2.37	5.45	10.0	4.97	8.35	2.86
3.03	4.95	8.68	4.84	8.34	3.36	2.55	5.24	8.65	5.35	7.94	2.34
2.77	4.42	9.97	4.74	8.57	2.84	2.38	5.21	10.0	4.86	7.43	2.35
2.67	4.91	9.41	4.70	8.13	3.04	2.79	5.78	9.80	5.24	7.68	2.70
3.04	5.70	8.76	5.90	8.74	2.64	2.46	4.64	10.0	5.27	9.85	2.70
2.64	5.95	9.59	5.77	8.35	2.71	2.66	3.86	9.47	5.27	7.02	2.90
2.70	5.19	8.20	4.99	8.95	3.29	2.52	4.60	10.0	4.86	9.15	2.87
2.82	4.47	9.44	5.25	7.42	2.87	3.58	4.74	8.37	5.17	8.02	2.90
2.82	5.36	9.93	4.78	7.84	2.54	3.11	5.76	10.0	4.59	8.81	2.34
2.83	5.00	8.78	4.18	8.56	3.24	2.98	5.00	8.83	4.66	9.86	3.20
2.65	4.99	8.89	4.90	8.41	2.17	2.95	5.53	9.95	4.82	8.33	2.61
2.95	5.08	9.96	4.82	8.65	2.89	2.32	4.83	10.0	5.46	8.34	2.89
3.12	5.52	9.50	4.87	7.68	2.72	2.93	4.69	9.09	4.66	8.03	2.64
2.94	4.27	8.61	4.19	8.48	3.14	2.84	5.38	8.41	5.35	9.18	2.75
2.74	6.62	9.96	5.70	8.72	2.50	2.87	4.88	10.0	4.82	9.03	2.72

we achieve this task randomly, on its course, indicating the $\mathrm{C_{hk}}$ attributes integrated within the learning activity $\mathrm{H_{ij}}$ (Table 7). As we mentioned, the teacher ought to achieve the alignment systematically, thereby only in extreme cases could they only align the competencies at random. However, the analysis of random alignment requires case attains relevance because it stands for the worst expected scenario for an inconsistent evaluation. In this sense, we can explore the suitability of the proposed method even in extreme cases of subjective alignment.

According with the alignment matrix presented in Table 7, we obtained the total number of attributes $\mathrm{C_{hk}}$ integrated in $\mathrm{H_{ij}}$ and U_i through Equations (1) and (2), respectively (Table 8).

Table 11. Simulated values $(n = 50)$ for x_1, x_2 and x_3 for each one of the $\mathrm{H_{ij}}$ learning activities corresponding to the unit $\mathrm{U_3}$.

U_3																		
H_{31} (1st to 25th)			H_{32} (1st to 25th)			H_{33} (1st to 25th)			H_{31} (25th to 50th)			H_{32} (25th to 50th)			H_{33} (25th to 50th)		
x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3
2.50	3.99	6.81	6.74	5.92	9.95	9.15	7.05	5.43	2.85	3.74	6.62	6.40	6.72	9.78	8.90	7.99	4.98
2.58	4.12	5.50	7.42	5.92	8.80	9.86	7.78	4.67	2.84	3.72	6.91	6.39	6.01	9.17	8.51	6.49	5.69
2.60	4.54	6.28	6.49	5.30	9.16	9.15	6.58	5.17	2.93	4.64	6.53	7.90	6.53	9.96	9.90	7.68	5.35
2.86	3.68	5.09	7.20	5.95	9.37	8.47	7.55	5.39	2.76	4.85	5.18	7.37	5.90	9.93	8.71	7.39	4.87
2.55	3.75	5.93	7.81	6.01	9.94	9.36	7.59	4.40	2.30	4.05	5.86	7.90	6.21	10.0	8.89	7.43	5.37
2.82	3.66	6.22	7.63	5.53	8.74	9.96	6.70	4.73	2.92	3.62	5.75	7.25	5.64	8.92	9.31	6.19	5.29
2.66	4.51	5.39	7.24	5.72	8.97	9.45	6.94	5.96	2.77	3.67	5.15	6.25	5.57	10.0	9.87	7.64	4.90
2.34	4.48	5.88	7.18	5.45	8.70	8.85	6.75	4.33	2.56	4.25	6.20	6.32	5.09	9.22	9.62	7.51	5.80
2.37	4.52	5.58	7.83	5.24	9.70	9.73	7.57	4.29	2.39	4.41	6.18	6.90	5.51	8.92	8.59	6.32	5.10
2.48	3.81	6.56	6.73	6.28	9.58	9.90	7.13	4.56	2.86	3.87	6.66	6.65	5.86	10.0	8.75	7.41	4.21
2.72	4.26	5.65	6.44	6.24	9.18	8.23	6.65	4.32	2.83	4.50	5.65	7.51	6.11	9.10	9.27	7.32	4.58
2.70	4.32	6.00	6.78	6.05	8.73	8.59	8.00	4.30	2.65	4.51	6.25	7.25	5.47	9.57	9.46	6.65	5.21
2.46	4.54	6.04	7.37	6.48	8.77	9.87	6.42	4.84	2.50	3.78	5.74	6.89	5.81	9.76	9.48	7.47	4.65
2.64	4.31	5.72	7.48	5.19	9.54	8.13	7.10	5.78	2.93	3.75	6.14	8.70	6.98	10.0	9.10	6.70	4.49
2.75	3.89	5.58	6.52	6.13	9.25	8.97	7.28	5.64	2.79	3.82	6.13	6.33	6.53	10.0	9.53	7.68	4.35
2.70	4.17	5.57	7.38	6.31	8.07	9.49	6.05	5.17	2.91	4.19	6.14	7.27	5.97	8.10	9.91	7.83	4.75
2.33	4.09	6.18	7.53	6.16	9.05	8.76	7.14	5.38	2.29	4.17	6.18	7.26	6.09	9.29	7.95	8.15	5.24
2.68	4.13	6.83	7.23	5.61	8.90	9.11	7.61	5.32	2.57	3.87	6.80	6.88	6.15	10.0	9.67	8.32	5.68
2.93	3.65	5.50	7.38	5.99	9.64	9.51	7.28	5.43	2.37	4.90	5.94	7.65	6.80	9.87	8.37	7.96	4.88
2.66	4.25	5.45	7.88	6.52	9.41	9.11	7.67	5.73	2.43	4.16	6.22	6.93	6.36	10.0	8.59	6.70	4.21
2.68	4.85	6.56	6.33	5.31	9.86	9.53	7.89	4.92	2.61	4.81	6.05	7.41	6.66	9.20	9.73	8.31	4.41
2.45	4.62	5.83	5.85	6.38	9.58	8.14	6.68	5.29	2.73	4.85	5.44	8.55	5.68	10.0	9.34	9.40	5.44
2.70	3.87	5.68	7.39	6.69	8.50	9.14	7.14	4.79	3.00	4.92	6.48	6.13	5.93	10.0	9.10	8.56	5.39
2.34	3.98	6.35	6.28	6.53	9.35	9.27	7.69	4.38	2.63	5.01	6.11	9.07	5.40	9.97	9.52	7.01	4.81
2.64	4.15	5.92	6.99	6.02	9.38	9.00	6.53	4.74	3.32	3.22	6.64	8.20	6.35	9.98	10.0	6.10	4.04

In turn, weight values v_{ij} and u_i calculated through Equations (4) and (5), respectively (Figure 7). Also, we considered simulated values for x_1, x_2 and x_3 generated so that the highest concentrations of values distributed around a given mean (Tables 9, 10 and 11).

Figure 8 shows the distributions of the x_1, x_2 and x_3 values fitted to a non-normal distribution curve. In Figure 8 domed lines differentiate the interval where the distributions of the input values x_1, x_2 and x_3 concentrates for each learning activity $\mathrm{H_{ij}}$.

Shown concentration intervals help the teacher visualizing the possible effects of the combinations of x_1, x_2 and x_3 values over the performance of the students during the development of their activities.

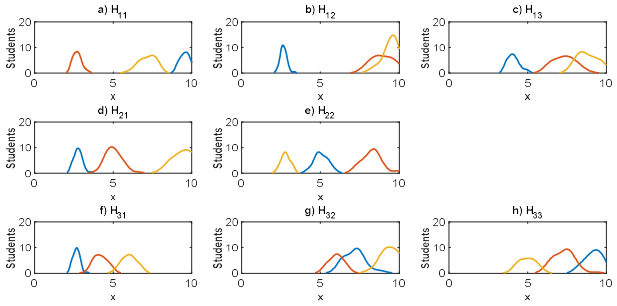


Fig. 8. Distribution of the x_1, x_2 and x_3 simulated inputs values. x_1 corresponds to $\text{knowledge} = \text{X}_1$ variable (blue lines), x_2 corresponds to the $\text{procedure} = \text{X}_2$ variable (red lines) and x_3 corresponds to the $\text{attitude} = \text{X}_3$ variable (yellow lines)

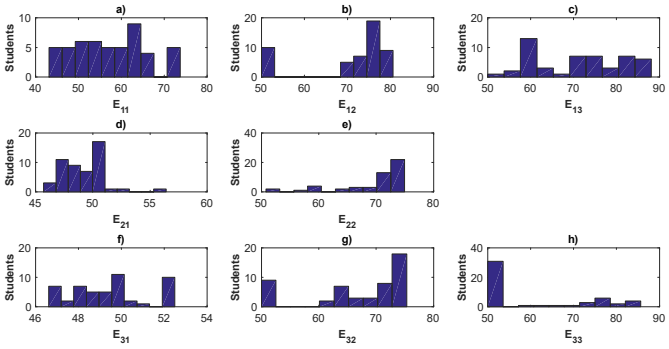


Fig. 9. Distribution of the E_{ij} evaluated by the fuzzy inference system (Figure 4) taking as an input the 50 simulated values presented in Tables 9, 10 and 11

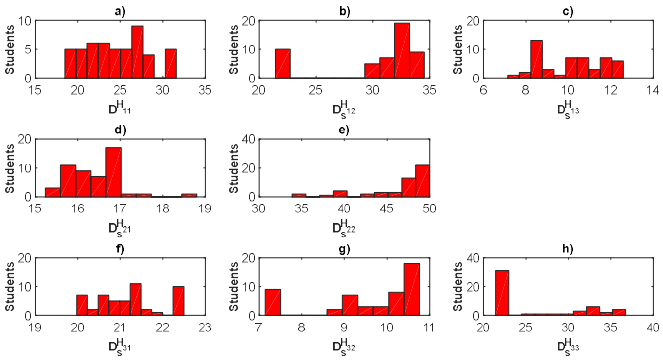


Fig. 10. Distribution of the $\text{D}_s^{(\text{H}_{ij})}$ grade of development of the competences acquired by the students in the learning activity H_{ij}.

For example, we can observe in Figure 8b that in all cases knowledge rated insufficient, however, an attitude with very high values in x_2 associates to a completed procedure. Contrary to Figure 8b, it is possible to observe in Figure 8a a non-attempted procedure, despite high values in attitude and

knowledge. In this case, the teacher can analyze and determine, if the technique and the teaching-learning strategy are adequate to evaluate the efficiency.

An efficiency value E_{ij} corresponding to each learning activity $\mathrm{H_{ij}}$ calculates from running our fuzzy inference system (Figure 4).

In performing such a task, input values are those as simulated for the hypothetical group of 50 students (Tables 9, 10 and 11). We show resulting E_{ij} values in Figure 9.

Figure 9 provides the teacher a clue on the performance of the students in each one of the learning activities $\mathrm{H_{ij}}$. For example, if most students obtained values of E_{ij} greater that 50% then this would mean that knowledge was at least sufficient, that the procedure was attempted in half of the associated learning activities associated, and that the attitude was more positive than negative.

But, in our hypothetical study case, there is a lot of variability in the distribution of the E_{ij} values presented in Figure 9. However, it is possible to identify the values x_1, x_2 and x_3 belonging to those students who had the lowest E_{ij} ratings, which allows to perform analytics and conceiving better teaching strategies aimed to improve learning.

Equation (6) allows calculation of $\mathrm{D}_s^{(\mathrm{H_{ij}})}$, interpreting as the grade of development of the competencies acquired by the sth student when performing the learning activity $\mathrm{H_{ij}}$, (Figure 10). Acquiring frequency distribution plots of resulting numbers permits visualizing the behavior of the grade of development of competencies.

Similarly, Equation (7) endures calculation of $\mathrm{D}_s^{(\mathrm{U_i})}$ standing for the grade of development of the competencies acquired by the qth student in unit $\mathrm{U_i}$, (Figure 11).

Additionally, D_s, identifying the grade of development of the competencies acquired by the sth student at the end of the study program, calculates by means of Equation (8) (Figure 12). Then, adding the results at the individual level, we obtain the grade of development of the competencies acquired by the student's group at the end of the study program.

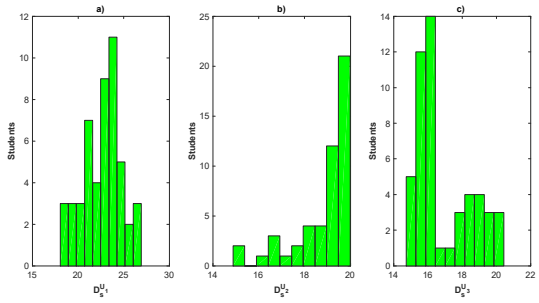


Fig. 11. Distribution of the $\mathrm{D}_s^{(\mathrm{U_i})}$ grade of development of the competences acquired by the students in unit $\mathrm{U_i}$

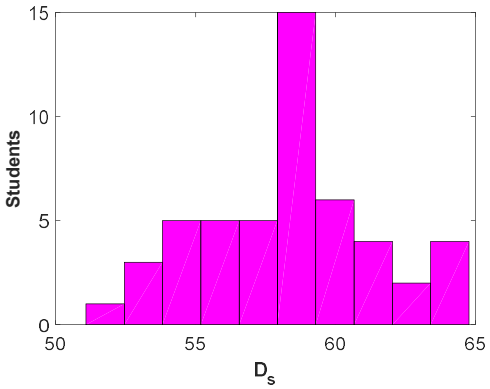


Fig. 12. Distribution of the $\mathrm{D_s}$ grade of development of the competences acquired by all students at the end of the study program

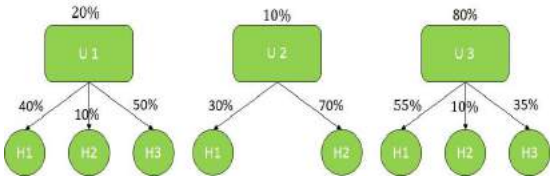


Fig. 13. Hypothetical weights v_i and u_{ij}

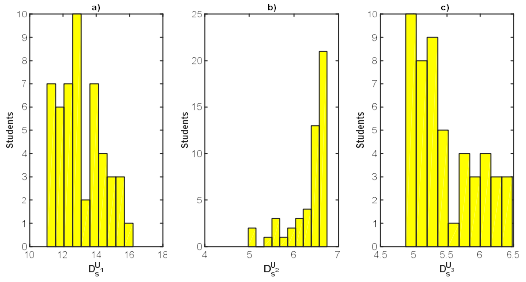


Fig. 14. Distribution of the $\mathrm{D}_s^{(\mathrm{U_i})}$ grade of development of the competences acquired by the students at the end to the unit U_i

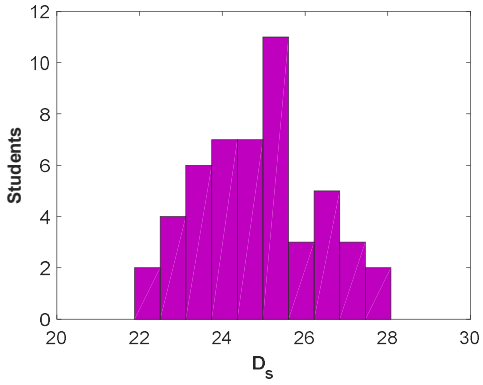


Table 12. Scores of students enrolling the subject “Operating Systems Management” presented as image in [12]

Units					Unit 1										Unidad 2					
Results of learning					RA 1				RA 2						RA 1			RA 2		
Activities to evaluate					AE 1 (20%)				AE 1 (20%)						AE 1 (25%)			AE 1 (35%)		
Indicators					1	2	3	4	1	2	3	4	5		1	2	3	1	2	3
%	1.1.1	1.2.1	2.1.1	2.2.1	30%	30%	35%	5%	20%	20%	20%	20%	20%		20%	20%	60%	20%	30%	50%
86.88	73.75	100	100	77.5	E	E	I	E	E	E	E	E	E		E	E	E	E	I	E
100	100	100	100	100	E	E	E	E	E	E	E	E	E		E	E	E	E	E	E
100	100	100	100	100	E	E	E	E	E	E	E	E	E		E	E	E	E	E	E
100	100	100	100	100	E	E	E	E	E	E	E	E	E		E	E	E	E	E	E
100	100	100	100	100	E	E	E	E	E	E	E	E	E		E	E	E	E	E	E
93.7	100	91	82	100	E	E	E	E	E	E	E	S	S		S	E	S	E	E	E
100	100	100	100	100	E	E	E	E	E	E	E	E	E		E	E	E	E	E	E
100	100	100	100	100	E	E	E	E	E	E	E	E	E		E	E	E	E	E	E
77.16	98.88	100	71.5	55.75	E	E	E	S	E	E	E	E	E		I	E	S	E	S	I
30.25	51.25	25	25	25	E	I	I	E	I	I	I	I	I		I	I	I	I	I	I
28.68	43.38	25	25	25	S	I	I	S	I	I	I	I	I		I	I	I	I	I	I
48.33	85.38	55	25	40	E	S	S	E	E	E	I	I	I		I	I	I	E	I	I
100	100	100	100	100	E	E	E	E	E	E	E	E	E		E	E	E	E	E	E
57.93	67	82	77.5	25	E	S	I	E	S	E	S	S	S		S	S	S	I	I	I
62.58	92.13	67	67	40	E	E	S	E	I	S	S	S	S		S	I	S	E	I	I
100	100	100	100	100	E	E	E	E	E	E	E	E	E		E	E	E	E	E	E
96.63	100	100	86.5	100	E	E	E	E	E	E	E	E	E		E	E	S	E	E	E
97.53	100	95.5	100	95.5	E	E	E	E	E	E	E	S	E		E	E	E	S	E	E
65.5	100	100	46	40	E	E	E	E	E	E	E	E	E		S	S	I	E	I	I
100	100	100	100	100	E	E	E	E	E	E	E	E	E		E	E	E	E	E	E
94.83	100	91	86.5	100	E	E	E	E	E	E	E	S	S		E	E	S	E	E	E
100	100	100	100	100	E	E	E	E	E	E	E	E	E		E	E	E	E	E	E
100	100	100	100	100	E	E	E	E	E	E	E	E	E		E	E	E	E	E	E
96.63	100	100	86.5	100	E	E	E	E	E	E	E	E	E		E	E	S	E	E	E
99.1	100	95.5	100	100	E	E	E	E	E	E	E	S	E		E	E	E	E	E	E
86.88	100	100	100	62.5	E	E	E	E	E	E	E	E	E		E	E	E	E	E	I
100	100	100	100	100	E	E	E	E	E	E	E	E	E		E	E	E	E	E	E

Fig. 15. Distribution of the D_s grade of development of the competences acquired by all students at the end of the study program

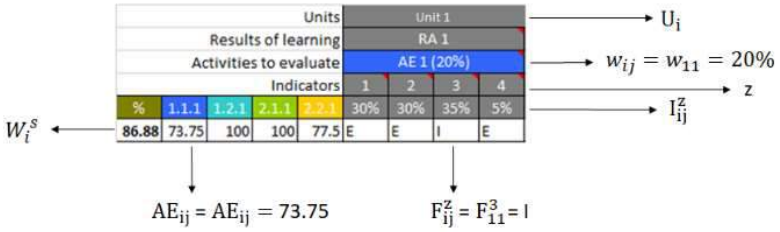


Fig. 16. Symbolic representation of the elements in Table 12

Figure 10b shows the distribution of the grade of development of competencies concentrated in an interval with values very close to the maximum weight value corresponding to the learning activity

H_{12}. Only the grade of a few students places outside this range. Such a result implies that they all performed their activities with the best attitude but with little knowledge and with the procedure attempted most of the time (Figure 8b). All students completed the whole procedure due to the coincidence of a positive attitude that favored obstacle elimination, in this case the barely small knowledge, and which also promoted pursuing objectives.

Otherwise, in Figure 10h the distribution of the grade of development of competencies is concentrated in an interval with values above the mean of the maximum weight of corresponding to activity H_{33} (Figure 7), only the score of a few students is outside this range. This means that despite knowledge being predominantly sufficient, most of the students did not totally take on the procedure because the attitude was oscillating between the negative and the positive (Figure 8h).

There is a transition point between a negative and positive attitude, called a neutral attitude; it is not very common. Our fuzzy inference system characterizes the attitude with a small interval that defines the transition from negative to positive. The neutral attitude renders the subject's action ineffective, causing a loss of the notion of time or continuity from the left-off point.

The teacher can analyze the results of Figures 8-10 and being aware at detail what is happening with the elements that rate measure the performance of students while achieving the learning activities.

The teacher could identify problems in assessment techniques or at teaching-learning strategies with the advantage of modifying or changing them to achieve other objectives during the learning sequence. It is necessary to clarify that the results presented in Figure 10 and Figure 11 do not determine whether the students are competent or not. At the end of the study program, it is possible to assert know to what grade a student is competent or not competent (Figure 12).

In addition, it is possible to ascertain the competences that a student or a group developed through Table 7.

It is also possible to learn from Figure 12, that the maximum grade of development of the competences acquired by the students was less than 65%. Only 13 students who represent 26% of the total number in the group completed learning activities with a grade grater that 60% but less than 65%. A 75% of the total number of students developed the learning activities with an insufficient score, that is, with a grade of achievement lower than 60%. The conclusion is that the study program had many deficiencies in teaching-learning strategies around the development of learning activities and possibly in evaluation techniques. It would be necessary to review, what was what happened? This question can be only answered once the study program completes.

Moreover, it is recommended to answer this question after having completed each one of the learning activities. The average grade of development of the competences acquired by the group was 58.37% (Equation (9)), which is too low. Recall that this study case considers simulated input values, it is obvious for other different input values the average grade of development of the competencies acquired by the group will change. It could be higher or lower than that obtained in this test.

In addition, if the competencies are aligned with the learning activities in a different way to that presented in Table 7 then the results of alignment presented in Table 8 would attain other values, so the weights shown in Figure 7 would modify, this in turn will produce a different final result than presented in Figures 10, 11 and 12. In order to explain these effects, we consider different weights to those presented in Figure 7, which are hypothetically generated, that is, these weights are not associated with an alignment matrix but we assume that they should be (Figure 14).

Then, as described, we can obtain the grades of development of competencies. The task completes through Equation (6), Equation (7), and Equation (8) by using hypothetical weights (Figure 12) and simulated input data (Tables 9, 10, and 11). Then we proceed to compare them with those shown in Figure 11 and Figure 12 through their distribution in frequency graphs (see Figure 14 and Figure 15, respectively). This proves that whenever the weights v_{ij} and u_{ij} change the results also change. Furthermore, we obtained the average grade of development of the acquired group's competencies using Equation (11). We found a value of 24.88%, which turns out to be

much lower than a calculated one of 58.34% obtained for the simulated data study case. Recall that the simulated data is generated randomly, thus precluding any significance or relationship either with the student's behavior or with the learning activities.

Equation (12) allowed calculation of the number $\mathrm{N}_{\mathrm{H}}^{+}$ of students who acquired their competencies at the end of a learning activity H_{13} with a grade of development P greater than 60%, that is $\mathrm{P} > 60\%$ (30% of u_{13}).

If we take the D_s values corresponding to the study case based on simulated data, we obtain $\mathrm{N}_{\mathrm{H}}^{+} = 40$, which typify 80% of n_s.

Likewise, if we aimed at calculating the number $\mathrm{N}_{\mathrm{H}}^{+}$ of students who acquired their competencies at the end of a learning activity H_{13} with a grade of development P greater than 60%, that is, $\mathrm{P} > 60\%$ (8.52% of u_{13}), but taking D_s values corresponding to the hypothetical study case data, we would have obtained $\mathrm{N}_{\mathrm{H}}^{+} = 38$, which represent 76% of n_s. In both the simulated data case and the hypothetical study cases, the $\mathrm{N}_{\mathrm{H}}^{+}$ values are different.

Correspondingly, if the goal is obtaining the value of N_H^- with a grade of development P less than 60%, that is $\mathrm{P} < 60\%$, then the result will be the complement of the value of $\mathrm{N}_{\mathrm{H}}^{+}$ obtained in both cases, but it is also possible to obtain N_H^- through Equation (13). This exercise can be illustrative whenever a comparison of the grade of development among learning activities H_{ij} of the same U_{i} requires. Also, if the aim is to make comparisons among the grades of development at the end of the units the task could be achieved through using D and Equations (14) and (15). In addition, to know the grade of development of the competencies acquired, it is mandatory knowing which competences the student developed. This is only possible through obtaining the alignment matrix.

3.2 Study Case Based on Real Data

In the addressed education evaluation system, a study program can compose several units (Unit), and each one can associate several learning outcomes (RA) associated with their corresponding learning activity (AE). However, it is possible that each learning activity (AE) be divided into one or more sub-activities called “indicators”. Adapting data from Romero-Escobar in [34], we can get a feel for an experimental study case.

The reference includes the results obtained in the study program corresponding to the subject "Operating Systems Management", which is pre-defined by the educational system in two units that comprise two learning outcomes (RA), and each one of them associating two activities to evaluate (AE) with 4 indicators each (Table 12). Percentage value of indicators is also pre-defined by the educational system.

In the Table 12, the scores of the 27 enrolled students identify each one by means of a letter. The letter 'E' stands for 'Competent', the letter 'S' signifies 'Sufficient' and the letter 'I' connotes 'Not Competent'. The official grading system does not consider other values for the score of the students. The meaning of 1.1.1 is: Unit 1, Result of learning 1 (RA 1) and Activity to evaluate 1 (AE 1), respectively. In addition, the report by [12], the evaluation of the activities (AE) was not carried out in such a way that the adopted techniques could be considered as a factor inducing evidence of the display of the indicators considered in the present study (knowledge, procedure, attitude).

In order to explain how the current education evaluation scheme works, we present in Figure 16 the symbolic representation of the elements of Table 12 describes as follows: the symbol U_{i}, for $i = 1 \ldots n$, stands for the ith unit, using $\mathrm{AE}_{\mathrm{ij}}$, for $j = 1 \ldots m$, we denote the jth activity of the unit U_{i} to evaluated, the symbol $\mathrm{F}_{\mathrm{ij}}^{\mathrm{z}}$, for $z = 1,2,3, \ldots$, will stands for the zth indicator of the jth activity of the unit U_{i} to be evaluated, w_{ij} standing for the weight of the activity $\mathrm{AE}_{\mathrm{ij}}$, and finally, $\mathrm{I}_{\mathrm{ij}}^{\mathrm{z}}$ taken to symbolize the weight corresponds to $\mathrm{F}_{\mathrm{ij}}^{\mathrm{z}}$. W_i^s is the percentage value corresponding to U_{i}

Using the listed notation convention, we go ahead and furnish the mathematical expressions that the education evaluation system takes into account at obtaining the results of Table 12. Alongside this, we elucidate what’s wrong with this evaluation scheme. Such a protocol assigns each evaluation result $\mathrm{AE}_{\mathrm{ij}}$ of an activity of the sth student as follows:

$$\mathrm{AE}_{\mathrm{ij}} = \sum_{z=1}^{n_z} G, \tag{16}$$

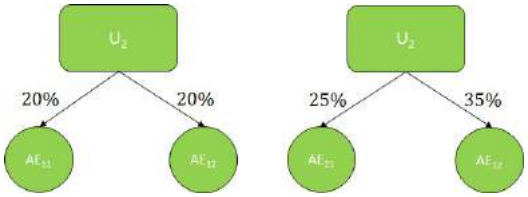


Fig. 17. Weight w_{ij} taken from Figure 16 corresponding to each activity $\mathrm{AE_{ij}}$

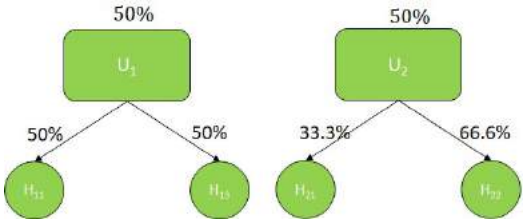


Fig. 18. Weight values, v_{ij} and u_i, obtained by Equations (4) and (5) corresponding to the alignment matrix presented in Table 13

Table 13. Alignment matrix: attributes as they integrate in the learning activities corresponding to the simulation study case

	U1			**U2**	
$\mathbf{C_{hk}}$	H11	H12	$\mathbf{C_{hk}}$	H21	H22
C11		1	**C22**	1	
C16	1		**C33**		1
C44		1	**C66**	1	
C55		1	**C73**		1
C93	1		**C82**		1
C11 1	1		**C91**		1

Table 14. Total number of attributes integrated in the leaning activities and units comprised in the alignment matrix presented in Table 13

Unidad	$V^{(\mathrm{H}_{i1})}$	$V^{(\mathrm{H}_{i2})}$	V^{U_i}
1	3	3	**6**
2	2	4	**6**

where $G = \begin{cases} \mathrm{I}_{\mathrm{ij}}^{\mathrm{Z}} & if \quad \mathrm{F}_{\mathrm{ij}}^{\mathrm{Z}} = \mathrm{E}, \\ 0.775 \times \mathrm{I}_{\mathrm{ij}}^{\mathrm{Z}} & if \quad \mathrm{F}_{\mathrm{ij}}^{\mathrm{Z}} = \mathrm{S}, \\ 0.250 \times \mathrm{I}_{\mathrm{ij}}^{\mathrm{Z}} & if \quad \mathrm{F}_{\mathrm{ij}}^{\mathrm{Z}} = \mathrm{I}, \end{cases}$

and where n_z is the total number of indicators in $\mathrm{AE_{ij}}$. In the Equation (16), we can be aware that the scheme of evaluation of the educational system considers only three qualitative values as rating, I, S, and E, the students who had a rating equal to 'I' will have only 25% of the value of $\mathrm{I}_{\mathrm{ij}}^{\mathrm{Z}}$.

Likewise, the students who had a rating equal to 'S' will credit for only 77.5% of the value of $\mathrm{I}_{\mathrm{ij}}^{\mathrm{Z}}$. It should be noted that the percentage represented by $\mathrm{I}_{\mathrm{ij}}^{\mathrm{Z}}$ determines by the teachers based on their own criteria and can be arbitrarily modified during the subject's program, if the teacher so wishes. In the other hand, If the teacher does not divide the activity to be evaluated (AE) into indicators then the official scheme evaluates the activity $\mathrm{AE_{ij}}$ as follows:

$$\mathrm{AE_{ij}} = \begin{cases} w_{ij} & if \quad \mathrm{F_{ij}} = \mathrm{E}, \\ 0.775 \times w_{ij} & if \quad \mathrm{F_{ij}} = \mathrm{S}, \\ 0.250 \times w_{ij} & if \quad \mathrm{F_{ij}} = \mathrm{I}, \end{cases} \tag{17}$$

where the $\mathrm{F_{ij}}$ value corresponding to rating to the activity $\mathrm{AE_{ij}}$. For the unit, W_i^s, the educational system evaluates the sth student as follows:

$$W_i^s = \left\{\sum_{j=1}^{m} \mathrm{AE_{ij}}\right\} \times w_{ij}. \tag{18}$$

The final evaluation, W^s, of the sth student is given by:

$$W^s = \sum_{i=1}^{n} W_i^s, \tag{19}$$

and the average rating of the group is obtained by:

$$W = \frac{\sum_{s=1}^{27} W^s}{27 \times 100}, \tag{20}$$

which is $W = 8.62$. Romero-Escobar [12] did not consider a formal relationship between competencies and learning activities to create the alignment matrix. Therefore, in contrast to the entries presented in Table 6, it is not possible to obtain the total number of attributes integrated into

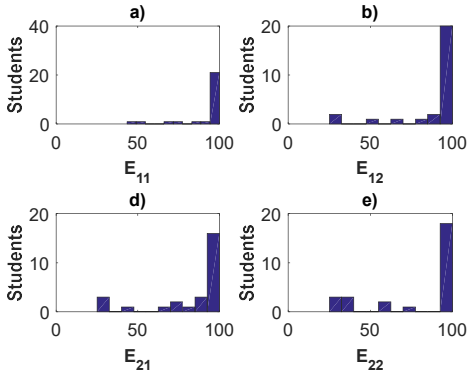


Fig. 19. Distribution of the scores of 27 students as presented in Table 12

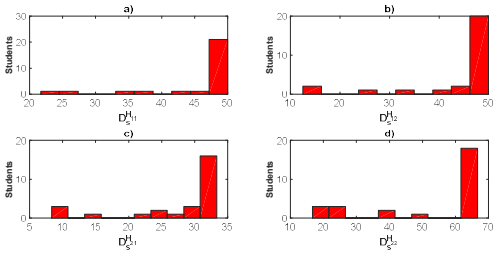


Fig. 20. Distribution of the grade of development of the competencies acquired by the sth student in the learning activity $\mathrm{H_{ij}}$

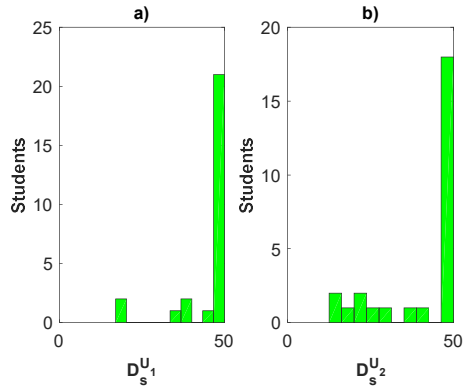


Fig. 21. Distribution of the $D_s^{(U_i)}$ grade of development of the competencies acquired by the students when ending all the learning activities of unit U_i

each activity to evaluate $\mathrm{AE_{ij}}$. Neither could we include the weights corresponding to each activity $\mathrm{AE_{ij}}$ to as shown in Figure 6. Besides, this restriction extends to the values of each unit $\mathrm{U_i}$. However, we can form a tree like diagram such as that appearing in Figure 6, provided we use as weights w_{ij} the ones taken from Figure 16.

The weights shown in Figure 17 are not the result of an alignment of competencies with the activities $\mathrm{AE_{ij}}$ to be evaluated. The official educational grading scheme relies on Equations (18) and (19) to obtain the rating after completing a unit and once the end the study program achieves (Figure 16), respectively. In other words, the rating of the unit W_i^s is the sum of ratings corresponding to each activity $\mathrm{AE_{ij}}$ and the W^s final rating is the sum of ratings corresponding to each unit W_i^s.

To acquire the comportments of the $\mathrm{AE_{ij}}$ values associating with the real data study case, we consider the E_{ij} efficiencies at which the students performed the learning activity $\mathrm{H_{ij}}$, as pertaining proxy values, that is, we take $E_{ij} = \mathrm{AE_{ij}}$. Recall that our method estimates E_{ij} values using a fuzzy inference system (Figure 4). Besides, since for the real data study case an alignment matrix is not available, we take the one corresponding to the simulated data study case as a possible surrogate (see Table 8). Given that, we only consider the first and second unit, both with the first and second corresponding activity $\mathrm{H_{ij}}$. Similarly, while analyzing the study case based on simulated data, the possibility of a random alignment was considered. It corresponds to the utmost subjective scenario compared to a consistent evaluation, so for this essay, taking as a base the random alignment associating with the case mentioned above renders convenient.

We calculate the total number of attributes integrated in the learning activity $\mathrm{H_{ij}}$ through Equation (1) and the total number of attributes integrated into U_i through Equation (2) (Table 14). And its weights v_{ij} and u_i obtained through Equations (4) and (5), respectively (Figure 18).

Let's show in Figure 19 the distribution of E_{ij} presented in a frequency graph. In panel (a) the distribution is concentrated in the highest value, 100%, and only three students presented evaluations lower than 100%. This means that the students performed the learning activity $\mathrm{H_{11}}$ skillfully and that the evaluation technique and the

teaching-learning strategies were correct. In addition, more than 71% of the students developed their skills with a grade equal to 100% in the learning activities H_{12}, H_{21} and H_{22}. In general, evaluation techniques and teaching-learning strategies were the most appropriate. Values of E_{ij} less than 77% correspond only to 11% of the total number of students.

Considering the weight values presented in Figure 18, we calculated through Equation (6) the grade $\mathrm{D}_s^{(\mathrm{H_{ij}})}$ of development of the competencies acquired by the sth student, while achieving the learning activity $\mathrm{H_{ij}}$ (Figure 20).

In all the learning activities presented in Figure 20, a 71% of the total number of students developed their competencies with a grade equal to 100%. Furthermore, Figure 20 shows that an 18% of the total number of students developed the competencies with a grade lower than 100% but higher than 70%. A quite significant result for the educational system and for the curricular achievement of the teacher. If the evaluation techniques in [12] had considered the elements to evaluate presented in the simulated data study case, then, it could be said, that almost all the students acquired all the knowledge, that the procedures were achieved and that the attitude of the students was always positive throughout the study program. The results could always be biased towards the maximum grade of development using the qualifications of any of the groups or subjects presented in [12], due to the evaluation criteria that the educational system manages.

To calculate $\mathrm{D}_s^{(\mathrm{U}_i)}$ the grade of development of the competencies acquired by the sth student when ending all the learning activities of unit U_i, we use Equation (7) and the weight values presented in Figure 18 (see Figure 21).

In the same way, we obtain the D_s grade of development of competencies acquired by students throughout the study program (Figure 22) through the Equation (8).

Finally, we obtain the average percentage of the grade of development of the competencies acquired by the group through Equation (11), which is $\mathrm{D} = 8.67$. This value is scarcely differing to the one obtained by the educational evaluation system because the values of the weights u_i are in

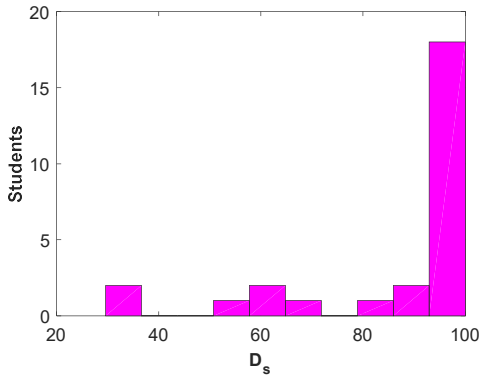


Fig. 22. Distribution of the D_s grade of development of the competencies acquired by the students throughout the study program

almost equal proportion (see Figure 17 and Figure 18).

4 Discussion

The current scheme that the Mexican Middle Higher Educational System (MMHES) addresses at evaluating students overlooks the effort of the teacher and the students. This is mainly explaining because referred scheme standardizes the evaluation of $\mathrm{F}_{\mathrm{ij}}^{\mathrm{z}}$ (Figure 16) to a percentage value of either the weights $\mathrm{I}_{\mathrm{ij}}^{\mathrm{z}}$ (Equation (16)) or w_{ij} (Equation (17)).

We can interpret it being subjective that a rating with a value of 8 entailing a value very close to 10 could be associated with the letter 'E' that regularly links to an excellent performance. Moreover, emphasizing on subjectivity a rating value of 8 being close to 6 could be also associated with the letter 'S' linking to sufficient. By the same token, we can trust that a rating value of 7 could be either associated with the letter 'S' or the letter 'I' for incompetent. The point is that the ranges of rating values that can be associated with the letter 'E', 'S' or 'I' are not known by default. Therefore, there is no clear evidence that the ranges of scores associated with the letter 'E', 'S' and 'I' are not granted subjectively. As a results, we can expect that a student who gets a rating of 8 performed similarly to one achieving a rating of 10, if both are graded with the letter 'E', the one rating 8, would be too overvalued. In the worst case, in no way

could it be known with the highest grade of certainty about the student's real progress in the development of their activities, whether they are based on the acquisition of competences or not. Furthermore, the overvaluation could be increased, for example: in the case of that a rating of 8 is associated to the letter $\mathrm{F}_{\mathrm{ij}}^{\mathrm{z}} = \mathrm{E}$ and its weight value $\mathrm{I}_{\mathrm{ij}}^{\mathrm{z}}$ (Figure 16) be the largest of all weights $\mathrm{I}_{\mathrm{ij}}^{\mathrm{z}}$ corresponding to $\mathrm{AE}_{\mathrm{ij}}$.

Our method conceives alignments as a formal mathematical procedure that establish a relationship between the attributes of the competences with the learning activities. In our approach the weights v_{ij} and u_{ij} assigned to each activity and unit are determined by Equations (4) and (5) one to one. Correspondingly, the attributes that ought to evaluate in each of the learning activities are known a priory (Table 5). Another feature of our method is that it allows a flexible slant for teachers as they are free to choose the most appropriate technique (Table 2) to evaluating each H_{ij} learning activity. In addition, presently offered evaluation scheme establishes the criteria to obtain evidence of student performance during the development of activities and achieves this task considering only three elements, knowledge, procedure and attitude.

Our fuzzy inference system also entails a synthetic evaluation of efficiency as it integrates the three elements in a single quantitative index value E_{ij}. Moreover, the knowledge that the student can handle or could display while achieving a given task can be evaluated both qualitatively and quantitatively, likewise, the procedures undertaken by the students can also be evaluated qualitatively and quantitatively. The evaluation of the attitude bases on the observation but it is valued by the effects that it may have on an individual [28], [32-33]. In summary our fuzzy system is built upon a set of rules so that it determines the relationship among the three essential elements: knowledge, procedure and attitude, thereby producing a result with less a subjective burden while assigning the rating of a student.

In the simulated data study case, efficiency under which students developed their competences evaluates by means of a fuzzy inference system. Through the conforming alignment matrix, that procedure identifies both the attributes that require an evaluation and their inclosing activities and units (Table 5). Then using Equations (6), (7), (8) and (11) it will be possible to determine to what grade the students acquired their competencies at the end of each learning activity, unit and study program, correspondingly. As we learnt from the real study case, the activities developed by students are evaluated by the teacher's own criteria, without clear evidence on how they achieved tasks. In addition, the evaluations can be overvalued or undervalued, this depends on the teacher and the educational evaluation system criterion. Certainly, it is not possible to know about the actual development of student learning.

Contributions by Romero-Escobar in [8] and Bedoya-Ruiz in [27] aim to evaluate the competencies acquired by students. But on spite of its relevance the approach by Romero-Escobar in [8] does not contemplate a formal mathematical method for the alignment of competencies, and neither involves a fuzzy system at sustaining the proposed method of evaluation. Bedoya-Ruiz [27] uses fuzzy logic to construct rubrics but does not develop a formal mathematical method to align competencies. Present approach attempts to fill these gaps. Blending the incipient approaches Romero-Escobar in [8] and Bedoya-Ruiz in [27] we integrate a Hybrid construct where the outstanding paradigm for management of vagueness provided by fuzzy logic integrates to a competencies-alignment matrix. The output composes a mathematically based tool for the evaluation of the grade of development of competencies by students enrolling at the MMHES.

According to obtained results, it turns out that present fuzzy system makes an acceptable interpretation if what it intends to be understood as an evaluation process for the establishment of competencies. Nevertheless, it is worth pointing out that the greatest difficulty that arises in the implementation of our method is the construction of the fuzzy system itself. If you want to change the rules to inference rules as to capture the effects induced by another type of attitude such as those presented in [28], special care must be taken in the choice of membership functions and their scale of values. They must be constructed through a fuzzy partition (see T_1 and T_2 in Appendix). Similarly, the

assembly of the definition of the rule base to be used must consider all possible anticipating cases and must reflect the desired level of demand of the institution.

Our method allows to process the evidence collected in the different moments of assessment, regardless of the technique that teachers choose to evaluate the performance of the activities developed by the students. Our results show that teachers can monitor the development of students' competencies, know what their points of greatest development are, as well as, ascertaining which competencies they have difficulties acquiring.

5 Conclusion

This work presents a first version of a protocol aimed at evaluating the grade of development of the competencies acquired by students enrolling at the Mexican Higher Middle Education System. Entailed scheme contemplates the essential elements sustaining an objective evaluation. The associated alignment of competencies to study units and learning activities relies on a formal mathematical relationship. Offered construct establishes as a condition that the teacher typifies the alignment before starting a subject and precludes the possibility of subsequent later modifications.

Composing alignment method deters inconsistent practices by students or teachers that impair efficiency thereby, it attempts to amend deficiencies detected on this matter at the whole Mexican Middle Higher Educational System. In addition, the alignment method contemplates the estimation of the weight values, v_{ij} and u_{ij} (see Equation (4) and (5), respectively) that sustain evaluation depending on a numerical scale, while also precluding the possibility that they are modified by non-teaching personal.

Teaching practice focuses on transmitting “knowledge” to students through teaching and learning activities. The evaluation of competencies is a process of extracting suitable evidence. In accordance, the teacher's efforts should aim towards correctly identifying the contents to be considered. Relevant to his task becomes the organization and dosage in time, the conscientious selection of exercises, and the clarification of the correspondence that they keep with the selected content.

Besides, teachers must search for evaluation tools (Table 2) that offer greater reliability this thorough assigning numerical scores, which in our scheme typify as "evidence". Our fuzzy inference system integrates the evidence, which typify as elements supporting efficiency, namely knowledge, procedure and the attitude of the student.

The nature of attitudes allowed the design of fuzzy instruments for their evaluation, such as the attitudinal evaluation scale that implements through membership functions and fuzzy rules. The attitudinal assessment represents the availability of criteria agreed between experts in the area.

On this matter, we stood by requirements in [28]. In turn, we considered that knowledge concerns to facts, concepts and principles; that requires students to consider and "know how to say", for example: Is it? What is it? What is it like? What are its most significant characteristics? When did it happen? Where did it happen? How long did it last? Why did it happen or behave like this? How does it resemble? What differences does it have with? It can also be valuated qualitatively and quantitatively, which allowed the design of fuzzy evaluation tools.

In the same way, we can assess the procedure, for example: knowing how to add fractions, preparing a summary or essay, and so on. In general, fuzzy systems are a valuable tool because they model the experts' criteria through in linguistic variables and inference fuzzy rules.

Future work will attempt to develop a comprehensive evaluation system where fuzzy instruments' design is adapted and considered to appraise any attitudes pertaining to the educational process [28] and intended to assess acquisition of values.

Also, to address the design of fuzzy tools to assess other elements as presented in Figure 2 but including more details. And, in a more general way, to develop a system that addresses firsthand the Mexican Higher Middle Education System. On view of present results, we do not preclude the possibility that the foreseen comprehensive evaluation system could suit similar evaluation endeavors in educational systems elsewhere.

6 Appendix. Some Concepts of Fuzzy Set Theory

We present some concepts and definitions of fuzzy set theory that are relevant to the contents of the methods section, particularly on building the addressed fuzzy inference system that conforms to the approach in references [34] and [36-38].

6.1 Fuzzy Sets and Associating Memberships Functions

Let U be a universal set that contains all the possible elements of concern in each context or application. A fuzzy set A is defined through:

$$A = \{(x, \mu_A(x)) | x \in U\}, \quad \text{(A1}$$

where $(x, \mu_A(x))$ is an ordered pair composing an element x of U, and being $\mu_A(x)$ a membership function that assigns to x a numerical value in the interval $[0,1]$, namely:

$$\mu_A(x) = \begin{cases} 1 & if\ x \in A, \\ 0 & otherwise. \end{cases} \quad \text{(A2)}$$

An extensive collection of membership functions is available to choose from, but forms can conceive to suit specific requirements.

A triangular fuzzy number is a fuzzy set $A \subset R$ that characterizes by a membership function $\mu_A: \mathrm{R} \to [0,1]$ defined by:

$$\mu_A(x): \begin{cases} 0 & if \quad x \le a \\ \dfrac{x-a}{m-a} & if \quad a \le x \le m \\ \dfrac{b-x}{b-m} & if \quad m \le x \le b \\ 0 & if \quad otherwise \end{cases} \quad \text{(A3)}$$

where a, b and m are real number such that a and b are the lower and upper limits of variation of x, that is, $a \le x \le b$, and m is the modal value. In the present settings, the triangular fuzzy number implements through the function $trimf(x, [a, m, b])$ of Matlab 2016b.

A Gaussian fuzzy number is a fuzzy set $A \subset R$ that identifies by a membership function $\mu_A: \mathrm{R} \to [0,1]$ defined as follows:

$$\mu_A(x): e^{\frac{-(x-m)^2}{2\sigma^2}}, \quad \text{(A4)}$$

where m is the medium value and σ stands for the wideness of the bell, both m and σ are real numbers. Here a Gaussian fuzzy number achieves through the function $gaussmf(x, [\sigma, m])$ of Matlab 2016b.

An s-shaped fuzzy number is a fuzzy set $A \subset R$ that typifies by a membership function $\mu_A: \mathrm{R} \to [0,1]$, which define as follows:

$$\mu_A(x): \begin{cases} 0 & if \quad x \le a, \\ 2\left[\dfrac{x-a}{b-a}\right]^2 & if \quad a \le x \le m, \\ 1 - 2\left[\dfrac{x-b}{b-a}\right]^2 & if \quad m < x < b, \\ 1 & if \quad x \ge b, \end{cases} \quad \text{(A5)}$$

where a and b stand one to one for the lower and upper limits of the range of x, and m picks out such that $a < m < b$ stands for the inflection point of $\mu_A(x)$. Usually, in practice a value $m = (a+b)/2$ chooses. The greater the distance $a - b$, the slower the growth of $\mu_A(x)$. In present work the s-shaped fuzzy number implements through the function $smf(x, [a, b])$ of Matlab 2016b.

A z-shaped fuzzy number is a fuzzy set $A \subset R$ associating with by a membership function $\mu_A: \mathrm{R} \to [0,1]$ defined as follows:

$$\mu_A(x): \begin{cases} 1 & if \quad x \le a, \\ 1 - 2\left[\dfrac{x-a}{b-a}\right]^2 & if \quad a \le x \le \dfrac{a+b}{2}, \\ 2\left[\dfrac{x-b}{b-a}\right]^2 & if \quad \dfrac{a+b}{2} \le x \le b, \\ 0 & if \quad x \ge b, \end{cases} \quad \text{(A6)}$$

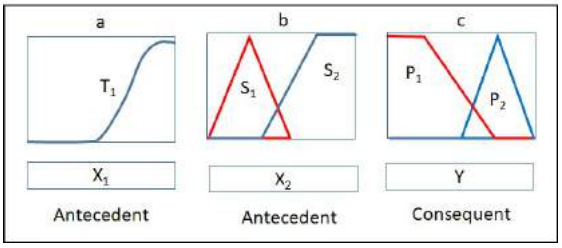


Fig. A1. Linguistic variables $\mathbf{X}_1, \mathbf{X}_2$ and $\mathbf{Y}$ characterized by fuzzy sets (a) $\mu_{\mathbf{T}_1}$ (b) $\{\mu_{\mathbf{S}_1}, \mu_{\mathbf{S}_2}\}$ and (c) $\{\mu_{\mathbf{P}_1}, \mu_{\mathbf{P}_2}\}$ respectively

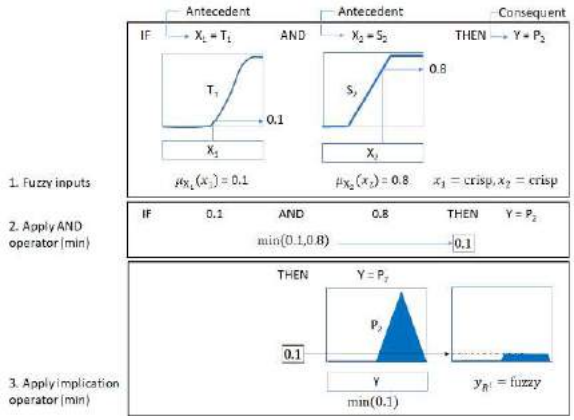


Fig. A2. Interpreting the if-then rule as a three-part process

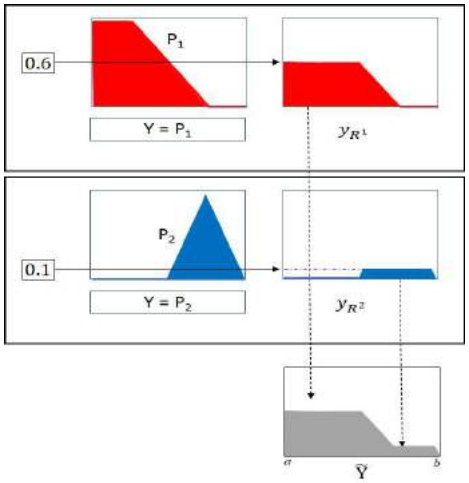


Fig. A3. The aggregation process illustrated for the case n_R=2

where a and b are real numbers typifying the lower and upper limits for x that is, $a < x < b$. The greater the distance $a - b$, the faster $\mu_A(x)$ grows. The z-shaped fuzzy number obtains here through the function $zmf(x, [a, b])$ in Matlab 2016b.

A trapezoidal fuzzy number is a fuzzy set $A \subset R$ that associates with a membership function μ_A: R → [0,1] defined as follows:

$$\mu_A(x): \begin{cases} 0 & if \quad x \le a, \\ \dfrac{x-a}{b-a} & if \quad a \le x \le b, \\ 1 & if \quad b \le x \le c, \\ \dfrac{d-x}{d-c} & if \quad c \le x \le d, \\ 0 & if \quad d \le x, \end{cases} \tag{A7}$$

where a, b, c and d are real numbers defined such that $a < b < c < d$. The parameters a and b locate the "feet" of the trapezoid and the parameters d and c locate the associating "shoulders". Trapezoidal fuzzy number computes here through the function $trapmf(x, [a, b, c, d])$ of Matlab 2016b.

6.2 Logical Operations

In fuzzy logic theory a proposition stating: if x is A then y is B, links to a characteristic function $\mu_{A\to B}(x, y)$ taking values in the interval [0.1]. Moreover, each if-then rule conceives as a fuzzy set with its characteristic function measuring the grade of truth of the implication relationship between x and y, formally:

$$\mu_{A\to B}(x, y) = \mu_A(x) \cdot \mu_B(y), \tag{A8}$$

For fuzzy sets A and B on a reference set U and characterizing by membership function $\mu_A(x)$ and $\mu_B(x)$, one to one, the standard set operations $A \cap B$ or $A \cup B$ define through:

$$\begin{aligned} \mu_{(A\cap B)}(x) &= \min[\mu_A(x), \mu_B(x)] \quad - \quad fuzzy\ intersecction, \\ \mu_{(A\cup B)}(x) &= \max[\mu_A(x), \mu_B(x)] \quad - \quad fuzzy\ union. \end{aligned} \tag{A9}$$

6.3 Fuzzy Rules

Fuzzy sets and operators are the subjects and verbs of a fuzzy inference system. The if-then rules compose conditional and consequent statements. Conditional ones define by means of a combination of one or more fuzzy input sets called antecedents. The consequent conceives as an associating output fuzzy set.

As an example, we can consider a fuzzy inference rule with a conditional statement that combines the conjunction of two antecedents, and with a consequent being a single fuzzy set, namely:

$$\text{If } \mathbf{X}_1 \text{ is } \mathbf{T}_1 \text{ and } \mathbf{X}_2 \text{ is } \mathbf{S}_2 \text{ then } \mathbf{Y} \text{ is } \mathbf{P}_2, \qquad \text{(A10)}$$

where $\mathrm{X}_1, \mathrm{X}_2$ and $\mathbf{Y}$ are linguistic variables, $\mathrm{T}_1, \mathrm{S}_2$ and P_2 are linguistic terms defined by fuzzy sets $\mu_{\mathrm{T}_1}, \mu_{\mathrm{S}_2}$ and μ_{P_2} in the universe $\mathrm{X}_1, \mathrm{X}_2$ and Y respectively. Customarily, in the fuzzy inference systems terminology X_1 and X_2 refer as antecedents and Y as the consequent.

For instance, given that the antecedent X_2 and the consequent Y in Equation (A10) respectively describe by means of the linguistic terms S_2 and P_2, we could correspondingly assume that the linguistic variables X_2 and Y characterize one to one by means of pairs of linguistic terms (S_1 and S_2) and (P_1 and P_2), (see Figure A1).

6.4 Operation of the Mamdani Fuzzy System

A Mamdani fuzzy system conceives as an expert system with approximate reasoning that maps a vector of inputs to a single output (scalar).

In order to explain how such construct operates, we take the example represented by Equation (A10) and Figure (A1). The architecture of this system composes by fourth phases:

a. Phase of fuzzification: identified as the conversion of numeric input values into fuzzy sets. Upon input data the grade of belonging to each of the fuzzy sets $\mu_{\mathbf{T}_1}, \mu_{\mathbf{S}_2}$ and $\mu_{\mathbf{P}_2}$ determines by means of Equation (A3), Equation (A4), Equation (A5), Equation (A6), Equation (A7) or another) (see Figure A2 step 1).
b. Phase of inference: numeric values x_1 and x_2 are made correspond to the input variable X_1 and X_2 respectively (see Figure A2 step 1). The minimum (fuzzy intersection in Equation (A9)) is generally used to evaluate the "and" that connects the propositions associated to the antecedents (see Figure A2 step 2). If the fuzzy rule being evaluated is R^l, where $1 \leq l \leq n_R$, that is, n_R is the total number of rules in conceived fuzzy inference system, the grade of certainty or activation of the antecedents for the current values of the input variables is represented by y_{R^l} which for the case $n_R = 2$, defines as follows:

$$y_{R^l}(x_1, x_2) = \min\{\mu_{\mathbf{X_1}}(x_1), \mu_{\mathbf{X}_2}(x_2)\}, \qquad \text{(A11)}$$

The execution of the rule y_{R^l} (Equation (A11)) is done by applying a fuzzy operator of implication (fuzzy union in Equation (A9)). Then, the output of each rule is the fuzzy set y_{R^l} resulting from the implication (see Figure A2 step 3).

c. Phase of aggregation: unification of the outputs of all rules. We take the membership function of all rule consequents previously clipped or scaled, and combine them into a single fuzzy set, $\widetilde{\mathrm{Y}} = \{(x, \mu_{\widetilde{\mathrm{Y}}}(x)) | x \in U\}$, this becomes (see Figure A3 for an illustration of the case $n_R = 2$)

$$\widetilde{\mathrm{Y}} = y_{R^1}(x_1, x_2) \oplus y_{R^2}(x_1, x_2) \oplus, \qquad \text{(A12)}$$

where $\oplus$ means maximum operator (fuzzy union in Equation (A9)). Although, there are other operators available, the maximum is commonly used to perform this operation.

d. Phase of defuzzification: the input is the aggregate output fuzzy set $\widetilde{\mathrm{Y}}$ (Equation (A12) and the output is a single number y. We selected the centroid defuzzification method, which finds a point representing the center of gravity of the fuzzy set, $\widetilde{\mathrm{Y}}$, on the interval $[a, b]$ (see Figure 3A):

$$y = \frac{\sum_{i=a}^{b} x_i \, \mu_{\widetilde{\mathrm{Y}}}(x_i)}{\sum_{i=a}^{b} \mu_{\widetilde{\mathrm{Y}}}(x_i)}. \qquad \text{(A13)}$$

The selection of the defuzzification method can play a decisive role in the synthesis of fuzzy models for many areas of application. It heavily relies on the expert's input.

References

1. **Moreno, O.T. (2012).** La evaluación de competencias en educación. Sinéctica, No. 39, pp. 1–20.
2. **SEP. (2008).** Reforma integral de la educación media superior: La creación de un sistema nacional de bachillerato en el marco de la diversidad. Secretaría de Educación Pública, DOF: Acuerdo No. 442.
3. **PISA. (2015).** Programa para la evaluación internacional de alumnos. OCDE.
4. **PISA. (2012).** Desempeño de los estudiantes al final de la Educación Media Superior. OCDE.
5. **INEE. (2009).** PISA para docentes: La evaluación como oportunidad de aprendizaje. SEP.
6. **PISA. (2006).** Marco de la Evaluación: Conocimientos y habilidades en Ciencias, Matemáticas y Lectura (PISA). OCDE.
7. **Ferrández, A. (1997).** El perfil profesional de los formadores. Universidad Autónoma de Barcelona, Departamento de Pedagogía Aplicada.
8. **Ma J., Zhou, D. (2000).** Fuzzy set approach to the assessment of student-centered learning. IEEE Transactions on Education, Vol. 43, No. 2, pp. 237–241. DOI: 10.1109/13.848079.
9. **Nykänen, O. (2006).** Inducing Fuzzy Models for Student Classification. Educational Technology & Society, Vol. 9, No. 2, pp. 223–234.
10. **Huapaya, C.R., Lizarralde, F.A., Arona, G.M. (2012).** Modelo basado en Lógica Difusa para el Diagnóstico Cognitivo del Estudiante. Formación universitaria, Vol. 5, No. 1, pp. 13–20. DOI: 10.4067/S0718-50062012000100003.
11. **Arroyo, B., Antolínez, N. (2015).** La Lógica Difusa como herramienta de evaluación en el sector universitario. Alteridad, Vol. 10, No. 2, pp. 132–145. DOI: 10.17163/alt.v10n2.2015.01.
12. **Romero-Escobar, H.M. (2018).** Modelo para alinear las competencias del docente, del alumno y las requeridas por la instrucción, y mejorar la calidad de la educación en el nivel de educación media superior. Tesis de doctorado. Universidad Iberoamericana.
13. **Biggs, J.B. (2005).** Calidad del aprendizaje universitario. Narcea Ediciones.
14. **Vargas, H., Heradio, R., Chacon, J., De la Torre, L., Farias, G., Galan, D., Dormido, S. (2019).** Automated assessment and monitoring support for competency-based courses. IEEE, Vol. 7, pp. 41043-41051. DOI: 10.1109/ACCESS.2019.2908160.
15. **Valdez, E., Nirvana, V. (2007).** Análisis de lineamientos y programas de difusión de actividades educativas, culturales y deportivas del Cobach Villa de Seris, **Castañeda, R.D., ed.**, Aprender Investigando: Formulación de proyectos de investigación en comunicación educativa.
16. **Alonso-García, C.M., Gallego-Gil, D.J. (2010).** Los estilos de aprendizaje como competencias para el estudio, el trabajo y la vida. Revista de Estilos de Aprendizaje, Vol. 6, No. 6, 4–22.
17. **García-Retana, J.A. (2011).** Modelo educativo basado en competencias: Importancia y necesidad. Revista Electrónica Actualidades Investigativas en Educación, Vol. 11, No. 3, pp. 1–24.
18. **Zadeh, L.A. (1996)**. Fuzzy logic = computing with words. IEEE Transactions on Fuzzy Systems, Vol. 4, No. 2, pp. 103–111. DOI: 10.1109/91.493904.
19. **Sanchez, M.A., Castillo, O., Castro, J.R., Melin, P. (2014).** Fuzzy granular gravitational clustering algorithm for multivariate data. Information Sciences, Vol. 279, pp. 498–511. DOI: 10.1016/j.ins.2014.04.005.
20. **Castillo, O., Cervantes, L., Soria, J., Sánchez, M., Castro, J.R. (2016)**. A generalized type-2 fuzzy granular approach with applications to aerospace. Information. Science, Vol. 354, pp. 165–177. DOI: 10.1016/j.ins.2016.03.001.
21. **Cervantes, L., Castillo, O. (2015).** Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control. Information Sciences, Vol. 324, pp. 247-256. DOI: 10.1016/j.ins.2015.06.047.

22. **Moreno, J.E., Sánchez, M.A., Mendoza, O., Rodríguez-Díaz, A., Castillo, O., Melin, P., Castro, J.R. (2020).** Design of an interval Type-2 fuzzy model with justifiable uncertainty. Information Sciences, Vol. 513, pp. 206–221. DOI: 10.1016/j.ins.2019.10.042.
23. **Weon, S., Kim, J. (2001).** Learning achievement evaluation strategy using fuzzy membership function. 31st Annual Frontiers in Education Conference, Impact on Engineering and Science Education, Vol. 1, pp. T3A–19. DOI: 10.1109/FIE.2001.963904.
24. **Bai, S.M., Chen, S.M. (2008).** Evaluating students' learning achievement using fuzzy membership functions and fuzzy rules. Expert Systems with Applications, Vol. 34, No. 1, pp. 399–410. DOI: 10.1016/j.eswa.2006.09.010.
25. **Saleh, I., Kim, S. (2009).** A fuzzy system for evaluating students' learning achievement. Expert Systems with Applications, Vol. 36, No. 3, pp. 6236–6243. DOI: 10.1016/j.eswa.2008.07.088.
26. **Yadav, R.S., Soni, A.K., Pal, S. (2014).** A Study of Academic Performance Evaluation Using Fuzzy Logic Techniques. International Conference on Computing for Sustainable Global Development, INDIACom, pp. 48–53. DOI: 10.1109/IndiaCom.2014.6828010.
27. **Bedoya-Ruiz. D.P. (2014).** Evaluación de aprendizaje por competencias utilizando lógica difusa. Tesis de Maestría, Universidad de Antioquia, Medellín.
28. **Castillero Mimenza, O. (2018).** Los 15 tipos de actitudes, y cómo nos definen. Psicología y Mente.
29. **Churches, A. (2009).** Taxonomía de Bloom para la era digital. Eduteka.
30. **Tobón, S. (2010).** Formación integral y competencias, Pensamiento complejo, currículo, didáctica y evaluación, 3rd ed. Centro de Investigación en Formación y Evaluación (CIFE), ECOE Ediciones.
31. **Gonczi, A., Athanasou, J. (1996)**. Instrumentación de la educación basada en competencias. Perspectivas de la teoría y la práctica en Australia. Limusa.
32. **Bohner, G., Wanke M. (2002).** Attitudes and Attitude Change. Social Psychology: A Modular Course, Psychology Press. pp. 1–308. DOI: 10.4324/9781315784786.
33. **Ajzen, I. (2005)**. Attitudes, Personality, and Behavior. Open University Press, McGraw-Hill.
34. **Pedrycz, W., Gomide, F. (1998)**. An introduction to fuzzy sets: Analysis and design. MIT Press.
35. **Wang, L.X., Mendel, J.M. (1992).** Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. IEEE Transactions on Neural Networks, Vol. 3, No. 5, pp. 807–814.
36. **Zadeh, L.A. (1996)**. Fuzzy logic = computing with words. IEEE Transactions on Fuzzy Systems, Vol. 4, No. 2, pp. 103–111. DOI: 10.1109/91.493904.
37. **De Barros, L.C., Bassanezi, R.C. (2010)**. Tópicos de lógica fuzzy e biomatemática, UNICAMP/IMECC.
38. **Leal-Ramirez, C., Echavarría-Heras, H., Villa-Diharce, E. (2020).** Applying fuzzy logic to identify heterogeneity of the allometric response in arithmetical space. Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications, Studies in Computational Intelligence, Vol. 862, Springer, Cham, pp. 11–24. DOI: 10.1007/978-3-030-35445-9_2.

Article received on 01/07/2021; accepted on 16/11/2021.
Corresponding author is Cecilia Leal-Ramírez.

ISSN 2007-9737

A Comparative Study in Machine Learning and Audio Features for Kitchen Sounds Recognition

Alain Manzo-Martínez, Fernando Gaxiola, Graciela Ramírez-Alonso, Fernando Martínez-Reyes

Universidad Autónoma de Chihuahua,
Facultad de Ingeniería,
Mexico

{amanzo, lgaxiola, galonso, fmartine}@uach.mx

Abstract. For the last decades the work on audio recognition has been directed to speech and music, however, an increasing interest for the classification and recognition of acoustic events is observed for the last years. This poses the challenge to determine the identity of sounds, their sources, and the importance of analysing the context of the scenario where they act. The aim of this paper is focused on evaluating the robustness to retain the characteristic information of an acoustic event against the background noise using audio features in the task of identifying acoustic events from a mixture of sounds that are produced in a kitchen environment. A new database of kitchen sounds was built by us, since in the reviewed literature there is no similar benchmark that allows us to evaluate this issue in conditions of 3 decibels for the signal to noise ratio. In our study, we compared two methods of audio features, Multiband Spectral Entropy Signature (MSES) and Mel Frequency Cepstral Coefficients (MFCC). To evaluate the performance of both MSES and MFCC, we used different classifiers such as Similarity Distance, k-Nearest Neighbors, Support Vector Machines and Artificial Neural Networks (ANN). The results showed that MSES supported with an ANN outperforms any other combination of classifiers with MSES or MFCC for getting a better score.

Keywords. Entropy, neural networks, mixture of sounds, MFCC.

1 Introduction

Sounds are around human being everywhere and due to physic properties of the sounds one can heard the acoustics of these. Acoustic events refer to several everyday sounds which are generated in natural or artificial form (namely, the sounds found in the environment of the everyday life, excluding speech and music). The development of an acoustic event recognition system (AERS), contributes to the development of intelligent systems capable to understand sound within a context. These systems are important for real-world applications such as activity monitoring systems [15, 39, 45], ambient assisted living [28, 40, 50], human-computer interaction [8, 41, 42], security surveillance [1, 2], assisted robotics [21, 38, 48], among others.

Automatic recognition of acoustic events in real situations is not an easy task, because the audio captured by microphones contains a mixture of different sources of sound. Recent research work about AERS has focused on two types of classification problems: acoustic events classification for a specific context and acoustic events recognition into contextual classes [52, 53]. The former can be associated to, for instance, the activity recognition in a home environment, where acoustic events can offer information that occur in a specific dwelling space.

The audio information for scene understanding, can be more assertive if they exclusively recognize the acoustic events that occur in a specific place. On the other hand, there is the recognition of human activity from the sounds that occur in different places, for instance, the difference of contextual events between the home and the office. It would be difficult to say what kind of activity is carried out, if the contextual classes of the sounds

are not clear. Besides, not limiting the sounds in the scene will be even more difficult this task.

Preliminary work with AERS adopted approaches used for the processing of speech and music, however, the non-stationary characteristics of the acoustic events made the recognition of events problematic for databases with a great number of sound sources [13]. For example, in speech recognition is common to use a phonetic structure that can be seen as a basic component of voice, therefore, spoken words can be divided into elemental phonemes over which it is possible the application of probabilistic models. Conversely, phoneme based approaches cannot be applied to acoustic events coming from sounds created by a car crash or due to pouring water in a glass. Even, if it is possible to create a dictionary of basic units of these events, modelling signal variation in time would be difficult. The same occurs whit the attempt to compare music and acoustic events because the latter does not exhibit significant stationary patterns such as melody and rhythm [13].

The recognition of acoustic events involves two phases: a feature extraction phase followed by a classification phase. The feature extraction phase allows to play two roles; a dimension reduction role, and a representation role. An AERS uses stationary and non-stationary feature extraction techniques. Most of the features extraction algorithms use a scheme called bag-of-frames. The bag-of-frame approach consists of considering the signal in a blind way, using a systematic and general scheme where the signal is divided into consecutive overlapping frames, from which a vector of features is determined. The features are supposed to represent characteristic information of the signal for the problem at hand. These vectors are then aggregated (hence the “bag”) and fed to the next phase of an audio recognition system [3].

Audio signals have been traditionally characterized by Mel Frequency Cepstral Coefficients (MFCC). The methodology for computing MFCC involves a filter bank that approximates some important properties of the human auditory system. MFCC has been shown to work well for structured sounds such as speech and music [16, 23, 25, 26, 27, 37]. Since MFCC has been successfully used in speech and music applications, some work suggests the use of MFCC for characterizing acoustic events that contains a large and diverse variety of sounds, including those with strong temporal domain [4, 35, 40]. In addition, MFCC are often used by researchers for benchmarking their works.

For the classification phase of an AERS there are different machine learning techniques such as Support Vector Machine (SVM). SVM is a classifier that discriminates the data by creating boundaries between classes rather than estimating class conditional densities, or in other words, that SVM could draw accurate classification rate even if the sample size is small, a common scenario for acoustic event classification [14, 24, 51].

Artificial Neural Networks (ANN) is another machine learning technique being widely used for audio recognition systems. ANN deals with the study and construction of systems able to learn from the data. ANN algorithms infer unknowns from known data a characteristic that might describe acoustic events where there is an acoustic event of interest that need to be differentiated from a mix of sounds. [7, 29, 36].

There are other techniques that can be used to identify acoustic events such as audio signatures recognition. In these technique the challenge is to find the acoustic events that sound similar to the audio that the system captures. The similarity rate is evaluated using a distance function. Audio signatures use two fundamental processes to be determined, a feature extraction process and a modeling process. The latter refers to the minimal compacted representation that can be achieved to describe a signal, but which robustness preserve the model against typical audio degradation [40]. Audio signatures thus work very well on AERS, but the problem is complicated when it is required to identify an acoustic event present in a mixture of sounds. This problem usually leads to apply source separation techniques and machine learning algorithms to treat with the complexity of the signals.

In this work we considered the signals unprocessed. Also, we use no source separation technique because our intention is to evaluate the robustness to retain the characteristic information

of an acoustic event against the background noise using two audio features, MFCC that is the state–of–the–art benchmark and the multiband spectral entropy signature (MSES), a technique that has been successfully used in audio fingerprinting, speech recognition and others applications of audio [6, 9, 10, 11, 12, 30].

In addition, MSES feature has never been studied to recognize acoustic events exclusively for indoor domestic environments. For the previously mentioned, the audio signature approach is used, namely, it is assumed that only there is one instance per acoustic event (for the traditional audio signature approach, only there is one version of the songs) for the types of sound classes to be considered and versions contaminated with noise of that instance (it is similar to distort each song with different types of degradations). Therefore, the aim is not to classify different instances of acoustic events into classes, but to evaluate robustness of MFCC versus MSES using a low level of SNR (Signal to Noise Ratio) in the mix of acoustic events.

The machine learning techniques utilized in this paper were selected according to the results in recognition and classification issues of related literature, besides the configuration of them are performed follow the experimental ideas in that literature and in some cases with optimization algorithms [7, 14, 21, 26].

Regarding classification, an optimization with genetic algorithm and particle swarm optimization were developed in order to improve the performance of the best combination achieved between audio features and the studied classifiers.

The built database is an additional contribution since there is no database in the reviewed works similar to the one that we propose in this paper. It has the particularity of being complex in its construction by mixing sounds at a low level of SNR. Forward, we describe in detail this database and we encourage to the readers to use it in their future works. In our case, it will be part of our tested towards exploring recognition of activity for elders living alone, for instance, to identify acoustic of events that might indicate whether the elder is using the blender or for identifying sounds of risk in a home environment.

2 Theoretical Background

The characterization of audio signals is related to the process of extracting the characteristics that abstractly describe a signal and reflect their most relevant aspects of perception. To extract the characteristics of an audio signal, it is common to segment the signal in short frames, possibly overlapping it sufficiently close to each other, in such a way that multiple events distinguishable or perceptual are not covered in a single frame [3]. This process of splitting the signal into frames is a characteristic part for computing MFCC and MSES. The next subsections describe the process for determining both audio features, as well as the different classification techniques used in the experiments that support our results.

2.1 Mel Frequency Cepstral Coefficients

MFCC are short-term spectral-based features and its success have been due to their ability to represent the amplitude spectrum in a compact form. MFCC is based on the non-linear frequency scale of human auditory perception which use two types of filters, linearly spaced filters and logarithmically spaced filters. The signal is expressed in Mel's frequency scale to capture the most important characteristics of an audio [46].

For computing MFCC, the audio signal is divided into short time frames for extracting from each one a feature vector with L coefficients. We compute the Short Time Fourier Transform for each frame, which it is given by (1), for $k = 0, 1, ..., N-1$, where k correspond to the frequency $f(k) = kf_s/N$, and f_s is the sampling frequency in Hertz. Here, $x(n)$ denotes a frame of length N and $w(n)$ is the Hann window function which it is given by $w(n) = 0.5 + 0.5cos(2\pi n/N)$:

$$X(k) = \sum_{n=0}^{N-1} x(n)w(n)e^{-i2\pi kn/N}. \qquad (1)$$

The process continues scaling the magnitude spectrum $|X(k)|$ in both frequency and magnitude. First, the frequency is scaled using the so-called

Mel filter Bank $H(k,m)$ and then the logarithm is taken using (2):

$$X^{'}(m) = \ln\left(\sum_{k=0}^{N-1}|X(k)|H(k,m)\right), \qquad (2)$$

for $m = 1, 2, ..., M$, where M is the number of filters and $M \ll N$. The Mel filter bank, $H(k,m)$, is a set of triangular filters, where the frequencies in Mel scale of the filter bank are computed with $\phi = 2595log_{10}(f/700 + 1)$, which is a common approximation. MFCC are obtained decorrelating the spectrum $X^{'}(m)$ by computing the Discrete Cosine Transform using (3):

$$c(l) = \sum_{m=1}^{M} X^{'}(m)cos\left[l\frac{\pi}{M}\left(m - \frac{1}{2}\right)\right], \qquad (3)$$

for $l = 1, 2, ..., L$, where $c(l)$ is the lth MFCC. With this procedure, a vector with L coefficients is extracted from each frame.

In this work, we will focus on the ISP implementation for computing MFCC [46], this implementation considers a filter bank $H(k,m)$ with logarithmic spacing and constant amplitude, where the number of filters is a custom parameter.

2.2 Shannon's Entropy and Spectral Entropy

When the audio signals are severely degraded, the features that describe it usually disappear, therefore, the problem becomes finding the features that would still be present in the signal despite the level of degradation to which it was subjected. Authors focused on this problem have explored entropy to characterize audio signals as robustly as possible to different types of degradations. In this address, we will start by discussing about the Shannon's entropy and spectral entropy concept.

In information theory, Shannon's entropy is related to the uncertainty of a source of information [43]. For example, entropy is used to measure the predictability of a random signal and the "peakiness" of a probability distribution function. In research, it is common to use (4) to measure, through entropy, the amount of information the signal carries. Here, p_i is the probability for any sample of the signal to have value i being n the number of possible values the samples may adopt:

$$H = -\sum_{i=1}^{n} p_i ln(p_i). \qquad (4)$$

Some estimate of the Probability Distribution Function (PDF) is needed to determine the entropy of a signal, therefore, it can be used both parametric and non-parametric methods, and histograms. If histograms are chosen, we have to be careful that the amount of data involved is high enough to avoid peaks in the histogram.

When talking about spectral entropy it is necessary to review Shen's work [44], since that concept was introduced for the first time as an additional feature for endpoint detection (voice activity detection). The idea of spectral entropy compromises to consider the spectrum of a signal as a PDF to capture the peaks of the spectrum and their location. In order to convert the spectrum into a PDF, the individual frequency components of the spectrum are separated and divided by sum of all the components, namely, $p_k = X(k)/\sum_{i=1}^{N} X(i)$, for $k = 1, 2, ..., N$, where $X(k)$ is the energy of kth frequency component of the spectrum, $\mathbf{p} = (p_1, ..., p_N)$ is the PDF of the spectrum and N is the total number of frequency components in the spectrum. This ensures the PDF area is one and can be used for computing entropy.

The concept of multiband spectral entropy was introduced by [32], and it consists of dividing the spectrum into equal-sized sub-bands to compute entropy on each one of them by using (4), where each sub-band spectrum should be assumed a PDF. Additionally, [33] proved that the multiband spectral entropy works very well with additive wide-band noise and at low levels of SNR.

2.3 Multiband Spectral Entropy Signature

Based on the idea presented by Misra et al. [32, 33], spectral entropy concept can be used for getting a robust signature that can be useful in different audio recognition issues [6, 9, 10, 11, 12, 30]. Unlike Misra et al., this work compute entropy

at each sub-band by using the entropy of a random process [9].

Let $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ be a vector of n real-valued random variables, then, $\mathbf{x}$ is said to be a Gaussian random vector where the random variables x_i are said to be jointly Gaussian if the joint probability density function of the n random variables x_i is given by $p(\mathbf{x}) = \mathcal{N}(\mathbf{m_x}, \mathbf{\Sigma_x})$, where $\mathbf{m_x} = [m_1, m_2, ..., m_n]^T$ is a vector containing the means of x_i, this is, $m_i = E[x_i]$. $\mathbf{\Sigma_x}$ is a symmetric positive definite matrix with elements σ_{ij} that are the covariances between x_i and x_j, this is, $\sigma_{ij} = E[(x_i - m_i)(x_j - m_j)]$.

Taking some precautions, the entropy of a Gaussian random vector can be determined using the continuous version of the Shannon's entropy, which is given by (5):

$$H(\mathbf{x}) = -\int_{-\infty}^{+\infty} p(\mathbf{x}) ln[p(\mathbf{x})] d\mathbf{x}. \tag{5}$$

If it is assumed that the random vector follows a Gaussian distribution with a mean of zero and the covariance matrix given by $N(0, \mathbf{\Sigma_x})$, then replacing $p(\mathbf{x})$ into (5), we get the equation for determining the entropy of a vector on a random process [34], equation (6), where $|\mathbf{\Sigma_x}|$ is the determinant of the covariance matrix:

$$H(\mathbf{x}) = \frac{n}{2} ln(2\pi e) + \frac{1}{2} ln(|\mathbf{\Sigma_x}|). \tag{6}$$

In order to compute MSES, the audio signal should be divided into frames, and for each of these to extract a vector with L coefficients of entropy. Next, the Short Time Fourier Transform is computed on each frame by using (1). For the division of the full-band spectrum into sub-bands, we take into account the idea about how people identify sounds. The human ear perceives better lower frequencies than higher ones, but not all frequencies can be heard with the same sensitivity. This process can be modeled in the whole bandwidth of the response of the ear using the Bark scale, which it is divided in 25 critical bands [49, 47]. Table 1 shows the first 24 critical bands with their respective bandwidths.

Table 1. Critical bands for the Bark scale

Critical Band	Lower cut-off (Hz)	Central Frequency (Hz)	Higher cut-off (Hz)	Bandwidth (Hz)
1	0	50	100	100
2	100	150	200	100
3	200	250	300	100
4	300	350	400	100
5	400	450	510	110
6	510	570	630	120
7	630	700	770	140
8	770	840	920	150
9	920	1000	1080	160
10	1080	1170	1270	190
11	1270	1370	1480	210
12	1480	1600	1720	240
13	1720	1850	2000	280
14	2000	2150	2320	320
15	2320	2500	2700	380
16	2700	2900	3150	450
17	3150	3400	3700	550
18	3700	4000	4400	700
19	4400	4800	5300	900
20	5300	5800	6400	1100
21	6400	7000	7700	1300
22	7700	8500	9500	1800
23	9500	10500	12000	2500
24	12000	13500	15500	3500

We use (7) to change Hertz to Barks, where f is the frequency in Hertz:

$$Barks = 13tan^{-1}\left(\frac{0.75f}{1000}\right) + 3.5tan^{-1}\left[\left(\frac{f}{7500}\right)^2\right]. \tag{7}$$

The process continues computing entropy for each one of the critical bands using (6). It was considered for each sub-band that spectral coefficients are distributed normally. This consideration is due to that a good estimate of the PDF cannot be determined by using non-parametric methods, since the lowest bands of the spectrum have too few coefficients. For computing entropy, a random process with two random variables was considered. Real and imaginary parts of the spectral coefficients are assumed to be random variables with a normal distribution and zero mean, hence, for the two-dimensional case the entropy is determined by $H = ln(2\pi e) + (1/2)ln(\sigma_{xx}\sigma_{yy} - \sigma_{xy}^2)$, where σ_{xx} and σ_{yy} are the variances of the real and imaginary parts, respectively, and σ_{xy} is the covariance between the real and imaginary parts. The result of this process is a $L \times T$ matrix (named as signature), where L is the number of coefficients of entropy and T denotes the number of frames.

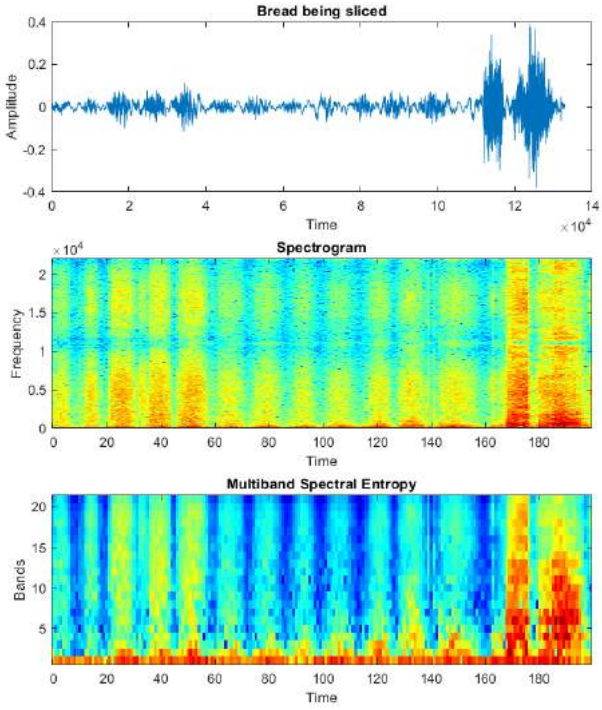


Fig. 1. Illustration of MSES signature with its corresponding signal and spectrogram. This signature corresponds to three seconds of audio from the acoustic event called "Bread being sliced"

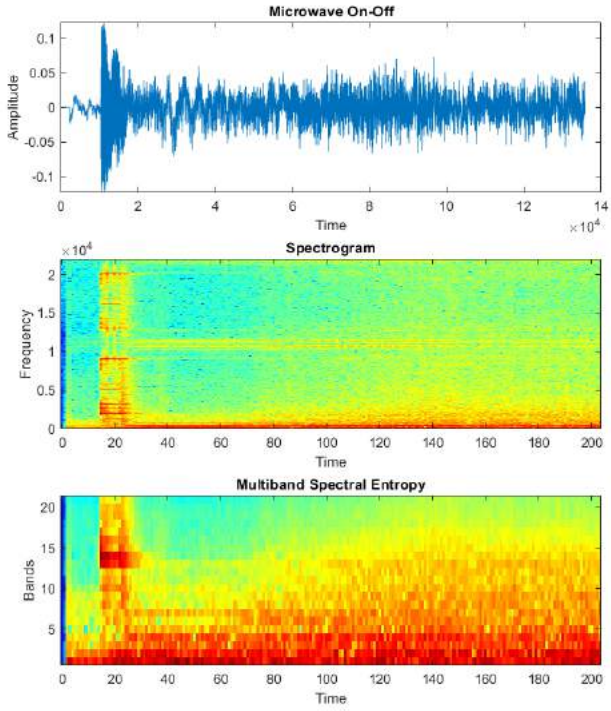


Fig. 2. Illustration of MSES signature with its corresponding signal and spectrogram. This signature corresponds to three seconds of audio from the acoustic event called "Microwave On-Off"

This signature captures the level of information content for every critical band and frame position in time.

Figures 1 and 2 show the signatures of two acoustic events that are obtained with the MSES method. The signals in time domain of the acoustic events "Bread being sliced" (Fig.1) and "Microwave On-Off" (Fig.2) are showed in the upper panels, whereas the spectrograms of both signals appears in the middle panels. The bottom of each one of the Figures displays the signatures for both acoustic events using MSES method.

2.4 Similarity Distance Functions

A measure of similarity indicates the strength of the relationship between two data points. The more the two data points resemble one another, the larger the similarity measure is. Let $\mathbf{x} = (x_1, x_2, ..., x_d)$ and $\mathbf{y} = (y_1, y_2, ..., y_d)$ be two d-dimensional data points. Then the similarity between $\mathbf{x}$ and $\mathbf{y}$ will be some function of their attribute values, as shown in (8):

$$s(\mathbf{x}, \mathbf{y}) = s(x_1, x_2, ..., x_d, y_1, y_2, ..., y_d). \quad (8)$$

A similarity distance function refers to a function $s(x, y)$ measured on any two arbitrary data points in a data set that satisfies the following properties [17]:

1. $0 \leq s(\mathbf{x}, \mathbf{y}) \leq 1$,
2. $s(\mathbf{x}, \mathbf{x}) = 1$,
3. $s(\mathbf{x}, \mathbf{y}) = s(\mathbf{y}, \mathbf{x})$.

The idea of similarity is more consistent if one considers the function of Hamming distance, since it determines the distance between two arbitrary

data points as the number of symbols or bits in which they differ. Another distance that is adopted to measure the similarity between two data points is the Cosine distance [17]. Cosine distance measures the similarity between two vectors in a space that has an internal product with which the value of the cosine of the angle between them is evaluated.

2.5 Artificial Neural Networks

The Artificial Neural Network (ANN) is a mathematical model that simulate the behavior of a biological neuron of humans. The ANN emulate the process of learning of the humans based at the equation (9). The approach for succeeding learning depends of the inputs x_i which they are multiplied with weights w_i, later, a transfer function $f(*)$ is applied for obtain the final result y for the ANN:

$$y = f\left(\sum_{i=1}^{n} x_i w_i\right). \tag{9}$$

The transfer function used in a neural network can be the sigmoidal function, linear function and hyperbolic tangent sigmoid function. For training the neural network is utilized the back-propagation method for update the weights in each epoch. The algorithm for learning can be the descendent gradient with the variants of learning rate, momentum and the use of both, also the scaled conjugate gradient and the variants of Fletcher-Reeves, Polak-Ribiére and Powell-Beale for the conjugate gradient.

2.6 Support Vector Machine

The Support Vector Machine (SVM) model is a supervised algorithm that creates a hyperplane which separates data into classes. The objective is to find an optimal plane that maximizes the distance between the separating hyperplane and the closed points (defined as support vectors) of the training data set. If the data is non-linear separable, there is a modified version of SVM which projects the original data to a high-dimensional space by the implementations of kernel functions. In the literature, there have been proposed different kernels such as linear, Gaussian and polynomial. In the case of a multi-class scenario, the SVM model assigns the label of +1 to one of them and -1 to all the remaining classes. This results in K binary SVM models, hence a model for each k class. This strategy is known the multi-class approach one versus all, and based on the principle of the "winner-takes-all".

3 Database

The kitchen is one of the home's spaces where different sound sources can occur at the same time specially when cooking. For this work we are interested in a kitchen environment where three different sound sources are occurring at the same moment. We believe that by mixing three sounds it can achieve a kitchen environment more realistic. Sounds mixing process considers as background disturbance (the noise) two of the three sounds sources, and the remain sound is the acoustic event (the signal) to be recognized. Additionally, we add an extra component to the sounds mixing process, which it consists of making the identification of the signal into the noise more perceptually difficult. The previous can be carried out using 3dB (decibels) of SNR.

In the literature, it is common to find databases containing different kinds of acoustic events, however, it is difficult to find a database with a mixture of kitchen sounds. Due to the above, our work consisted in building a database using the scheme presented in Beltrán-Márquez et al [5]. Sixteen archives of audio were collected where each one is a class of kitchen sound. The portals where these sounds were downloaded are, `www.soundsnap.com`, `www.freesfx.co.uk` and `www.sounddogs.com`. The audio files are WAV format, with a 44100Hz of sampling frequency and coded to 16 bits. No copyright infringement was intended. The downloaded sounds are presented in Table 2.

The audio signatures approach suggests the use of signatures between one to fifteen seconds. All downloaded audio files have a length of three seconds (we consider that three seconds of audio is enough to identify a sound from the environment). As indicated above, the database

consists of sixteen original sounds for mixing. First, mixing process consists of forming a dataset with the mixture of all the combinations of pairs of sounds. Second, all the elements from the dataset are combined with each one of the sixteen original sounds for getting mixtures with three sounds. Repetitions of sounds in a single mixture are avoided. All mixtures are obtained using 3dB of SNR, for this, the sixteen original sounds are considered the "signal" (the acoustic events to identify) and the elements of the dataset as the "noise". Figure 3 shows a illustration of the mixing process of sounds. The equation $SNR = 10 \times log_{10}(P_{signal}/P_{noise})$ is used to determine SNR between signal and noise, where P_{signal} is the power of the signal and P_{noise} is the power of the noise. Finally, the database has 1680 audio files, all of them grouped into 16 classes, where each class has 105 audio files.

In the experiments, we used classifiers such as Similarity Distance, k- Nearest Neighbors (KNN), SVM and ANN. For the experiments with KNN, ANN and SVM, we generated a training dataset to train the models of classification (this is because the elements of the database will be used as test elements to assess the classification models). This training dataset is built by using the original signal of each one of the sixteen kitchen sounds and two degraded versions of each one of them (this procedure guarantees having more data for training since there are not more instances for each class of sound). Degradation consists of distorting the signal by adding white Gaussian noise. We use $awgn(signal, SNR)$ MATLAB® function for this matter, where $signal$ is the original kitchen sound and SNR take the value of 35dB and 50dB respectively for each degraded version. The total number of audios in the training dataset is of 48. Original and mixtures of audios are available in `https://drive.google.com/open?id=1ALkT-nVt3HMFK66CjcWrC3dHrNhiyZuk`

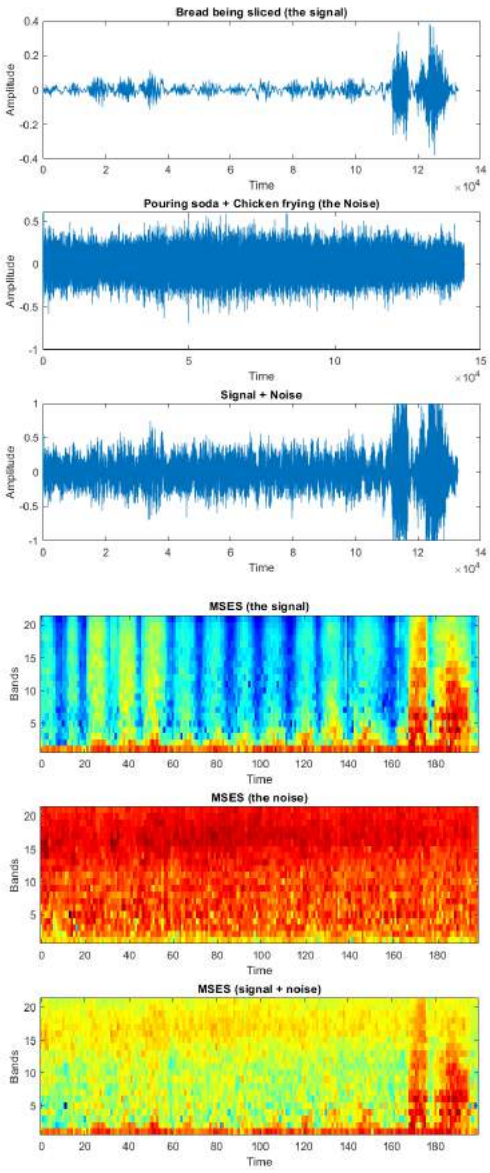


Fig. 3. Illustration of the mixing process of sounds considering the acoustic event named "Bread being sliced" as the signal and the couple of sounds "Pouring Soda - Chicken Frying" as the noise. First three panels show the signals in time domain and the last panels show the MSES signatures associated to every signal

4 Experiments

In this work, we use measures of similarity as baseline experiment to have a starting point or a first measurement in relation to the performance indicators of the considered classifiers. Certainly, the search by similarity identifies which candidate identities are more similar to one or more input entities for coincidence. In the next section,

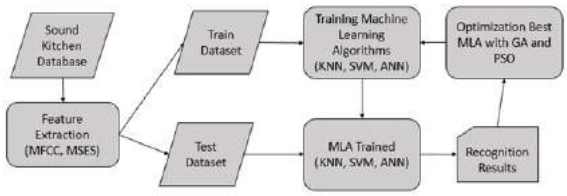


Fig. 4. Flow chart of the activities of the proposed approach

we describe how compute this entities from an approach of audio signatures.

On the other hand, as previously stated, the ANN and SVM algorithms have already been used in acoustic event classification tasks [7, 14, 24, 29, 51, 36]. Therefore, we consider it appropriate to include these models in our experiments by using the Bayesian optimization strategy for SVM and different architectures for ANN to identify their best parameters. Also, as a baseline model, the KNN algorithm was included in our study because of its easy implementation, and for this particular case, to test different distance metrics and number of neighbors.

To understand the process to be followed in our experiments, Figure 4 shows the block diagram of the sequential activities carried out in this section.

4.1 MFCC and MSES Signatures

To extract both MFCC and MSES signatures, the next procedure was implemented. a) First, stereo signals are changed to monoaural by averaging both channels, and each audio is cut to three seconds of length. b) Frames of 30ms are used to divide the monoaural signal (i.e. we use 1323 samples per frame using a sampling frequency of 44100Hz). c) Consecutive frames have an overlap of 50%, hence, there are 200 frames (T = 200) for three seconds of audio. d) A Hann window function is applied to each frame. e) The FFT is computed for each frame.

With the FFTs we are ready to compute Mel Frequency Cepstral Coeffcients (section 2.1), and the Multiband Spectral Entropy Signature (section 2.3). An additional point is that MSES signatures are extracted considering a bandwidth of 0Hz up to 8000Hz, hence, only 21 critical bands are used. The above entails each feature vector be 21-dimensional (L = 21). To have similar conditions between MSES and MFCC features, we compute MFCC using 21 triangular band-pass filters within the bandwidth mentioned before. Besides, MFCC vectors are also 21-dimensional.

4.2 Baseline Experiment with Similarity Distances

Baseline experiment consists of using similarity distances for recognition of acoustic events from the database of the kitchen sounds. Baseline experiment considers two different signatures, one uses normalized values and the other binary values. To normalize the signatures, we normalized the $L \times T$ matrix by computing the mean and standard deviation of all data of the matrix.

Haitsma's work presents a method to binarize audio signatures. This method consists of taking the sign of the differences between consecutive values [22]. For the baseline experiment the sign of the differences is encoded using $s_{ij} = 1$, if $v_{ij} - v_{ij-1} \geq 0$ and $s_{ij} = 0$ by other way, where s_{ij} denotes the i, jth binary value, v_{ij} denotes the ith value referred to the frame j, and v_{ij-1} denotes the ith value referred to the frame $j-1$ of the signature, for $i = 1, 2, ..., L$ and $j = 1, 2, ..., T$.

4.3 Experiment with Artificial Neural Networks

This experiment consists of training neural networks to classify the acoustic events that are considered the signal (not the noise) in the audios of the database. Two neural networks were considered, one trained with MFCC signatures and the other trained with MSES signatures. To train the neural networks, we used the normalized signatures that are extracted from each audio of the training dataset. Therefore, we have 48 signatures for training the neural network for MFCC and other 48 signatures for training the neural network for MSES.

For both MFCC and MSES, the neural networks consist of 2 hidden layers and 16 neurons in the output layer; the input layer has 4200 neurons (i.e., each signature of size 21×200 is converted to vector). The design of each ANN was proposed

according to the works cited in [18, 31] for the selection of the different elements of the learning algorithms. We tested three designs of neural networks with the following architectures: In the first design, the descendent gradient with adaptive learning rate back-propagation is implemented with 95 neurons in the first hidden layer and 28 neurons in the second hidden layer. For the second design, the descendent gradient with momentum and adaptive learning rate back-propagation is utilized with 150 neurons in the first hidden layer and 35 neurons in the second hidden layer.

In the third design, the scaled conjugate gradient back-propagation is applied with 79 neurons in the first hidden layer and 22 neurons in the second hidden layer. To set the number of neurons, a search was made for the best performance of the neural network in the learning stage by increasing one neuron from 10 up to 200 in the hidden layers. Finally, for each ANN, the first and second hidden layers, the hyperbolic tangent sigmoid transfer function is applied and for the output layer, the logarithmic-sigmoid transfer function is implemented.

The classification process consisted of assessing the neural networks using the normalized signature of the mixture of kitchen sounds of the database. If a neural network correctly classifies a given acoustic event in the entire database, then there will be 105 true positives for that class. The performance goal and numbers of epochs for all the neural networks are 1e-06 and 8000, respectively.

4.4 Experiment with Support Vector Machines

The same training dataset used for the ANN is used for the experiments with SVM. In our implementation, the $fitcsvm()$ MATLAB® built-in function has been used to train the SVM classifiers. There were trained 16 binary SVM models, one for each kitchen sound class. Gaussian, linear and polynomial kernels were compared in order to select the most appropriate for each model. The Bayesian optimization strategy was implemented in order to select optimal hyper-parameters by the evaluation of 30 models for each binary classifier. The best results were achieved with Gaussian kernels and the Sequential Minimal Optimization solver. Once the parameters of the 16 SVM models were defined, each mixture of sounds is classified with the model that achieved the highest score.

4.5 Experiment with K-Nearest Neighbors

Similar than the models based on SVM, the optimizer hyper-parameter function of MATLAB®, $fitcknn()$, was implemented to perform a Bayesian optimization strategy. In this implementation, different distance metrics, such as Euclidean, City-block, Cosine, Minkowski, Correlation, Spearman, Hamming, Mahalanobis, Jaccard, and Chebychev, were evaluated. Also, different number of neighbors were implemented within each search. In total, there were compared the performance of 30 different models.

5 Results and Discussion

In this section, we compare results about the performance of MSES and MFCC using four types of classifiers: similarity distances, KNN, ANN and SVM. Results are showed using True Positives (TP) and False Positives (FP) from the confusion matrices, the best experimental outcomes and the averages achieved with each classifier are summarized in Table 7.

5.1 Similarity Distance Results

Table 2 shows the results for each signature using Hamming distance and Cosine distance, here the recall metric is used for results comparison. Although it is common to use binary signatures in an audio signature-based approach, the results of Table 2 suggests that binary signatures are not convenient to represent acoustic events, especially, when they have non-stationary characteristics.

The difference in recall between both features is about the 3.46%, therefore, no advantage can be seen by using MFCC or MSES features. An audio signature using normalized values seens to work better, allowing to differentiate more the performance of both feature extraction methods, especially, when working with low levels of SNR.

Table 2. Results about Recall

Sounds[a]	Hamming Distance		Cosine Distance	
	MFCC	MSES	MFCC	MSES
C1	43.80	**51.42**	49.52	**95.23**
C2	**0**	**0**	0.95	**6.66**
C3	**100**	**100**	90.47	**94.28**
C4	**100**	**100**	84.76	**100**
C5	99.04	**100**	80	**80.95**
C6	**100**	98.09	**100**	51.42
C7	**41.90**	40	44.76	**97.14**
C8	**65.71**	62.85	42.85	**100**
C9	27.61	**40**	36.19	**45.71**
C10	57.14	**79.04**	69.52	**78.09**
C11	14.28	**38.09**	**28.57**	17.14
C12	**43.80**	35.23	61.90	**87.61**
C13	**100**	**100**	94.28	**96.19**
C14	53.33	**70.47**	63.80	**97.14**
C15	**98.09**	84.76	85.71	**100**
C16	**100**	**100**	81.90	**100**
Average	65.29	**68.75**	63.45	**77.97**

[a] The different acoustic events are: (C1) Bread being sliced, (C2) Chop food quickly and strongly, (C3) Pouring soda into a glass, (C4) Electric blender liquefying food, (C5) Frying chicken in a pan, (C6) Hot oil in a pan, (C7) Burner of a stove, (C8) Making popcorn in a microwave, (C9) Cooking fryer, (C10) Peeling potatoes, (C11) Making popcorn in a pot, (C12) Turning a microwave on and off, (C13) Pouring water into a glass, (C14) Slicing onions, (C15) Boiling teapot, and (C16) Boiling eggs.

Hamming distance results, Table 2, shows that C2 was the worst classified class because it has zero in recall score, while, C3, C4, C13 and C16 are the classes of sounds with the higher recall in both features, 100% in all of them. The average recall for MFCC features is 65.29% and 68.75% for using MSES features.

The results with Cosine distance using MSES feature, shows that C4, C8, C15 and C16 are the classes of sounds getting the higher recall score, whereas C2 was the worst classified class for both features.

In this case, the average recall obtained for MFCC features is 63.45% and 77.97% for MSES features (i.e., the difference of recall between both features is about the 14.52%).

The results of this experiment mark the baseline and the starting point to evaluate the contribution of machine learning methods. An image-based representation of the results with similarity distances, KNN, SVM and ANN methods for both MFCC and MSES methods is showed in Figure 5 using confusion matrices.

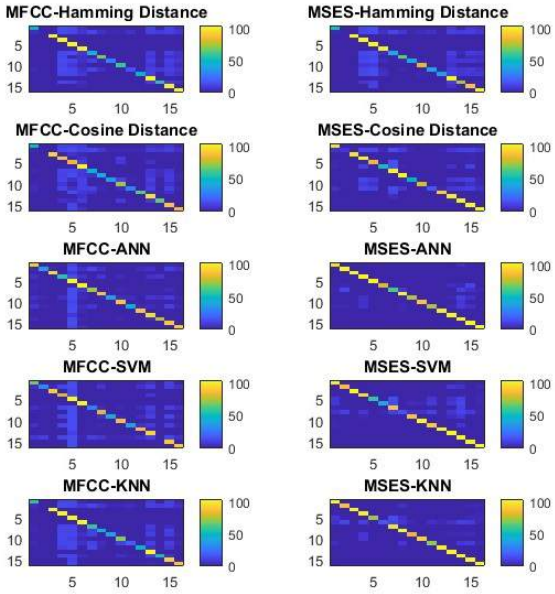


Fig. 5. Confusion matrices obtained from similarity distances, KNN, SVM and ANN methods.

5.2 Artificial Neural Network Results

Table 3 shows the results obtained with the neural network architectures using back-propagation with gradient descendent and adaptive learning rate (NNGDA), gradient descendent with momentum and adaptive learning rate (NNGDX) and scaled conjugate gradient (NNSCG). The best recall achieved for MFCC features is of 75% and for MSES features is 90.95%, both with NNGDX. The average is obtained for 30 experiments, but only 10 experiments are presented in Table 3. The best average recall score was 73.42% and 88% for MFCC and MSES respectively, both from NNGDX method.

Table 4 shows the results about True Positives (TP) and False Positives (FP) from the confusion matrix obtained for the best performance with artificial neural networks using MFCC and MSES, respectively. For MFCC, C5 and C6 are the classes that obtained the higher scores, 1 and 2 errors

Table 3. Results for artificial neural networks using recall metric

Experiment	NNGDA		NNGDX		NNSCG	
	MFCC	MSES	MFCC	MSES	MFCC	MSES
1	73.21	88.1	**75**	**90.95**	73.69	89.05
2	73.15	87.92	**74.88**	**90.24**	73.51	88.99
3	73.04	87.8	**74.52**	86.76	73.27	**88.33**
4	73.04	87.8	**74.17**	**89.23**	73.15	88.21
5	72.98	87.8	**74.11**	**89.17**	73.15	88.21
6	72.92	87.68	**73.87**	**89.17**	73.1	88.1
7	72.92	87.5	**73.81**	**89.11**	72.98	87.74
8	72.86	87.5	**73.81**	**89.05**	72.92	87.74
9	72.8	87.5	**73.75**	**88.75**	72.92	87.62
10	72.8	87.44	**73.75**	**88.69**	72.86	87.62
Best Result	73.21	88.1	**75.00**	**90.95**	73.69	89.05
Average	72.62	86.94	**73.42**	**88.00**	72.59	87.23

Table 4. Results about True Positives (TP) and False Positives (FP) with artificial neural networks

Sounds	MFCC		MSES	
	TP	FP	TP	FP
C1	91	32	98	5
C2	37	5	**105**	18
C3	97	3	104	0
C4	56	5	**105**	17
C5	104	176	**105**	1
C6	103	41	81	1
C7	72	19	63	0
C8	88	18	104	14
C9	38	0	77	1
C10	89	15	90	0
C11	46	4	75	1
C12	89	9	101	0
C13	96	26	**105**	9
C14	75	16	**105**	48
C15	90	37	**105**	31
C16	89	14	**105**	6

respectively. At least 14 samples of each class of kitchen sounds (except by C6) are classified erroneously as C5.

For MSES, C7 is the class with more errors, 42 in total, followed by C11 (30 errors) and C9 (28 errors). Unlike the experiment with similarity distances, here the sound class C2 is 100% classified. The others classes with higher scores are C3, C4, C5, C8, C13, C14, C15 and C16. Indeed, experiments with ANN show that there is an increase in the recall with which kitchen sounds are identified. Comparing the average value achieved with distances of similarity and neural networks, there is an increase of recall of 8.13% for MFCC and 10.03% for MSES.

5.3 Support Vector Machine Results

Table 5 shows the results for TP and FP from the confusion matrix obtained with the SVM classifier using MFCC and MSES features, respectively. The recall obtained by using the MFCC features is 67.2%. C5 and C6 are the classes that obtained the higher recall, zero and one errors respectively. For MFCC, at least 14 samples of each class of kitchen sounds (except by C5 and C6) are classified erroneously as C5. All the sound samples of C14 are miss classified (105 errors). C7 and C2 obtained 77 and 73 errors, respectively. For MSES, the recall achieved is 83.99%. C8 is the class with more errors, 89 in total, followed by C6 (61 errors) and C5 (46 errors). Comparing the average value achieved with distances of similarity and SVM, there is an increase of recall of 1.91% for MFCC and 6.02% for MSES.

5.4 K-Nearest Neighbors Results

As previously mentioned, the $fitcknn()$ MATLAB® function was used to compare the performance of 30 different models. The one that obtained the best performance with MFCC features was the model that uses the Spearman distance function with two neighbors. Table 6 shows the results about TP and FP from the confusion matrix of this implementation. The recall metric was 65.77%. C3, C4, C5, C6 and C13 obtained the best results. Contrary, only one sample of C2 was correctly classified. The results obtained with MSES feature (Table 6) showed that the best KNN model uses the correlation distance function and one neighbor. The recall metric obtained was 87.38%. C8, C13, C14 and C16 obtained zero errors in classification. Four classes obtained between 1, 2 or 3 errors. The more difficult class to identify was C6 with 88 errors in total.

Comparing the average value achieved with distances of similarity and KNN, there is an increase of recall of 0.48% for MFCC and 9.41% for MSES.

Table 7 shows the summary of the obtained results for both MFCC and MSES features and all classifiers: Similarity distances, ANN, SVM and KNN. We can observe first that all the classifiers

Table 5. Results about True Positives (TP) and False Positives (FP) with support vector machine

Sounds	MFCC		MSES	
	TP	FP	TP	FP
C1	68	24	103	26
C2	32	13	88	12
C3	93	3	104	0
C4	81	63	97	10
C5	**105**	191	59	0
C6	104	3	44	2
C7	28	10	90	40
C8	83	29	16	0
C9	43	4	89	9
C10	88	9	**105**	9
C11	40	3	93	27
C12	88	31	**105**	7
C13	94	72	**105**	31
C14	0	0	**105**	71
C15	91	72	103	9
C16	91	24	**105**	16

Table 6. Results considering True Positives (TP) and False Positives (FP) for k-nearest neighbors

Sounds	MFCC		MSES	
	TP	FP	TP	FP
C1	58	11	104	18
C2	1	0	90	12
C3	104	1	103	0
C4	**105**	113	103	25
C5	104	134	73	1
C6	**105**	37	17	0
C7	60	57	100	29
C8	45	17	**105**	34
C9	39	3	85	3
C10	72	21	95	0
C11	19	6	72	5
C12	50	2	102	4
C13	**105**	77	**105**	11
C14	54	9	**105**	22
C15	94	70	104	29
C16	90	17	**105**	19

have an improvement in the recall metric when working with MSES feature.

Second, the ANN classifier has the highest performance for both MFCC and MSES (73.42% and 88%, respectively), followed by a combination MSES-KNN (87.38%), then a combination MSES-SVM (83.99%), and finally, similarity distances-MSES with a score of 77.97%. Regarding MFCC, the second best performance was achieved with SVM (i.e., 67.2%).

Table 7. Best results for the classification of kitchen sounds

Method	Feature	
	MFCC (%)	MSES (%)
Similarity Distance	65.29	77.97
ANN	**73.42**	**88.00**
SVM	67.20	83.99
KNN	65.77	87.38

Third and fourth best performance were achieved with KNN and similarity distances (65.77% and 65.29%, respectively). We attribute the good performance of ANN to the fact this machine learning technique works with variations that allow their learning to be more robust and effective than the other methods.

5.5 Test of Statistical Significance

To further analyze the differences between MFCC and MSES methods, we applied a non-parametric Mann-Whitney's test with a significance level of $\alpha = 0.05$. For this test, two population samples were related which belong to the recall metric scores of the 30 models evaluated using ANN, SVM and KNN for both MFCC and MSES features. The results show a value $p = 0.0003$, which makes us reject the null hypothesis and conclude that the medians of both methods are different and that they do not depend on the type of classifier or the sounds to be recognized.

5.6 Optimization of ANN with GA and PSO

Previous results showed that the combination MSES-ANN (audio features-classifier) achieved the best score for all the combinations. In this part, we realized an optimization looking for the best artificial neural network with MSES. This optimization is performed using the genetic algorithm (GA) and particle swarm optimization (PSO). The use of the GA and PSO optimization algorithms are decided in consideration because

these algorithms performed good results in optimization of parameters for machine learning algorithms [19, 20].

The optimization looks up for the following ANN's values and parameters:

1. Number of neurons in the first hidden layer.
2. Number of neurons in the second hidden layer.
3. The transfer functions for the neurons in the first and second hidden layer, and for the neurons in the output layer. The transfer functions for optimizing are the next:
   - Positive linear transfer function.
   - Linear transfer function.
   - Inverse transfer function.
   - Log-sigmoid transfer function.
   - Hyperbolic tangent sigmoid transfer function.
   - Triangular basis transfer function.
   - Hard-limit transfer function.
   - Saturating linear transfer function.
   - Elliot symmetric sigmoid transfer function.
   - Symmetric saturating linear transfer function.
   - Symmetric hard-limit transfer function.
   - Elliot 2 symmetric sigmoid transfer function.
4. The learning algorithms implemented in the neural network:
   - Levenberg-Marquardt backpropagation.
   - One-step secant backpropagation.
   - Gradient descent with adaptive learning rate backpropagation.
   - Gradient descent with momentum and adaptive learning rate backpropagation.
   - Scaled conjugate gradient backpropagation.
   - Resilient backpropagation.
   - Gradient descent backpropagation.
   - Gradient descent with momentum backpropagation.
   - Conjugate gradient backpropagation with Fletcher-Reeves updates.
   - Conjugate gradient backpropagation with Polak-Ribiére updates.
   - Conjugate gradient backpropagation with Powell-Beale restarts.

Table 8. Parameters for GA

Population	100 Individuals
Individual	6 Genes (real)
Generations	100
Assign Fitness	Ranking
Selection	Stochastic universal sampling
Mutation	16.67 %
Crossover	Single Point (80%)

Table 9. Parameters for PSO

Population	100 Particles
Particle	6 Dimensions (real)
Iterations	100
Constriction Coefficient	1
Inertia Weight	0.1
R1, R2	Random in the range [0,1]
C1	Lineal decrement (2–0.5)
C2	Lineal increment (0.5–2)

In Table 8, the parameters for the performance of GA are showed and Table 9 shows the parameters for the performance of PSO.

Table 10 shows the results for acoustic event recognition for 10 experiments that combine MSES-ANN with both optimization techniques GA and PSO. The average in recall metric was 91.46% and 91.55% for GA and PSO, respectively,

The best recall for the optimization of the neural network was obtained with PSO achieving a 93.93 % of recognition for the kitchen sounds. The parameters of the best ANN architecture with PSO are:

- 1st Hidden layer (1HL) with 186 neurons.
- 2nd Hidden layer (2HL) with 238 neurons.
- The transfer function in 1HL was saturating linear transfer function.

Table 10. Results about optimization of ANN-MSES with GA and PSO

Experiment	Algorithm	
	GA	PSO
1	92.20	90.89
2	91.31	89.35
3	91.67	91.85
4	91.01	93.21
5	91.67	91.90
6	91.31	91.37
7	91.19	91.13
8	91.13	90.71
9	91.19	91.13
10	91.90	93.93
Best result	**92.20**	**93.93**
Average	**91.46**	**91.55**

Table 11. Results with the best ANN architecture

Experiment	**Recall**
1	94.52
2	93.51
3	94.11
4	94.70
5	93.21
6	93.39
7	93.15
8	93.81
9	93.04
10	93.10
Best result	**94.70**
Average	**93.46**

— The transfer function in 2HL was symmetric saturating linear transfer function.

— The transfer function in output Layer was symmetric saturating linear transfer function.

— The training learning algorithm was conjugate gradient backpropagation with Fletcher-Reeves updates.

Finally, 30 experiments were realized using ANN with the above configuration parameters with the aim of testing the optimization robustness. Table 11, presents only the best 10 results where one can observe that the average recall achieved for the optimized combination MSES-ANN was 93.46%, that is, 15.49% of improvement when compared with the average value achieved with distances of similarity and optimized ANN-MSES.

Table 12 shows the results about True Positives (TP) and False Positives (FP) from the confusion matrix obtained for the best performance with the couple MSES-ANN and optimized with PSO. The results of the table showed that, excepting the C7 sound, all classes have a success ratio between 90% and 100% for the recognition of acoustic events that define each class. The optimization of the neural network helps to improve the recognition rate and to reduce the number of miss classified sounds (False Positives). An image-based representation of the confusion matrix of this experiment is showed in Figure 6. Notice that the color of the diagonal indicates that there is a high recognition rate for each of the classes.

6 Conclusions

In this work, we identify acoustic events using the approach of audio signatures in combination with machine learning algorithms. When different instances of a sound class are not available, the audio signatures approach should be used since this approach only requires the original sound and degraded versions of it.

Audio signatures help us to cope with the small database of the kitchen sound sources, which in our case consisted of sixteen original sounds and some degraded versions of these. In order to complement the audio signatures approach, we studied the performance of machine learning algorithms when there is only an instance of the sound classes and degraded versions of them.

The two audio features considered in this work are MFCC and MSES. MFCC is one most cited audio feature when working with audio-based activity recognition, and the reason to be considered as our benchmark feature. MSES is an interesting audio feature being widely adopted because of its robustness to noise.

The results showed that the representation of acoustic events based on MSES is more convenient when working with different classification methods. Although the comparison between MSES and MFCC is not conclusive, it seems that

Table 12. Results about True Positives (TP) and False Positives (FP) with optimized ANN

Sounds	MSES	
	TP	FP
C1	96	1
C2	99	6
C3	104	0
C4	103	14
C5	104	5
C6	100	1
C7	63	0
C8	104	3
C9	102	4
C10	95	0
C11	102	6
C12	103	2
C13	104	3
C14	104	16
C15	103	3
C16	**105**	26

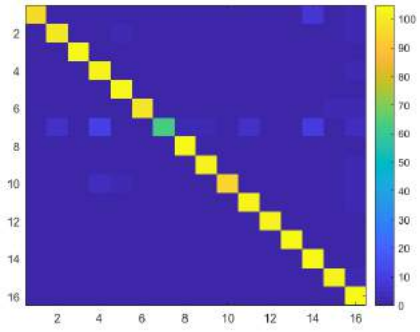


Fig. 6. Confusion matrix obtained with the couple ANN-MSES and optimized with PSO

MSES is an audio feature that is very robust for identifying acoustic events in a mixture of sounds.

One thing to note is that MSES captures the location of energy peaks in each sub-band that are less corrupted by noise, even in the presence of low SNR levels, something that affect the performance of MFCC. Nevertheless, both MFCC and MSES represent very well the non-stationary characteristics of audio signals.

A database with a mixture of everyday kitchen sounds was created using 3dB of SNR. The way in which this database is constructed should encourage readers to use it in future works since this database considers noisy contexts, something that to our understanding is not available in the literature. Yet there are databases with sound sources from different and independent tasks but never mixed, such the one provided by the DCASE2020 database.

The results presented here showed a way for identifying acoustic events when they are immersed in a mixture of sounds and they are not predominant, which is important for recognizing activities in real indoor environments. In the classification stage, four types of classifiers were used, Similarity Distances, k-Nearest Neighbors, Support Vector Machines and Artificial Neural Networks.

The results of Table 7 showed that MSES combined with Artificial Neural Networks has an score of 88% in recall metric which outperforms any other combination of classifiers with MSES or MFCC. In addition, a test of statistical significance was realized, getting a value of p = 0.0003, which makes us reject the null hypothesis and conclude that MFCC and MSES features have different level of robustness and that their performance do not depend on the type of classifier nor on the sound to be recognized.

Furthermore, the use of a genetic algorithm and a particle swam optimization improved the performance of audio features recognition supported by machine learning classifiers, being the combination MSES-ANN the one the best performance (93.46%). Table 10 showed that PSO performs better than GA achieving a average recall of 91.55%.

Finally, the experiments presented in this work focused on the evaluation MSES and MFCC audio features techniques that are supported by machine learning algorithms for the recognition of acoustic events on noisy environments. We considered the context of a kitchen context where different sound sources are present, for instance, when a person is preparing meals. In an attempt to make a more realistic scenario sound sources were mixed and applied a low SNR level. This is an acoustic recognition approach that would help better understand the nature of human activity in the home setting.

The identification of all the sounds that are present in the environment might help to develop systems that can assist people or that can be aware of potential dangers.

References

1. **Almaadeed, N., Asim, M., Al-Maadeed, S., Bouridane, A., Beghdadi, A. (2018).** Automatic detection and classification of audio events for road surveillance applications. Sensors, Vol. 18(6), pp. 1–19.
2. **Alsina-Pagès, R. M., Navarro, J., Alías, F., Hervás, M. (2017).** homesound: Real-time audio event detection based on high performance computing for behaviour and surveillance remote monitoring. Sensors, Vol. 17(4), pp. 1–22.
3. **Aucouturier, J.-J., Defreville, B., Pachet, F. (2007).** The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. The Journal of the Acoustical Society of America, Vol. 122(2), pp. 881–891.
4. **Bansal, R., Shukla, N., Goyal, M., Kumar, D. (2020).** Information and Communication Technology for Intelligent Systems, chapter Enhancement and Comparative Analysis of Environmental Sound Classification Using MFCC and Empirical Mode Decomposition. Springer, Singapore, pp. 227–235.
5. **Beltrán, J., Chávez, E., Favela, J. (2012).** Environmental sound recognition by measuring significant changes in the spectral entropy. Lecture Notes in Computer Science, Mexican Conference on Pattern Recognition, Vol. 1, pp. 334–343.
6. **Beltrán, J., Chávez, E., Favela, J. (2015).** Scalable identification of mixed environmental sounds, recorded from heterogeneous sources. Pattern Recognition Letters, Vol. 68(1), pp. 153–160.
7. **Bountourakis, V., Vrysis, L., Konstantoudakis, K., Vryzas, N. (2019).** An enhanced temporal feature integration method for environmental sound recognition. Acoustics, Vol. 1(2), pp. 410–422.
8. **Bryan-Kinns, N. (2017).** Interaction design with audio: Speculating on sound in future design education. The 4th Central China International Design Science Seminar 2017, pp. 1–9.
9. **Camarena-Ibarrola, A., Chávez, E. (2006).** On musical performances identification, entropy and string matching. 2006 Mexican International Conference on Artificial Intelligence, pp. 952–962.
10. **Camarena-Ibarrola, A., Chávez, E. (2010).** Real time tracking of musical performances. 2010 Mexican International Conference on Artificial Intelligence, pp. 138–148.
11. **Camarena-Ibarrola, A., Figueroa, K., García, J. (2020).** Speaker identification using entropygrams and convolutional neural networks. 2020 Mexican International Conference on Artificial Intelligence, pp. 23–34.
12. **Camarena-Ibarrola, A., Luque, F., Chávez, E. (2017).** Speaker identification through spectral entropy analysis. 2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), pp. 1–6.
13. **Chachada, S., Kuo, J. (2014).** Environmental sound recognition: A survey. APSIPA Transactions on Signal and Information Processing, Vol. 3, pp. 1–15.
14. **Chandrakala, S., Jayalakshmi, S. L. (2019).** Environmental audio scene and sound event recognition for autonomous surveillance: A survey and comparative studies. ACM Computing Surveys, Vol. 52(3), pp. 1–34.
15. **Cheng, C.-F., Rashidi, A., Davenport, M. A., Andersona, D. V. (2017).** Activity analysis of construction equipment using audio signals and support vector machines. Automation in Construction, Vol. 81, pp. 240–253.
16. **Deepsheka, G., Kheerthana, R., Mourina, M., Bharathi, B. (2020).** Recurrent neural network based music recognition using audio fingerprinting. 2020 Third International Conference on Smart Systems and Inventive Technology, pp. 1–6.
17. **Gan, G., Ma, C., Wu, J. (2007).** In Data Clustering: Theory, Algorithms and Applications, chapter Similarity and Dissimilarity Measures. ASA-SIAM Series on Statistics and Applied Probability, pp. 67–106.
18. **Gaxiola, F., Melin, P., Valdez, F., Castillo, O. (2011).** Modular neural networks with type-2 fuzzy integration for pattern recognition of iris biometric measure. Batyrshin I., Sidorov G. (eds) Advances in Soft Computing. MICAI 2011. Lecture Notes in Computer Science, Vol. 7095.

19. **Gaxiola, F., Melin, P., Valdez, F., Castro, J. (2018).** Optimization of deep neural network for recognition with human iris biometric measure. Melin P., Castillo O., Kacprzyk J., Reformat M., Melek W. (eds) Fuzzy Logic in Intelligent System Design. NAFIPS 2017. Advances in Intelligent Systems and Computing, Vol. 648.

20. **Gaxiola, F., Melin, P., Valdez, F., Castro, J., Manzo-Martínez, A. (2019).** Pso with dynamic adaptation of parameters for optimization in neural networks with interval type-2 fuzzy numbers weights. Axioms, Vol. 8(1).

21. **Grama, L., Rusu, C. (2019).** Extending assisted audio capabilities of tiago service robot. 2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), IEEE Xplore, pp. 1–8.

22. **Haitsma, J., Kalker, T. (2002).** A highly robust audio fingerprinting system. Proceedings of International Symposium on Music Information Retrieval, pp. 1–9.

23. **Huang, W., Zhang, Y. (2020).** Application of hidden markov chain and artificial neural networks in music recognition and classification. Proceedings of 2020 the 6th International Conference on Computing and Data Engineering, pp. 49–53.

24. **Jatturas, C., Chokkoedsakul, S., Na-Ayudhya, P. D., Pankaew, S., Sopavanit, C., Asdornwised, W. (2019).** Recurrent neural networks for environmental sound recognition using scikit-learn and tensorflow. 2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, pp. 1–6.

25. **Kar, B., Samanta, S., Prasad-Manna, R., Chatterjee, S. (2019).** An optimized music recognition system using mel-frequency cepstral coefficient (mfcc) and vector quantization (vq). Special Issue International Business Research Conference on Transformation Opportunities and Sustainability Challenges in Technology and Management., Vol. 45489(1208), pp. 100–106.

26. **Kaur, G., Srivastava, M., Kumar, A. (2018).** Genetic algorithm for combined speaker and speech recognition using deep neural networks. Journal of Telecommunications and Information Technology, Vol. 2, pp. 23–31.

27. **Kumar, A. P., Roy, R., Rawat, S., Sudhakaran, P. (2017).** Continuous telugu speech recognition through combined feature extraction by mfcc and dwpd using hmm based dnn techniques. International Journal of Pure and Applied Mathematics, Vol. 114(11), pp. 187–197.

28. **Kumar, A. S., Erler, R., Kowerko, D. (2019).** Audio-based event recognition system for smart homes. Proceedings of the 27th ACM International Conference on Multimedia, ACM, pp. 2205–2207.

29. **Li, J., Dai, W., Metze, F., Qu, S., Das, S. (2017).** A comparison of deep learning methods for environmental sound detection. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 126–130.

30. **Luque-Suarez, F., Camarena-Ibarrola, A., Chávez, E. (2019).** Efficient speaker identification using spectral entropy. Multimedia Tools and Applications, Vol. 78, pp. 16803–16815.

31. **Melin, P. (2012).** Modular neural networks for person recognition using the contour segmentation of the human iris. Modular Neural Networks and Type-2 Fuzzy Systems for Pattern Recognition. Studies in Computational Intelligence, Vol. 389.

32. **Misra, H., Ikbal, S., Bourlard, H., Hermansky, H. (2004).** Spectral entropy based feature for robust asr. 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 1–8.

33. **Misra, H., Ikbal, S., Sivadas, S., Bourlard, H. (2005).** Multi-resolution spectral entropy feature for robust asr. 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 1–9.

34. **Mohammad-Djafari, A. (2001).** Entropie en traitement du signal. Laboratoire des Signaux et Systemes, pp. 1–9.

35. **Moreaux, M., Garcia-Ortiz, M., Ferrané, I., Lerasle, F. (2019).** Benchmark for kitchen20, a daily life dataset for audio-based human action recognition. 2019 International Conference on Content-Based Multimedia Indexing, pp. 1–6.

36. **Mushtaq, Z., Su, S.-F. (2020).** Environmental sound classification using a regularized deep convolutional neural network with data augmentation. Applied Acoustics, Vol. 167, pp. 1–13.

37. **Naithani, K., Thakkar, V. M., Semwal, A. (2018).** English language speech recognition using mfcc and hmm. 2018 International Conference on Research in Intelligent and Computing in Engineering (RICE), IEEE Xplore, pp. 1–7.

38. Naronglerdrit, P., Mporas, I. (2017). Interactive Collaborative Robotics, chapter Recognition of Indoors Activity Sounds for Robot-Based Home Monitoring in Assisted Living Environments. Springer, Cham, pp. 153–161.

39. Naronglerdrit, P., Mporas, I., Sotudeh, R. (2017). Improved automatic keyword extraction given more linguistic knowledge. 2017 IEEE 13th International Colloquium on Signal Processing and its Applications (CSPA), IEEE Xplore, pp. 23–28.

40. Pires, I. M., Santos, R., Pombo, N., Garcia, N. M., Flórez-Revuelta, F., Spinsante, S., Goleva, R., Zdravevski, E. (2018). Recognition of activities of daily living based on environmental analyses using audio fingerprinting techniques: A systematic review. Sensors, Vol. 18(1), pp. 1–23.

41. Ren, F., Bao, Y. (2020). A review on human-computer interaction and intelligent robots. International Journal of Information Technology and Decision Making, Vol. 19(1), pp. 5–47.

42. Robinson, F. A., Bown, O., Velonaki, M. (2020). Implicit communication through distributed sound design: Exploring a new modality in human-robot interaction. Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction, ACM, pp. 597–599.

43. Shannon, C. E. (1948). A mathematical theory of communication. The Bell System Technical Journal, Vol. 27(3), pp. 379–423.

44. Shen, J.-l., Hung, J.-w., Lee, L.-s. (1998). Robust entropy-based endpoint detection for speech recognition in noisy environments. 5th International Conference on Spoken Language Processing, pp. 1–4.

45. Shen, Y.-H., He, K.-X., Zhang, W.-Q. (2018). Home activity monitoring based on gated convolutional neural networks and system fusion. DCASE2018 Challenge Tech. Rep., pp. 1–5.

46. Sigurdsson, S., Petersen, K. B., Lehn-Schiøler, T. (2006). Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music. 2006 International Society for Music Information Retrieval, pp. 1–4.

47. Smith, J. O., Abel, J. S. (1999). Bark and erb bilinear transforms. IEEE Transactions on Speech and Audio Processing, Vol. 7(6), pp. 697–708.

48. Telembici, T., Grama, L., Rusu, C. (2020). Integrating service robots into everyday life based on audio capabilities. 2020 International Symposium on Electronics and Telecommunications (ISETC), IEEE Xplore, pp. 1–8.

49. Traunmüller, H. (1990). Analytical expressions for the tonotopic sensory scale. The Journal of the Acoustical Society of America, Vol. 88(97), pp. 97–100.

50. Vafeiadis, A., Votis, K., Giakoumis, D., Tzovaras, D., Chen, L., Hamzaoui, R. (2017). Audio-based event recognition system for smart homes. 2017 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computed, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovation, IEEE Xplore, pp. 1–8.

51. Wei, P., He, F., Li, L., Li, J. (2020). Research on sound classification based on svm. Neural Computing and Applications, Vol. 32, pp. 1593–1607.

52. Zhang, X., Zou, Y., Shi, W. (2017). Dilated convolution neural network with leakyrelu for environmental sound classification. 2017 22nd International Conference on Digital Signal Processing, IEEE Xplore, pp. 1–5.

53. Zhang, Z., Xu, S., Cao, S., Zhang, S. (2018). Pattern Recognition and Computer Vision, chapter Deep Convolutional Neural Network with Mixup for Environmental Sound Classification. Springer, Cham, pp. 356–367.

Article received on 01/06/2021; accepted on 18/11/2021.
Corresponding author is Alain Manzo-Martínez.

ISSN 2007-9737

Analysis of CNN Architectures for Human Action Recognition in Video

David Silva, Alain Manzo-Martínez, Fernando Gaxiola,
Luis Gonzalez-Gurrola, Graciela Ramírez-Alonso

Universidad Autónoma de Chihuahua,
Facultad de Ingeniería, Chihuahua,
Mexico

david.a.silva.carnero@gmail.com,
{amanzo, lgaxiola, lcgonzalez, galonso}@uach.mx

Abstract. Every year, new Convolutional Neural Network (CNN) architectures appear to deal with different problems in the activity of image and video recognition. These architectures usually work along the ImageNet dataset for looking for the best performance of the CNNs without taking into account the video task where they are used. This can represent a problem if the task is Human Action Recognition (HAR) in video, since the CNN architectures are pre-trained with an image dataset that can practically contain any object, while HAR problem requires consecutive frames of people doing actions. To prove the idea that using CNNs pre-trained on an image dataset does not always achieve the best performance on a video dataset and that, therefore, it is worth comparing the performance of different CNNs under similar circumstances for the HAR problem, this work proposes an analysis between eight different CNN architectures. Each one of the CNN was exclusively trained with RGB images, which were extracted from the frames of the different classes of videos of HMDB51 dataset. To make the classification of an activity in video, we average the predictions taking into account the successes. We also made some ensembles with the best performance CNNs to measure the improvement in accuracy. Our results suggest that Xception is a strong baseline model that could be used by the community to make their comparisons of their proposals more robust.

Keywords. Human action recognition, convolutional neural network, HMDB51.

1 Introduction

In recent years, the problem of HAR has received a lot of attention from researchers. This is because today it is common to find problems related to video surveillance, behavior analysis or Human Computer Interaction (HCI) [1].

The first attempts to solve this problem using hand crafted features such as Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF) or Motion Boundary Histogram (MBH) [2-6]. However, the main issue of using these types of approaches is that it is difficult to transfer the handcrafted features of a training dataset to another [7]. This issue was solved by the introduction of convolutional neural networks (CNN), which are able to automatically detect features in raw images, to find the connection between them and use the learned features of a training dataset to train a different dataset [8-13].

With the breakthrough that CNN caused in 2012 in the machine vision community given their tremendous reduction of error rates of up to 20% to closest participants, it was clear that CNN would be the approach to exploit in image/video classification problems. In fact, two of the three most popular approaches (two-stream and 3DCNN) use CNN as a pillar while the third one uses recurrent neural networks [13].

One of the main questions when building a HAR model is which CNN to use, since every year there are new state-of-the-art CNN architectures on the ImageNet dataset. One may think that using the CNN with the top performance on the ImageNet dataset can achieve the best results. The main issue of thinking this way is that is not taking into consideration that the CNN was trained to classify images with any object class of the 1000 classes that the ImageNet dataset has and not frames of

human actions, which they are the main components of the videos in a HAR dataset. With this in mind, the main objective of this work is to prove that a CNN with the best performance on the ImageNet dataset does not always achieve the best results on a video dataset, thus it is important to test different CNNs under the same conditions when building a HAR model.

Regarding the originality of this study, we argue that even when new CNN models appear practically every year, very little is known regarding how these models compare to each other, or even against the previous competitive proposals, over the HAR problem, since no systematic and exhaustive experimental comparison, to the best of our knowledge, had been done until now.

This work makes an analysis of comparison about training time and accuracy of 8 different CNN architectures using different sets of RGB images that were built from the videos of the HMDB51 dataset. The CNN models were trained as image classifiers and it was used the average of the predictions of each image frame to generate the classification label of the activity in video. Lastly, different ensembles were considered using the best accuracy performance on the CNN architectures to prove if there is an increment in the accuracy using ensemble predictions.

The main contribution of this study is twofold. First, we empirically show that for a CNN having top performance on the Imagenet dataset does not imply top performance on the HAR task, as it usually is assumed by the community. This opens up important questions regarding what would be the best experimental setting for these neural models to achieve better results on such task.

Additionally, we tackle a long standing question for the HAR problem, which is to make the first exhaustive evaluation that considers up to 8 different state-of-the-art CNN-based approaches under very similar experimental settings that will allow to have the first impressions of who is who regarding performance and efficacy for Human Action Recognition endeavors. As a whole, with these results, the community will have enough evidence regarding what baseline model to use from now on, this being the Xception network, to compare their new contributions against.

2 Related Work

This section includes a description of previous works related to the HAR problem using the HMDB51 dataset. We made a revision based on the three main approaches for handling the HAR problem. It is important to note, that this work is not going to consider all the approaches revised here and the papers cited are only to tell the viewer what has been done in relation to HAR using the HDMB51 dataset and which CNNs are the most used among researchers on this field.

Two-stream approach was proposed by Simonyan et al. in 2014 [15] by the idea that they can have a CNN trained with raw RGB frames and another CNN trained with optical flow, which represents the moving vectors between two consecutive frames. They later combined the predictions of the two-streams using a weighted averaged of the predictions. Each stream had a CNN called ClarifaiNet and their best accuracy was 59.4%.

Wang et al. [16] decided to divide a video into 3 segments so that each segment have their own two-stream network and then combine the predictions of all segments of a certain stream and after that combine the stream predictions. All segments of all streams used the Inception-V2 CNN and they obtained an accuracy of 69.4%.

Zhu et al. [17] designed an architecture that was able to combine the feature vector of different frames into a video representation by using max pooling and a pyramidal layer. They also used Kinetics as the pre-trained dataset for the CNN, which result in better accuracy than using the ImageNet dataset. They also used Inception-V2 and their best result was an 82.1% in accuracy.

Cong et al. [18] developed an adaptive batch size K-step model averaging algorithm (KAVG). They customized the Adam optimizer and proposed to use a network to determine the best optical flow images from RGB frames. They attached that model to the two-stream network to form a three-stream network, which increases their accuracy even more. For the three streams, they used the ResNet152 network obtaining an 81.24% in accuracy.

He et al. [19] added an additional stream to the two-stream approach, which it was able to fuse the features of a frame with the features of its two

neighbor frames. This was done several times with the purpose of improve the frames features and that proved to be beneficial for the model. The CNN they chose was Inception-V2 obtaining a 73.1% in accuracy.

Wan et al. [1] proposed to combine the 3DCNN approach with the two-stream approach by using 3D convolutions on the spatial-stream and the VGG16 CNN on the temporal-stream. They also used a SVM after the combination of the two-stream features for the final prediction and obtained a 70.2% in accuracy.

Sun et al. [20] preferred to use a 3DCNN to model the relationship between the features of multiple frames, but instead of using a 3D convolution, they decided to divide the convolution in a set of 2D convolutions followed by a 1D convolution to model the temporal relationship between frames. They created their own 3D CNN and their best result was 59.1% in accuracy.

Carreira et al. [21] combined the two-stream approach with the 3DCNN approach by using a 3D CNN in both streams. They also proposed to use the Kinetics dataset for pre-training instead of the ImageNet dataset. The 3D CNN is based on Inception-V1 and it was called I3D CNN. The best result obtained was 80.9% in accuracy.

Wang et al. [22] used the I3D CNN proposed by Carreira et al. to build an architecture capable of learning the Fisher vector and bag of words representation of a combination of features extracted from RGB and optical flow frames. Their best result showed an 82.1% in accuracy.

Piergiovanni et al. [23] designed an evolving algorithm, which it was able to create convolutional models with different number and type of layers for the best detection of spatial and temporal features in videos. The model is based on Inception-V1 and the best result was 82.1% in accuracy.

Yang et al. [7] also attacked the computational cost of the 3D convolutions just like Sun et al. [20] but they used unidirectional asymmetric 3D convolutions. They also made their own CNN architecture achieving a 65.4% in accuracy.

Stroud et al. [24] proposed a model called D3D, which it was trained with RGB frames and with extracted knowledge of a temporal network trained with optical flow images. The CNN that they used was a 3D CNN called S3D-G and it is based on the I3D CNN. Their best result showed an 80.5% in accuracy.

Sharma et al. [25] chose to make a visual attention model using the Inception-V1 CNN as a feature extractor, an attention mechanism that was in charge of selecting which parts of the feature tensor were the most important ones and an RNN to model the relationship between the most important features of each frame. Their best result was a 41.3% in accuracy.

Ye et al. [26] proposed to combine the features of the last convolutional layer of each of the two ResNet101 CNN in a two-stream network and feed the combined feature vector to a convolutional LSTM to make the final prediction. They obtained a 69.3% in accuracy.

Outside HMDB51 dataset there is also numerous works on HAR in videos using different datasets. For example, He et al. [27] proposed to create a high accuracy architecture based on the integration of information from audio, RGB frames and two different types of optical flow images. They used ResNeXt101 and InceptionResNetV2 CNNs on their experiments. The Kinetics 400 database was used for pre-training and the final training and evaluation were on the Kinetics 600. The best accuracy was 85% using an ensemble of individual models.

Donahue et al. [28] decomposed the video into frames; each frame entered to a CNN to extract its characteristics and then passed to a LSTM. The prediction of each LSTM was averaged to have the final video tag. The base architecture in their experiments was a combination of the CaffeNet architecture and another network proposed by other authors. They got 82.37% accuracy in the UCF101 dataset.

Yue-Hei Ng et al. [29] experimented with various numbers of frames, various CNN architectures such as feature extractors (AlexNet and Inception-V1), various feature grouping architectures, and a recurring network in order to model a higher level of temporal features between frames in the video. They used the UCF101 database, obtaining an 88.6% accuracy.

Limin et al. [30] used UCF101 dataset. A valuable observation about their work is that they used 3 different CNNs (ClarifaiNet, GoogleNet and VGG16) on their experiments and the results showed that VGG16 outperforms GoogleNet,

which is interesting because the later performs better on ImageNet dataset. Although they did not train their CNNs under the same circumstances, due to the random corner-center cropping and random resizing techniques that they applied to the frames, this last work is an example of why we do not have to assume that a single CNN will be the best on every single dataset in existence.

3 Theoretical Background

3.1 Artificial Neural Networks (ANN)

The ANN is a machine learning (ML) algorithm based on the operation of neurons in the human brain. ANN uses mathematic equations to learn patterns of the training data and they are made up by the union of multiple units called perceptrons.

Frank Rosenblatt made the perceptron and he defined it as an artificial neuron that receives multiple inputs and produces one binary output that is feed to the next neuron. A perceptron also receives the name of neuron [31].

Equation (1) shows the process of calculating the output of a neuron, where $f(\cdot)$ represents the activation function, b represents the bias of the neuron, x_i represents the input i and w_i represents the weight associated to the input i:

$$output = f\left(b + \sum_{i=1}^{n} x_i w_i\right). \quad (1)$$

The following are some of the activation functions that can be used on a neuron [31]:

- Sigmoid function: It is an S-shaped function and it converts the input values into probabilities between 0 and 1.
- Softmax function: It is commonly used in the output layer of neural networks in classification algorithms. Computes the probability of the output being one of the target classes compared to the other classes.
- Tanh: This function represents the relationship between the hyperbolic sine and the hyperbolic cosine. It is S-shaped and converts input values to probabilities between -1 and 1.
- ReLU: It is conventionally used in the hidden layers of neural networks. It works in such a way that, if the input is greater than 0, the output is the same input value; if it is less than 0, the output is equal to 0.

The process of all the calculations that are made from left to right through all the neurons in an ANN is called “forward propagation”. The output of this process is used to generate the error of the network in comparison to the target. The error is used to adjust the network parameters (weight and bias), and that adjustment process is called back propagation [31].

3.2 Convolutional Neural Networks (CNN)

In an ANN each neuron in the input layer is connected to each neuron on the subsequent layer, this is known as a dense layer. However, in a CNN, a dense layer is not used until the last layers of the networks. In this way, a CNN can be defined as a neuronal network that exchanges a dense layer for a convolutional layer in at least one layer of the network [32].

A convolution can be defined as the sum of the element-wise multiplication between the values of the filters that overlap the values of the input tensor. A convolution takes into consideration the spatial relationship between pixels and its main goal is to extract useful features from the input tensor [33].

Nonlinear functions such as ReLU are applied to the output of the convolutions and then the new output is passed to the next layer and the process continues. A CNN also includes a pooling layer, which it helps to reduce the width and height of the input tensor [32].

Finally, the feature tensor is flattened to produce a 1-dimensional vector, which it is feed to one or more dense layers to make the predictions [32].

In practice, CNNs provide two key benefits: local invariance and compositionality. The concept of local invariance allows to classify an image that contains a particular object, regardless of where in the image the object appears. This local invariance is obtained by using "pooling layers" that identify regions in the input volume with a high response to a particular filter. The second benefit is compositionality. Each filter composes a local patch of lower-level features into a higher-level

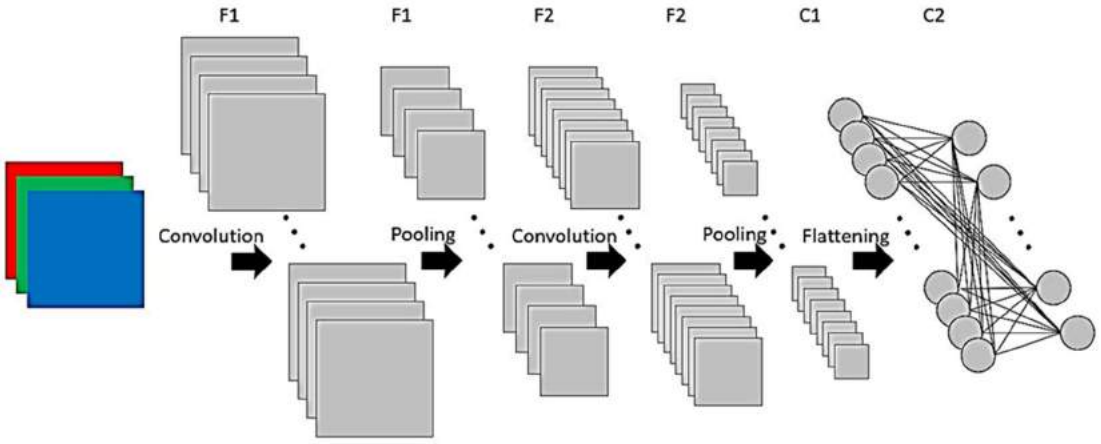


Fig. 1. Example of an architecture CNN

representation, similar to how you can compose a set of mathematical functions that are based on the output of previous functions. This composition allows the network to learn richer features and deeper into the network.

For example, the network can build edges from pixels, shapes from edges, and then complex objects from shapes, all in an automated manner that occurs naturally during the training process [32].

Figure 1 shows an example of an architecture CNN, where F1, F2 are the number of feature maps on each layer and C1 and C2 are the number of neurons in each dense layer.

3.2.1 CNN Architectures

In this section, we are going to review some important characteristics of the CNN architectures that will be considered later in the analysis. Since most of the literature revised use Inception-V1 or Inception-V2, we decided to consider the most similar one that belongs to the Keras library for python, which was Inception-V3.

Inception-V3 [34] is a 48-layer CNN and its main difference from the Inception-V2 CNN is that it uses RMSProp Optimizer, 7x7 factorized convolutions, batch normalization in the auxiliary classifiers and label smoothing, which is a type of regularizing component added to the loss formula that prevents the network from becoming too confident about a class avoiding the overfitting.

DenseNet201 [36] was selected because it is the best of all DenseNet CNNs and because it introduced the concept of dense connections between features maps. This proved to be beneficial because it solves the vanishing gradient problem as ResNet did and at the same time, it maintains the low-level features through all the convolutional layers within a dense block.

ResNet architectures family are also common in the revised works, we decided to consider ResNet152 [35] because we want to compare only the most accurate CNN within a group of related CNN. ResNet152 is a 152-layer CNN and its CNN family was the first that attacked the problem of vanishing gradient by using residual connections and residual blocks. A residual block is a stack of layers set in such a way that the output of a layer is taken and added to another layer deeper in the block using a residual connection. Finally, a non-linearity is applied to the result of the sum.

For the comparative analysis, the remaining CNN architectures were chosen from the Keras library, according to the next criteria. Since we work with the Keras library to get the previous CNNs architectures, we decided to also compare some of the other CNNs that the Keras library offers to work with.

The Xception [37] CNN stands for an extreme version of Inception and has 36 convolutional layers and was chosen because it proposed the use of modified Depthwise Separable Convolutions (DSC) with no intermediate non-linearity. These types of convolutions were stacked in the Xception model like Inception modules and the Xception accuracy on ImageNet probed to be

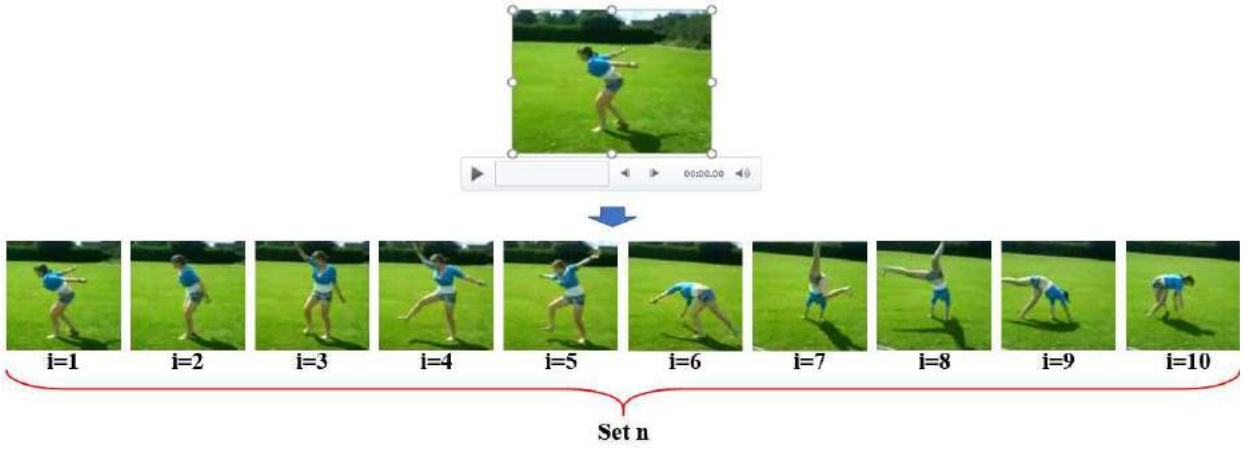


Fig. 2. Set of frames extracted per video

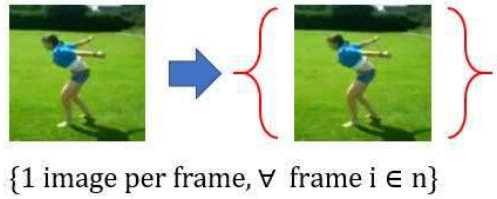


Fig. 3. No augmentation data was used for each frame in the first set of frames

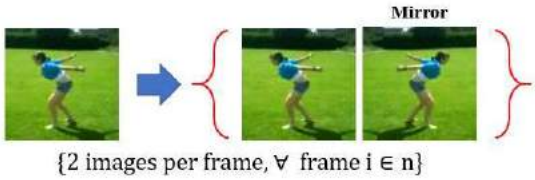


Fig. 4. Image augmentation for each frame in the second set of frames

better than InceptionV3, so we wanted to know if this feature was maintained with the HMDB51 dataset.

The EfficientNetB0 [38] and EfficientNetB3 [38] CNNs are part of a large family of CNNs known as EfficientNet. There are CNN architectures going from EfficientNetB0 all the way to EfficientNetB7.

At first, we used 3 CNNs of this family, EfficientNetB0, EfficientNetB3 and EfficientNetB7, but we decide to skip the use of EfficientNetB7 in the experiments because the difference in accuracy between EfficientNetB3 and EfficientNetB7 was not significant and the number of parameters increased.

The main contribution of this type of CNNs is the introduction of compound scaling which uniformly scales network width, depth, and resolution with a set of fixed scaling coefficients.

For instance, if the aim is to use 2^N times more computational resources, then the network depth can increase by α^N, width by β^N, and image size by γ^N, where α, β, and γ are constant parameters computed by a small grid search on the original small model.

EfficientNet uses a compound coefficient ø to uniformly scale the width, depth, and resolution of the network in a novel manner. The use of the compound scaling method is justified since, when

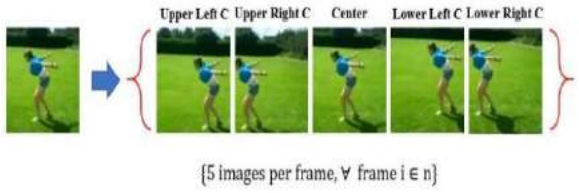


Fig. 5. Image augmentation for each frame in the third set of frames

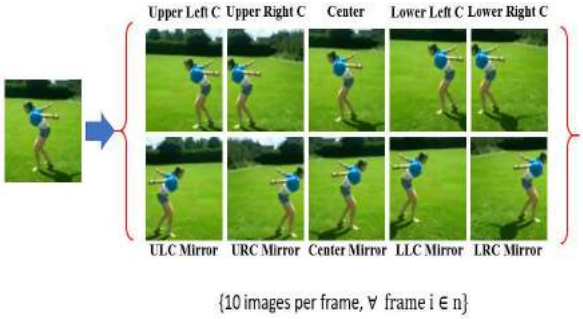


Fig. 6. Image augmentation for each frame in the fourth set of frames

the input image is bigger, the network needs additional layers and channels to increase the receptive field and to capture more fine-grained patterns on the bigger image, respectively.

Finally, we decided to select MobileNetV2 and NASNetMobile from Keras library to compare the accuracy of mobile CNNs in respect to the other ones. MobileNetV2 [39] is a 53-layer CNN and is the successor of MobileNetV1 which introduced the concept of DSC which dramatically decreased the number of parameters in the network. MobileNetV2 like its predecessor uses DSC, but non-linearities in narrow layers are removed this time, which was beneficial for the model classification performance. It also introduced inverted residual blocks (as opposed to ResNet) to improve parameter efficiency.

NASNetMobile [40] CNN is part of the NASNet CNN family and it was built using reinforcement learning using an RNN that selected the best combinations between a predefined set of states and actions, these combinations are called blocks. The main idea behind this approach was to make use of transfer learning by searching for an architectural building block that work on a small dataset (CIFAR10) and then transfer the block to a larger dataset (ImageNet). It is also introduce a new regularization technique called ScheduledDropPath which improved the generalization of the NASNet models.

4. Methodology

4.1 HMDB51 Dataset

HMDB51 is a 6766-video dataset with 51 human action classes and for each class there are at least 100 videos. The dataset has 3 sets of videos for training and testing.

The spatial resolution of the videos is 320x240 pixels. All videos were extracted from YouTube or digitalized movies. The dataset can be downloaded using this link[1].

4.2 Set of Frames

By the intuition that the random selection of frames in the training stage of a CNN affects the accuracy of the architecture, we decided to create 4 different sets of frames for the use in all CNNs. All frame sets described here used the videos from the set 1 of the HDMB51 dataset.

4.2.1. First Set of Frames

This set of frames was built taking 10 evenly spaced frames from each training and testing video. The frames extracted from the training videos were saved in a different folder that the ones extracted from the test videos.

To extract the frames, first the program gets the total number of frames in the video and this value is divided by 10 to calculate the position between each frame that will be extracted. Finally, the program saves the index of each frame and if the quotient between the frame index and the position between frames is zero, then the frame is extracted. Since the index for the first frame is 0, the first frame of the video is always extracted (see Fig. 2). For this set of frames, we did not include

[1] https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/#Evaluation

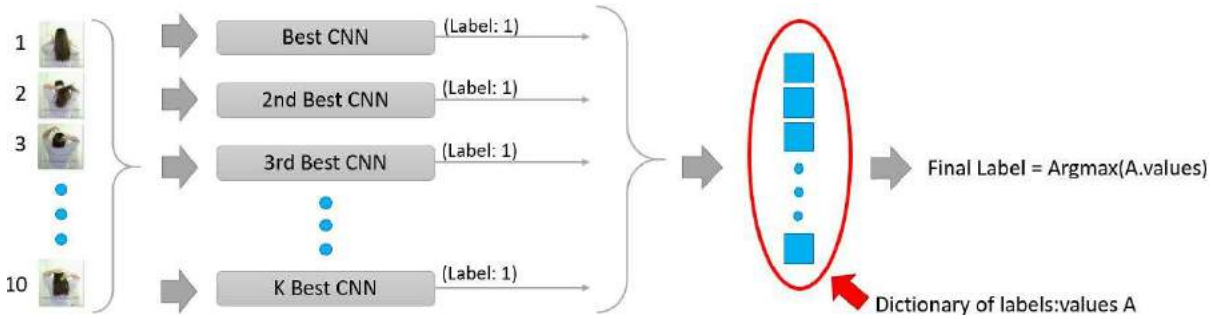


Fig. 7. Final label generated by simple voting

any sort of image augmentation, so there is only one image per frame (see Fig. 3).

4.2.2 Second Set of Frames

With the aim of evaluating the effect of data augmentation, we decided to use 3 different techniques. The first one used in this set of frames considers the horizontal mirror of each frame, so that instead of having 10 frames per video, this set of frames will have 20 frames (10 original and 10 horizontal mirror). The saving and extraction of the frames is as described in the first set of frames (see Fig. 4).

4.2.3 Third Set of Frames

The second data augmentation technique consists of resizing each frame to 256x256 pixels, and from the resized frame cut a 224x224 region from the center to each one of the four corners. This process generated 5 sub-frames from each frame, so at the end each video will have 50 frames (40 frames in total from all the corners and 10 central frames). The saving and extraction of the frames is as described in the first set of frames (see Fig. 5).

4.2.4 Fourth Set of Frames

The third data augmentation technique consists of resizing each frame to 256x256 pixels, and from the resized frame cut a 224x224 region from the center to each one of the four corners. After that, we took the horizontal mirror from each one of the 5 generated images. This process generated 10 subframes from each frame, so at the end each video will have 100 frames (80 frames in total from all the corners plus their mirrors and 20 frames from the center of each one and its mirror). The saving and extraction of the frames is as described in the first set of frames (see Fig. 6).

4.3 Ensembles

For the experiments with ensembles, we considered the best CNN architectures. Each ensemble was built by 3 or 5 CNN architectures and we used 5 different methods based on averaging and voting to obtain the final classification tag for each video. All CNN architectures were trained with the same set of frames. Each one of the 5 methods is described below.

4.3.1 Using Simple Voting with n Frames of Each Test Video

This method consists of extracting 10 frames evenly spaced and applying the corresponding image augmentation technique according to the set of frames used for training. Each frame of the set passes through all the CNNs in the ensemble and they generate a tag and a number which are added to a label dictionary. The key stored within the dictionary corresponds to the tag predicted by at least one of the CNNs and the number of classifiers that predicted that tag. To obtain the final tag, the voting method was used where the tag that had the most votes by CNNs within the

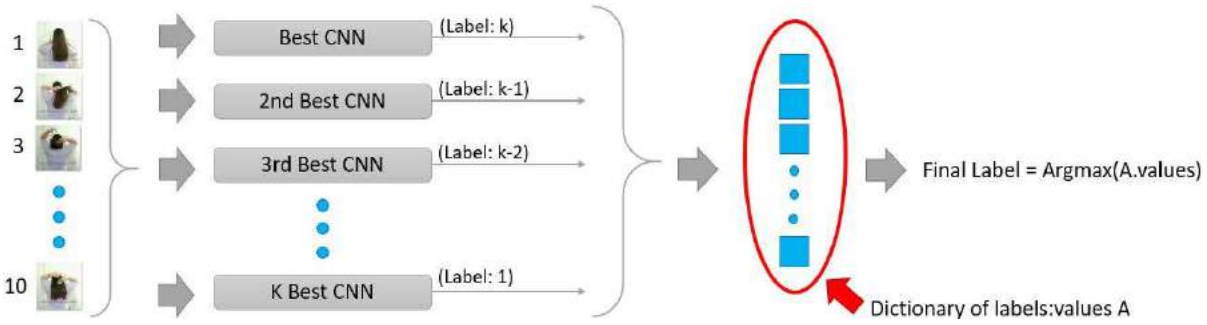


Fig. 8. Final label generated by weighted voting

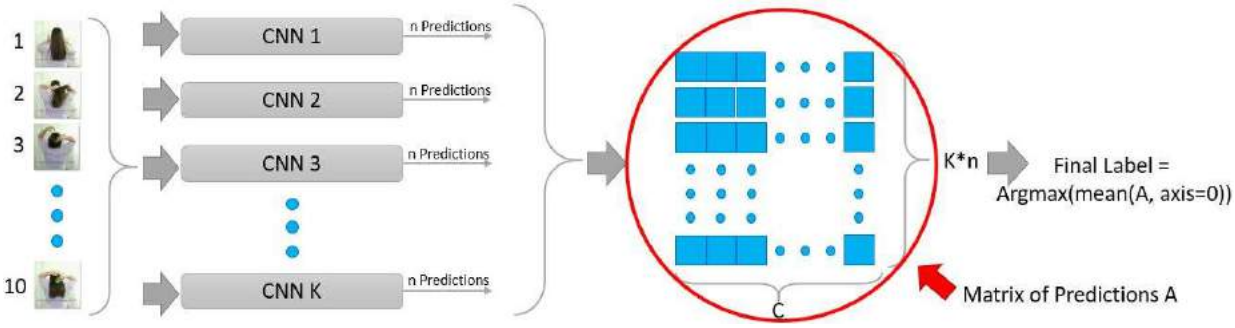


Fig. 9. Final label generated by prediction averaging

dictionary is used to establish the final tag for the video (see Fig. 7).

4.3.2 Using Simple Voting with All Frames of Each Test Video

This method works exactly as the previous method; the only difference is that instead of using only 10 frames we used all the frames in the test video.

This was done with the purpose of seeing the change in accuracy when considering all the frames of the video.

4.3.3 Using Weight Voting with n Frames of Each Test Video

This method consists of extracting 10 frames evenly spaced and applying the corresponding image augmentation technique according to the set of frames used for training.

Each frame of the set passes through all the CNNs in the ensemble and they generate a tag and a value which are added to a label dictionary.

The key stored within the dictionary corresponds to the tag predicted by at least one of the CNNs and the value is the weight referred to the CNN. The weight of each CNN was determined according to its individual performance in experiments before ensembles.

The best CNN has a weight of k, the second best has a weight of k-1, and this continues until we reached the weight of 1.

This was done for proving if exist any significant difference between simple and weighted voting method when taking in consideration the individual performance of each CNN.

To obtain the final tag for the video, we looking for the tag with the greater score within the dictionary (see Fig. 8).

Table 1. Average accuracy of each optimizer in each fold using the best models and the first set of frames

Optimizer	Model	Fold 1	Fold 2	Fold 3	Average
Adagrad	Val_Loss	43.49%	39.12%	40.22%	40.95%
	Val_Acc	44.03%	39.70%	40.69%	41.48%
Adam	Val_Loss	39.48%	37.42%	37.89%	38.26%
	Val_Acc	44.67%	43.29%	42.80%	43.59%
Nadam	Val_Loss	42.58%	40.60%	36.00%	39.73%
	Val_Acc	47.19%	**45.90%**	43.21%	45.43%
RMSprop	Val_Loss	43.12%	38.45%	40.95%	40.84%
	Val_Acc	42.73%	40.54%	42.84%	42.04%
SGD	Val_Loss	43.90%	40.02%	39.74%	41.22%
	Val_Acc	**47.26%**	45.23%	**44.41%**	**45.63%**

Table 2. Average accuracy of each optimizer in each fold using the best models and the second set of frames

Optimizer	Model	Fold 1	Fold 2	Fold 3	Average
Adagrad	Val_Loss	45.83%	42.65%	40.97%	43.15%
	Val_Acc	46.05%	42.75%	41.29%	43.36%
Adam	Val_Loss	44.76%	43.01%	43.59%	43.78%
	Val_Acc	50.59%	49.67%	48.24%	49.50%
Nadam	Val_Loss	48.57%	43.53%	43.83%	45.31%
	Val_Acc	**53.80%**	50.03%	49.41%	51.08%
RMSprop	Val_Loss	41.89%	39.25%	40.11%	40.42%
	Val_Acc	43.87%	39.23%	42.04%	41.71%
SGD	Val_Loss	49.04%	45.62%	44.26%	46.31%
	Val_Acc	53.71%	**51.39%**	**49.71%**	**51.60%**

4.3.4 Using Prediction Averaging with n Frames of Each Test Video

This method consists of extracting 10 frames evenly spaced and applying the corresponding image augmentation technique according to the set of frames used for training. Each frame of the set passes through all the CNNs in the ensemble.

Each CNN generates *n* predictions, which are stored in a matrix of predictions. The final tag for the video is obtained by averaging all the predictions of all CNN in the prediction matrix, which gives us a vector with *C* classes.

Finally, we took the index that has the greater value in the vector to generate the corresponding tag (see Fig. 9).

4.3.5 Using Prediction Averaging with All Frames of Each Test Video

This method works exactly as the method of section 4.3.4; the only difference is that instead of using only 10 frames we used all the frames in the test video.

This was done with the purpose of seeing the change in accuracy when considering all the frames of the video.

4.4 Our Proposal

The main purpose of this work is to make an analysis of comparison about training time and accuracy of 8 different CNN architectures using the HMDB51 dataset.

By no means has it intended to achieve a better accuracy than cited works. The training of each CNN architecture was done by using only RGB frames from the set of videos of the HMDB51 dataset.

By this statement, our proposal is to treat each CNN as an image classifier leaving aside the three popular approaches for the HAR problem. To make a prediction of human action in the video, we averaged the predictions of all the frames in a given test video.

4.5 Environment Setup

We used python as the programming language. Regarding training variables, we used the default input tensor dimension for all CNNs and the batch size that we used was set to 16.

For the learning rate, we used the default value that each optimizer has in Keras; refer to https://keras.io/api/optimizers/ for more details. Since the default image size is 299x299 for InceptionV3 and Xception CNNs, we resized the frames to have that size and take advantage of the pre-trained weights of those CNNs. All other CNNs work with 224x224 size.

We ran the experiments in 3 different computers each one with a different GPU. The GPUs that we used were as follows: NVIDIA 1060, NVIDIA 1080 and NVIDIA TITAN RTX. The main libraries used were: Keras implementation in TensorFlow, efficientnet, NumPy, OS, OpenCV, Shutil and Pickle.

We used Tensorflow as a backend to run the CNN experiments. We loaded most of the CNN architectures using the module of Keras library except for the EfficientNet CNNs in which case we used the efficientnet library. The metric that was used to measure the performance of the CNNs was accuracy in all experiments. In some experiments, we also measured the training time in different GPUs for all the CNNs.

5 Experiments and Results

The first experiment consists of selecting the best optimizer to use in all CNNs. For this, we considered 5 different optimizers: “Adagrad”, “Adam”, “Nadam”, “RMSprop and “SGD” with their default values in Keras library. We used a K-Fold of 3 to validate our results. For this, we divided the training videos of set 1 in 2 folders, 70% of the videos were used for training and 30% for validation.

Since each one of the 51 folders (classes) in the HMDB51 dataset has 70 videos, each class was divided in 49 videos for training and 21 videos for validation. To extract the frames of each video in both the training and validation folders, we used the process described in 4.2.1 section.

Each fold was run 5 times to mitigate the bias produced by random weight initialization on the CNNs that was used. Due to the excessive time that it would take to train 15 times each CNN for each optimizer, we decided to use a less deep CNN only for the first three experiments. The CNN architecture is the following [41]:

- An input layer where the dimension for the input tensor is 224x224x3.
- A convolutional layer with 32 filters of 3x3 followed by a “ReLU” activation function and a 2x2 maxpooling.
- A second convolutional layer with 32 filters of 3x3 followed by a “ReLU” activation function and a 2x2 maxpooling.
- A third convolutional layer with 64 filters of 3x3 followed by a “ReLU” activation function and a 2x2 maxpooling.
- A flatten layer followed by a dense layer of 64 neurons with “ReLU” activation function, a Dropout layer with a value of 0.5 and a final dense layer with 51 neurons with a “softmax” activation function.

The training was done for 50 epochs and the model with the best validation accuracy (Val_Acc) as well as the model with best validation loss (Val_Loss) were saved. For the validation phase, the final label of the video was obtained by averaging the predictions of the *n* frames generated of each validation video. The results showed the average accuracy of the 5 best

Table 3. Running time in seconds of each CNN architecture on a Titan RTX and 1080 GPU.

CNN	TITAN RTX	1080
EfficientNetB0	9133 s	18717 s
EfficientNetB3	17464 s	35321 s
Xception	28549 s	61730 s
InceptionV3	12033 s	26869 s
ResNet152	20878 s	49355 s
DenseNet201	15239 s	33675 s
MobileNetV2	**7034 s**	**12913 s**
NASNetMobile	14025 s	25519 s

Table 4. Prediction, updating and training time of Xception CNN and InceptionV3 CNN

CNN	Prediction	Updating	Training
Xception	389.17 s	1538.89 s	1928.06 s
Inception	**226.90 s**	**612.48 s**	**839.38 s**

Val_Acc and the 5 best Val_Loss models in the validation set on each fold (see Table 1).

The second experiment was realized almost exactly as the first one, but instead of extracting the frames using the previous process, we use the process of section 4.2.2. This was done with the main purpose of observing if any optimizer performs better than others when considering a larger number of frames (see Table 2).

Based on the previous results, we decided to use SGD optimizer for the training of all the CNNs in the next experiments. For the third experiment, we measure the training time of all the CNNs architectures with the frames of the set 1. We trained each of the CNNs 3 times for 50 epochs. The CNNs were pre-trained with the ImageNet dataset. We reported the average running time of each CNN in seconds when using a GPU 1080 and a GPU TITAN RTX (see Table 3).

From Table 3 we can observe that the fastest CNN is the MobileNetV2 architecture, which is understandable because it contains the lowest number of parameters. An interesting fact is that the Xception CNN is from 2 to 3 times slower than the Inception CNN and they share almost the same number of parameters. We decided to conduct another experiment to understand why that happened.

For the fourth experiment, we used the GPU 1060 and computed the average of the training time of 5 epochs on both CNNs. We also measured the average prediction time of 5 epochs, which was calculated by measuring how much time the CNN needed to make a prediction of all the training frames. Finally, with both times we calculated the time that the CNNs used to update their weights by extracting the average prediction time to the average training time (see Table 4).

With the previous results, we observed that even if both CNNs share almost the same number of parameters, the inner structure of the Xception CNN made the network slower than the InceptionV3, especially when we compared the updating time of both networks.

In the fifth experiment, we trained all the CNNs with each one of the four different sets of frames. For the third and fourth set of frames, we decided to train only four CNNs due to the excessive training time since this process is carried on using one GPU. We trained each one of the CNNs three times for 50 epochs. The CNNs were pre-trained with the ImageNet dataset. For each set of frames, we used the corresponding testing frames for

validation and we saved the model with the best validation accuracy for testing. For the testing phase, the final label of each video is obtained by averaging the predictions of the *n* frames generated from each test video according to the set of frames used during training. We reported the average accuracy of the 3 runs of each CNN on the testing videos for each set of frames used during training (see Table 5).

According to the results of the previous experiment, we noticed that all the CNNs were benefited from using the second set of frames, but on the third one, only two of the four CNNs improved their accuracy. What is even more interesting and that none of the four CNNs that were trained on the fourth set of frames improved their performance, instead of that, the performance was worse than when using the second and third set of frames.

This result can be explained by the fact that when building the first and second set of frames, we worked with the whole image, but when building the third and fourth set of frames, we took five different subsections of the whole image and some of them did not contain the person doing the action. With this in mind, we can argue that both the third and fourth set of frames have many frames with noise, and that is why the accuracy performance on these two sets was affected negatively. Based on that information, we decided to not train any of other remaining CNNs on the 3rd and 4th set of frames. Since the 2nd set of frames prove to be the set with better results on the CNNs, we used that set for the training of the CNNs for the next experiments.

The sixth experiment was done with the purpose of proving how well the CNNs perform on the set 2 and set 3 of videos of the HMDB51 dataset. Aiming at this, we used the procedure described in section 4.2.2 to generate new sets of frames from the sets 2 and 3 of videos of the HMDB51 dataset. We trained each one of the CNNs 3 times for 50 epochs. The CNNs were pre-trained with the ImageNet dataset. For each set of frames, we used the corresponding testing frames for validation and saved the model with the best validation accuracy for testing. For the testing phase, the final label of each video was obtained by averaging the predictions of the 20 frames generated from each test video according to the set of frames that was used during training. We reported the average accuracy of the 3 runs of each CNN in the testing videos for each set of frames used during training and included the results obtained in the set 1 of videos using the second set of frames from the previous table (see Table 6).Something that caught our attention on the result of the fifth and sixth experiment was the fact that the Xception network proved to be better than the EfficienNetB3 network, which is on a higher rank on the ImageNet dataset.

For knowing if these results were caused by the greater entry resolution of the Xception network, we decided that the aim of the seventh experiment would be to compare these two networks with the same input resolution.

We fixed the input resolution of each of the two CNNs to be 224x224 and trained both CNN for 50 epochs on the second set of frames of the set 1 of videos of HMDB51. The CNNs were pre-trained with the ImageNet dataset. We used the testing frames of the second set of frames for validation and saved the model with the best validation accuracy for testing. For the testing phase, the final label of each video was obtained by averaging the predictions of the 20 frames generated from each test video. We reported the average accuracy of the 3 runs of each CNN in the testing videos (see Table 7).

Based on the previous results, we can see that even when both CNNs have the same input resolution, the Xception CNN managed to outclass the EfficientNetB3 CNN by a significant margin.

We can also see that the Xception network works better with a 299x299 input image resolution and that is due that input images of 299x299 resolution were used to generate the pre-training weights of the Xception CNN on the ImageNet dataset.

The eighth experiment was envisioned to take advantage of those CNN models that showed the best performance on previous evaluations. For this, we thought of building several ensembles made of such CNN's to evaluate if they could, as a team, outperform the best individual model for HAR, that is, Xception. If that is the case, then, this ensemble can also be considered as a baseline for future evaluation. Since we trained 3 times each CNN, we selected the CNN model which test accuracy was the closest to the average that was

Table 5. Average accuracy of each CNN on each set of frames

CNN Architecture	1st set	2nd set	3rd set	4th set
EfficientNetB0	47.39%	49.67%	51.35%	48.17%
EfficientNetB3	47.84%	50.70%	N/A	N/A
Xception	**51.33%**	**53.99%**	**52.68%**	**50.26%**
InceptionV3	48.21%	48.56%	48.39%	46.95%
ResNet152	44.81%	45.80%	N/A	N/A
DenseNet201	45.95%	45.99%	N/A	N/A
MobileNetV2	42.53%	43.75%	N/A	N/A
NASNetMobile	43.86%	44.47%	45.40%	44.18%

Table 6. Average accuracy of each CNN on each of the three set of videos of HMDB51 dataset.

CNN Architecture	Set 1 HMDB51	Set 2 HMDB51	Set 3 HMDB51	Average
EfficientNetB0	49.67%	45.62%	45.66%	46.98%
EfficientNetB3	50.70%	46.97%	45.88%	47.85%
Xception	**53.99%**	**50.00%**	**51.76%**	**51.92%**
InceptionV3	48.56%	46.25%	47.25%	47.35%
ResNet152	45.80%	39.96%	40.46%	42.07%
DenseNet201	45.99%	42.42%	43.75%	44.05%
MobileNetV2	43.75%	41.33%	41.22%	42.10%
NASNetMobile	44.47%	40.04%	40.76%	41.76%

Table 7. Accuracy of EfficientNetB3 and Xception using 224x224 images.

CNN Architecture	Accuracy
EfficientNetB3	50.70%
Xception	**52.46%**

reported on each set of videos to be part of the ensembles.

We decided to separate the ensembles that considered only *n* frames (*n* = 20) of each test video and the ones that considered all video frames of each test video. The ensembles were formed in the following way:

– Ensemble 1: Ensemble of the 3 best CNNs using simple voting and *n* frames.
– Ensemble 2: Ensemble of the 5 best CNNs using simple voting and n frames.
– Ensemble 3: Ensemble of the 3 best CNNs using weighted voting and n frames.
– Ensemble 4: Ensemble of the 5 best CNNs using weighted voting and n frames.
– Ensemble 5: Ensemble of the 3 best CNNs using prediction averaging and n frames.

Table 8. Accuracy of the different type of ensembles on each set of videos of HMDB51 dataset

Ensemble	Set 1 HMDB51	Set 2 HMDB51	Set 3 HMDB51	Average
Ensemble 1	52.88%	48.30%	50.00%	50.39%
Ensemble 2	53.92%	49.80%	50.26%	51.33%
Ensemble 3	54.31%	50.39%	51.31%	52.00%
Ensemble 4	54.64%	50.98%	51.50%	52.37%
Ensemble 5	**54.77%**	51.11%	51.90%	52.59%
Ensemble 6	52.94%	50.33%	51.90%	51.72%
Ensemble 7	52.75%	51.37%	51.18%	51.77%
Ensemble 8	53.27%	50.65%	50.59%	51.50%
Ensemble 9	54.64%	**53.07%**	**52.42%**	**53.38%**
Ensemble 10	52.29%	52.03%	52.22%	52.18%

Table 9. Comparison with previous models

Paper	Acc.
I3D Spatial stream [20]	74.80%
Two-stream Conv LSTM CNN Spatial-stream [25]	64.80%
KAVG Spatial stream [17]	61.44%
LSF CNN Spatial stream [1]	61.30%
DTPP Spatial stream [16]	61.06%
F_{ST}CN [19]	59.10%
TSN Spatial stream [15]	53.70%
Best Ensemble	**53.38%**
LFN Spatial stream [18]	52.14%
Best Individual CNN	**51.92%**
Visual Attention Model [24]	41.30%
Spatial stream [14]	40.50%

– Ensemble 6: Ensemble of the 5 best CNNs using prediction averaging and n frames.
– Ensemble 7: Ensemble of the 3 best CNNs using simple voting and all frames.
– Ensemble 8: Ensemble of the 5 best CNNs using simple voting and all frames.
– Ensemble 9: Ensemble of the 3 best CNNs using prediction averaging and all frames.
– Ensemble 10: Ensemble of the 5 best CNNs using prediction averaging and all frames.

We reported the average accuracy of each ensemble in each set of videos of the HMDB51 dataset (see Table 8).

5.1 Statistical Tests

To verify the robustness of the results of the first and second experiments, three paired t-test were conducted. The first one compared the vector containing the average of hits per class from the 5 runs using the set 1 of frames, the SGD optimizer

and the best Val_Acc model, against the vector containing the average of hits per class from the 5 runs using the set 1 of frames, the SGD optimizer and the best Val_Loss model. The p-value obtained was 5.512e-09.

The second paired t-test compared the vector containing the average of hits per class from the 5 runs using the set 1 of frames, the Adagrad optimizer and the best Val_Acc model, against the vector containing the average of hits per class from the 5 runs using the set 1 of frames, the SGD optimizer and the best Val_Acc model. The p-value obtained was 1.079e-09.

The third paired t-test compared the vector containing the average of hits per class from the 5 runs using the set 1 of frames, the SGD optimizer and the best Val_Acc model, against the vector containing the average of hits per class from the 5 runs using the set 2 of frames, the SGD optimizer and the best Val_Acc model. The p-value obtained was 2.2e-16. With all these values, we rejected all the null hypotheses, and thus show the robustness of the results.

5.2 Comparison of Results with Previous Works

To see where our best performance individual CNN and ensemble with no fine-tuning stand against the fine-tuned models of the state of the art of the HMDB51 dataset, we decided to make a comparison with 10 of the most accurate or most popular models of the state of the art.

For a fair comparison, we only cited the models that used only RGB frames as input (see Table 9). Something to take in consideration is that our work never intended to compete with the results of the state of the art, but instead to demonstrate that the idea of choosing the best performance CNN trained with an image dataset will not always lead to the best performance on a video dataset.

6 Conclusions

In this work, we compared the performance of eight different CNNs on the different sets of frames generated on the HMDB51 dataset. The Xception network proved to be the best individual CNNs out of all the CNNs that we chose to work with. We argued that this was because of the absence of non-linearity on the intermediate step of a DSC, which is the main difference between the Xception and the rest of the CNN. However, further experiments modifying this feature of the Xception CNN need to be done on the HMDB51 dataset to see if this network feature is truly the reason behind the good performance of the CNN.

Results also showed that mobile CNNs such as MobileNetV2 and NASNetMobile are low on accuracy when comparing to newer and bigger models such as Xception or EfficientNets. The best accuracy achieved was 53.38% when we used the ensemble of the best three individual CNNs, prediction averaging and when we took into consideration all frames of a video during testing.

We proved that the performance of a CNN above others in terms of accuracy can change depending on the dataset that is used, i.e., between the ImageNet dataset and the HMDB51 dataset. Thus, we encourage the authors to include the election of the CNN (at least experiment with 2 different CNN) as a hyperparameter of their models.

We also proved that considering more frames that were created using image augmentation techniques during training does not necessarily improve the accuracy of a network such as happen when the CNNs used the third and fourth set of frames; but taking into consideration, more frames during testing time can achieve better results like what happened with the ensembles.

We have trained each CNN using only 10 frames per video along to corresponding extra frames generated with the image augmentation techniques, nevertheless, we encourage to use more frames to improve classification accuracy on each CNN. Results can also be improved with the fine-tuning of hyperparameters such as learning rate and batch size, with the use of regularization techniques such as dropout and with the use of different data augmentation techniques such as RGB and scale-jittering.

Due to the use of image classifier models, the accuracy on classes like sit, stand, walk, run and others very similar classes was very low, because the model was not made to capture the motion feature that distinguishes one class from another. However, these results can be improved with the use of more complex models that include motion

features like optical flow or with the use of any of the three popular approaches for HAR previously explained. Running time of the CNNs can also be improved with more powerful GPUs and with the use of a cluster of GPUs.

For future work, we would like to test more CNN architectures with more different video datasets like UCF101. We will also like to include motion for the training of the CNN by extracting optical flow of the frames and training a temporal stream or by building a 3D CNN with every CNN tested.

The idea of including motion is because we want to see if the current ranking of CNNs that we have in our experiments by using only RGB frames changes either with the use of only motion data or with the inclusion of motion data with RGB frames. We also want to test with more than 10 frames per video for training, so we would like to analyze the performance of CNNs while using different number of frames.

Finally, as previously stated we would like to run different tests on the best CNN to see what part of its inner structure is responsible of its performance.

Acknowledgments

This work was supported by CONACyT and UACH – FING through the Thesis titled as "Analysis of state-of-the-art architectures CNN for activity recognition in video".

References

1. **Wan, Y., Yu, Z., Wang, Y., Li, X. (2020).** Action Recognition Based on Two-Stream Convolutional Networks with Long-Short-Term Spatiotemporal Features. IEEE Access, Vol. 8, pp. 85284–85293. DOI: 10.1109/ACCESS.2020.2993227.
2. **Laptev, I. (2005).** On space-time interest points. International Journal of Computer Vision, Vol. 64, pp. 107–123. DOI: 10.1007/s11263-005-1838-7.
3. **Dollár, P., Rabaud, V., Cottrell, G., Belongie, S. (2005).** Behavior Recognition Via Sparse Spatio-Temporal Features. 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 65-72. IEEE. DOI: 10.1109/VSPETS.2005.1570899.
4. **Willems, G., Tuytelaars, T., Van Gool, L. (2008).** An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector. In: **Forsyth, D., Torr, P., Zisserman, A.,** editors, Computer Vision – ECCV'08. Lecture Notes in Computer Science, Vol 5303. Springer, pp. 650–663 DOI: 10.1007/978-3-540-88688-4_48
5. **Wang, H., Kläser, A., Schmid, C., Liu, C.L. (2011).** Action Recognition by Dense Trajectories. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3169–3176. DOI: 10.1109/CVPR.2011.5995407.
6. **Wang, H., Schmid, C. (2013).** Action Recognition with Improved Trajectories. Proceedings of the IEEE International Conference on Computer Vision, pp. 3551–3558.
7. **Yang, H., Yuan, C., Li, B., Du, Y., Xing, J., Hu, W., Maybank, S.J. (2019).** Asymmetric 3d Convolutional Neural Networks for Action Recognition. Pattern Recognition, Vol. 85, pp. 1–12. DOI: 10.1016/j.patcog.2018.07.028.
8. **Varela-Santos, S., Melin, P. (2021).** A New Approach for Classifying Coronavirus COVID-19 Based on its Manifestation on Chest X-rays Using Texture Features and Neural Networks. Information Sciences, Vol. 545, pp. 403–414. DOI: 10.1016/j.ins.2020.09.041.
9. **He, K., Zhang, X., Ren, S., Sun, J. (2016).** Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.
10. **Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erham, D., Vanhoucke, V., Rabinovich, A. (2015).** Going Deeper with Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.
11. **Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012).** Imagenet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, Vol. 25, pp. 1097–1105.

12. **Simonyan, K., Zisserman, A. (2014).** Very Deep Convolutional Networks for Large-Scale Image Recognition. Computer Vision and Pattern Recognition (cs.CV), DOI: 10.48550/arXiv.1409.1556.
13. **Huang, G., Liu, Z., van der-Maaten, L., Weinberger, K.Q. (2017).** Densely Connected Convolutional Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700-4708.
14. **Xie, D., Deng, C., Wang, H., Li, C., Tao, D. (2019, July).** Semantic Adversarial Network with Multi-scale Pyramid Attention for Video Classification. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, No. 1, pp. 9030–9037. DOI: 10.1609/aaai.v33i01.33019030.
15. **Simonyan, K., Zisserman, A. (2014).** Two-Stream Convolutional Networks for Action Recognition in Videos. Advances in Neural Information Processing Systems, Vol. 27, pp.1–9.
16. **Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L. (2016, October).** Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In: **Leibe, B., Matas, J., Sebe, N., Welling, M. (eds)** Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, Vol. 9912. Springer, Cham. DOI: 10.1007/978-3-319-46484-8_2.
17. **Zhu, J., Zhu, Z., Zou, W. (2018).** End-to-end Video-Level Representation Learning for Action Recognition. 24th International Conference on Pattern Recognition (ICPR), pp. 645–650. DOI: 10.1109/ICPR.2018.8545710.
18. **Cong, G., Domeniconi, G., Yang, C.C., Shapiro, J., Zhou, F., Chen, B. (2019).** Fast Neural Network Training on a Cluster of GPUs for Action Recognition with High Accuracy. Journal of Parallel and Distributed Computing, Vol. 134, pp. 153–165. DOI: 10.1016/j.jpdc.2019.07.009.
19. **He, F., Liu, F., Yao, R., Lin, G. (2019).** Local Fusion Networks with Chained Residual Pooling for Video Action Recognition. Image and Vision Computing, Vol. 81, pp. 34–41. DOI: 10.1016/j.imavis.2018.12.002.
20. **Sun, L., Jia, K., Yeung, D.Y., Shi, B.E. (2015).** Human Action Recognition Using Factorized Spatio-temporal Convolutional Networks. Proceedings of the IEEE International Conference on Computer Vision, pp. 4597–4605.
21. **Carreira, J., Zisserman, A. (2017).** Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308.
22. **Wang, L., Koniusz, P., Huynh, D.Q. (2019).** Hallucinating Idt Descriptors and I3d Optical Flow Features for Action Recognition with Cnns. Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8698–8708.
23. **Piergiovanni, A.J., Angelova, A., Toshev, A., Ryoo, M.S. (2019).** Evolving Space-Time Neural Architectures for Videos. Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1793–1802.
24. **Stroud, J., Ross, D., Sun, C., Deng, J. & Sukthankar, R. (2020).** D3d: Distilled 3d networks for video action recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 625-634.
25. **Sharma, S., Kiros, R., Salakhutdinov, R. (2015).** Action recognition using visual attention. arXiv preprint arXiv:1511.04119.
26. **Ye, W., Cheng, J., Yang, F. & Xu, Y. (2019).** Two-stream convolutional network for improving activity recognition using convolutional long short-term memory networks. IEEE Access, 7, 67772-67780.
27. **He, D., Li, F., Zhao, Q., Long, X., Fu, Y. & Wen, S. (2018).** Exploiting Spatial-Temporal Modelling and Multi-Modal Fusion for Human Action Recognition.
28. **Donahue, J., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., Darrell, T. (2015).** Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2625–2634.

29. **Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R. Toderici, G. (2015).** Beyond Short Snippets: Deep Networks for Video Classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4694–4702.
30. **Limin, W., Xiong, Y., Zhe, W., Yu, Q. (2015).** Towards Good Practices for Very Deep Two-Stream ConvNets.
31. **Saleh, H. (2019).** Applied Deep Learning with PyTorch. Birmingham: Packt Publishing Ltd.
32. **A. Rosebrock. (2017).** Deep Learning for Computer Vision with Python (1st ed.). PyImageSearch.
33. **Bai, K. (2019).** A Comprehensive Introduction to Different Types of Convolutions in Deep Learning. Towards Data Science.
34. **Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erham, D., Vanhoucke, V., Rabinovich, A. (2015).** Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9.
35. **He, K., Zhang, X., Ren, S., Sun, J. (2016).** Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.
36. **Huang, G., Liu, Z., van der-Maaten, L., Weinberger, K.Q. (2017).** Densely Connected Convolutional Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708.
37. **Chollet, F. (2017).** Xception: Deep Learning with Depthwise Separable Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258.
38. **Tan, M., Le, Q. (2019, May).** EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. International Conference on Machine Learning, PMLR. pp. 6105–6114.
39. **Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C. (2018).** Mobilenetv2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520.
40. **Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V. (2018).** Learning Transferable Architectures for Scalable Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 8697-8710.
41. **Chollet, F. (2016).** Building Powerful Image Classification Models Using Very Little Data. The Keras Blog.

Article received on 07/06/2021; accepted on 17/11/2021.
Corresponding author is Alain Manzo-Martinez.

ISSN 2007-9737

Fuzzy Flower Pollination Algorithm: Comparative Study of Type-1 and Interval Type-2 Fuzzy Logic System in Parameter Adaptation Optimization

Hector Carreon-Ortiz, Fevrier Valdez, Oscar Castillo

Tijuana Institute of Technology,
Mexico

fevrier@tectijuana.mx

Abstract. State-of-the-art algorithms are competitive, because they get the most out of available resources. Metaheuristic algorithms solve optimization problems from a search space. The proposal in this research work is to use the algorithm bio-inspired by nature Flower Pollination Algorithm (FPA) for the optimization of the membership functions of an Interval Type-2 Fuzzy Logic system, which we will call IT2FLS-FPA (Interval Type-2 Fuzzy Logic System-Flower Pollination Algorithm). This work is presented to continue with one that we developed before [6], in this investigation we made a comparison between a non-optimized IT2FLS-FPA and an optimized IT2FLS-FPA where we demonstrate that the latter is better by means of statistical hypothesis tests.

Keywords. Bioinspired algorithm, flower pollination algorithm, optimization, interval type-2 fuzzy logic.

1 Introduction

Optimization minimizes or maximizes a function by randomly choosing the values of the variables within an admissible range [94, 96]. Research continues to develop algorithms that achieve the above purpose. The development of algorithms for real problems is of interest to many research studies. In the beginning optimization techniques used gradient based algorithms, where the main idea was to find a range of solutions near the origin [2, 55], these methods provide accurate solutions and fast convergence, better than stochastic approaches. The problem is that this type of algorithms will only tend to local minima and not to the global minimum.

The resource constraint is faced in daily competition to all types of systems, in this struggle different strategies have been employed to change the established order. Optimization is used to handle the problem of limited resources (producing more with less). In the search for optimization, goals must be achieved with few resources [3, 4]. The objective function is the set goal that varies depending on the problem [5, 6]. The goal of optimization is to find the parameter values that minimize or maximize a specific objective, for example, in an engineering design is to find the parameter values that satisfy the needs of the design with minimum cost, optimization solves this type of requirement.

FPA is a very popular optimization method among researchers because of its characteristics as it has few parameters and has demonstrated a robust performance when applied to various optimization problems, that is why we decided to use this metaheuristic inspired by nature, besides that we have worked previously with this algorithm [7] and has proven to be very good and we can see it in the work done by [8, 17, 36, 37, 41, 42, 50, 51, 52, 64, 87, 88, 89, 90], there are variants of FPA developed by [59, 60, 61, 62, 63], in Figure 1 we can see a graphical summary of the variants [86], also hybrid algorithms have been developed with the FPA as [64, 65, 66, 67, 68], the applications of the FPA in the areas Chemical Engineering, for thermodynamic systems [69], in petroleum industry [70] where FPA is one of the effective algorithms in this area, in the preparation of triaxial porcelain from Palm Oil Fuel Ash (POFA) [71] and POFA

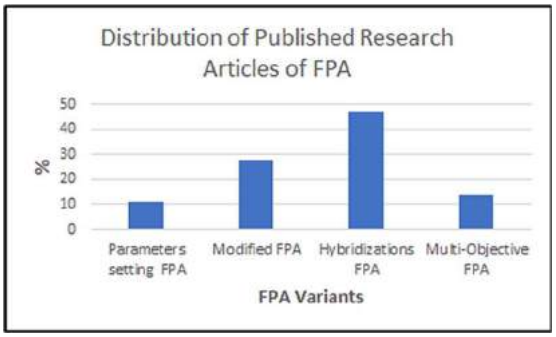


Fig. 1. Distribution of published research articles of FPA

was used as the cement filler for enhancing the EMI absorption of cement-based composites [72].

In civil engineering it is one of the most important areas of applied optimization, because nonlinear design problems with complex constraints, costs, architectural design constraints, physical requirements often generate a complex engineering problem [73, 74, 75]. In mechanical engineering FPA has contributed in solving speed reducer, gear train, tension-compression spring design problems using hybrid algorithms with FPA with local search [66, 74, 76, 77, 78].

In Electronical and Communication Engineering, metaheuristic methods have also been employed in wireless communication systems such as [79, 80], using global pollination, enhanced local pollination and dynamic shift probability FPA was improved by [81], also FPA was used to solve radio spectrum optimization problems. In Energy and Power Systems, Dubey et al [59] modified FPA to solve practical power system test cases, Prathiba et al [82] employed FPA to minimize fuel cost in a bus system.

Lenin et al [67] hybridized FPA with harmony search algorithm to optimize reactive power dispatch. In Computer Science, FPA was employed in image compression Kaur et al [83], for multilevel image FPA was used for Ouadfel and Taleb-Ahmed [84].

A binary FPA was employed for Rodrigues et al [85] for solutions across the corner in electro encephalogram, Jensi and Jiji [68] proposed a hybrid approach combining K-Means algorithm and FPA that finds the center of the optimal cluster.

The main contribution of this work is to use the metaheuristic Flower Pollination Algorithm (FPA) and the Type 2 Fuzzy Logic System (IT2FLS) to dynamically adjust the parameters of the FPA in order to obtain better results than in the previous work [6], where experiments with FPA and Fuzzy Logic were performed.

In other published works [30, 31, 32, 33, 34] and in the most recent ones [35, 36] it has been shown that the use of Fuzzy Logic can be better because good results are obtained than not using it but using (IT2FLS) in the dynamic adaptation of the parameters in the FPA metaheuristic is much better by the results obtained in this research.

The remainder of this article is organized as follows. Section 2 describes works that other authors have done on the FPA algorithm and the IT2FLS method, Section 3 gives a very general review of the bio-inspired algorithms, Section 4 is basic information on the FPA algorithm, Section 5 describes the origin and development of the type-1 and interval type-2 fuzzy systems, Sections 6 and 7 present the model and the proposed parameters, in Section 8 we show the results of this research and finally in Section 9 we present the conclusions of this paper.

2 Related Works

Several researches on the Flower Pollination Algorithm (FPA) and a Fuzzy System (FS) have been developed in the 8 years, one of them is where the optimization of the parameters of the membership functions is performed using the FPA algorithm to simulate the motion of a robot [8], according to [7] in the simulation the FPA algorithm calls the model and, in the process, updates the variables. In another paper [9] where a hybrid approach for fire outbreak detection based on FPA algorithm and IT2FLS using meteorological parameters is proposed.

According to fire information, numerous grammatical uncertainties can be assumed in type-2 membership functions, so that the accuracy of fuzzy systems can be increased [50]. In a work we conducted in 2020 [6] where we used the FPA Algorithm and a FS to solve a water tank control problem, by means of the FPA algorithm, the parameters of the membership functions of the fuzzy system simulating the water tank were optimized.

3 Bioinspired Optimizations

Bio-inspired optimization is based on biological systems, which have been the inspiration for solving optimization problems. The subsets of natural computation according to [10, 53] are biological computation and optimization. Metaheuristic optimization simulates the biological behaviors of animals or plants and has been used to find the optimal solution to a problem. A metaheuristic is a heuristic strategy to solve complex optimization problems.

Optimization methods according to Fevrier Valdez, et al. 2020 [11, 58] in 1960 Holland at the University of Michigan started working with Genetic Algorithms (GAs) [12, 93, 95], in 1995 Eberhart and Kennedy, inspired by the social behavior of bird flocking or fish schooling, developed Particle Swarm Optimization (PSO) [13], in 1983 Kirkpatrick et al. And in 1985 Cerny proposed the simulated annealed probabilistic (SA) method [14] and the Pattern Search developed by Robert Hooke and T. A. Jeeves [15] is a family of numerical optimization methods that does not require the gradient of the problem to be optimized, so it can be used on functions that are not continuous or differentiable.

4 Flower Pollination Algorithm (FPA)

FPA was developed by Xin-She Yang in 2012, inspired by the pollination process of flowering plants [16, 17, 36, 37], let us analyze the general pollination behavior of plants, there are two forms of pollination: biotic and abiotic.

Biotic pollination: pollen is transported to the stigma by insects and animals. Abiotic pollination: wind and water are the means of pollination. Research says that 10% of pollination has an abiotic pollination process and, therefore, does not require any pollinator.

There are two ways of pollination: self-pollination and cross-pollination. Self-pollination occurs from the pollen of the same flower or from different flowers of the same plant, in this process local pollination occurs and cross-pollination occurs through a flower of a different plant and can occur over long distances by means of bees, bats, birds, flies, etc., which fly long distances, these pollinators make global pollination possible [17, 38, 39, 40]. The author of the algorithm describes the flower constancy and the behavior of pollinators in the pollination process with the following four rules [17, 41, 42, 43, 44, 45]:

1. Global pollination process takes place by biotic and cross-pollination and pollinators perform Lévy flights [17, 46, 47].
2. Local pollination process is considered abiotic and self-pollinating.
3. Flower constancy, pollinators visit plants with specific flowers to increase reproductive success.
4. A probability of change p ∈ [0, 1] that controls global and local pollination.

The basic idea is the fact that each plant has a flower and each flower originates a gamete, so it is established is that it is not necessary to distinguish between a plant, a flower or a gamete [51]. Birds, insects, etc. (pollinators) can fly enormous distances for biotic and cross-pollination to occur. Lévy's flight perfectly describes the flight of pollinators, rules 1 and 3 using Levy's distribution [52] describe global pollination to plot random step sizes (L) as Eq. (1).

The mathematical modeling of the 4 rules is as follows; the processes of global pollination (Rule 1) and flower constancy (Rule 3) are represented by the following equation:

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(g^* - x_i^t), \tag{1}$$

where:

x_i^{t+1} is the generated solution vector at iteration $t+1$, x_i^t is the pollen i or the solution vector x_i at iteration t, g^* is the current best solution, γ is a scaling factor used to control the step size, $L(\lambda)$ is the Lévy flights-based step size, it corresponds to the strength of the pollination. In reality, pollinators can fly long distances with different lengths (step size), this can be modeled using a Lévy distribution according to the following equation:

$$L \sim \frac{\lambda \Gamma(\lambda) sin\left(\frac{\pi\lambda}{2}\right)}{\pi} \frac{1}{s^{1+\lambda}} \quad (s \gg s_0 > 0), \tag{2}$$

where:

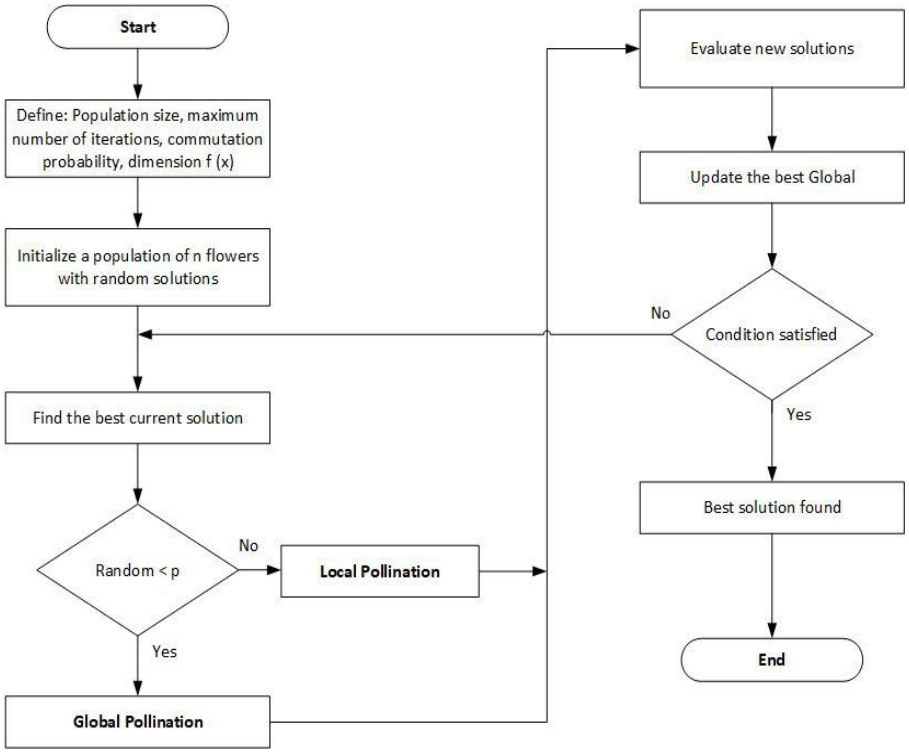


Fig. 2. Flower Pollination Algorithm Flowchart

Objective function f(x), x = (x1, x2, ..., xd)
Initialize a population of ***n*** *flowers in random solutions*
Find the best solution ***g**** *in the initial population*
Define a switch probability ***p*** $\in$ *[0, 1]*
while ($t < t_{max}$)
  for i =1: n (all n flowers in the population)
    if *rand* < *p*,
      Draw a d-dimensional step vector ***L*** *which obeys a Lévy distribution*
      Global pollination via (1)
    else
      Draw ϵ *from a uniform distribution in [0,1]*
      Randomly choose j and k among all the solutions
      Do local pollination via (3)
    end if
    Evaluate new solutions
    If new solutions are better, update them in the population
  end for
  Find the current best solution $g*$
end while

Fig. 3. Pseudo code of the proposed Flower Pollination Algorithm (FPA)

$\Gamma(\lambda)$ is the standard gamma function, and this distribution is valid for large steps $s > 0$.

Local pollination (Rule 2), and flower constancy (Rule 3) can be represented as follows:

$$x_i^{t+1} = x_i^t + \varepsilon(x_j^t - x_k^t), \quad (3)$$

where:

x_j^t and x_k^t are pollen gametes obtained from different flowers of the same plant species, randomized ϵ between 0 and 1 to approximate this selection to a local random walk. $\left(x_j^t - x_k^t\right)$ is used to imitate the flower constancy in a limited neighborhood.

Fourth rule, flower pollination processes can occur randomly at all scales, both in the local and global case. Therefore, to emulate this biorientation, a switching parameter p chosen randomly from [0,1] can be effectively used.

In the following, the flowchart and pseudocode of the flower pollination algorithm are shown in Figures 2 and 3.

5 Fuzzy Logic

Uncertainty, doubt, skepticism, suspicion, imprecision, approximation and distrust mean lack of certainty about someone or something. Uncertainty can range from lack of certainty to almost total lack of conviction or knowledge, especially about an outcome. Uncertainty has always been present in human life, one of the main advances on uncertainty in recent years is the introduction of fuzzy logic, which means a deep understanding of approximate reasoning [18], the origin of fuzzy logic comes from fuzzy set theory, its principles come from two sources of the last century [19]:

– First: Charles S. Peirce, who applied the term "Logic of vagueness" and was unable to develop and complete his theory before his death [19]. The mathematician and philosopher Max Black (1937) took up the concept of "Logic of vagueness" [20]. In 1923, the philosopher Bertrand Russell proposed that "vagueness" is a matter of value [21]. Therefore, the "Logic of vagueness" turned out to be the subject of interest of other researchers such as Brock [22], Nadin [23, 24], Engel-Tiercelin [25] and Merrell [26, 27] [19]

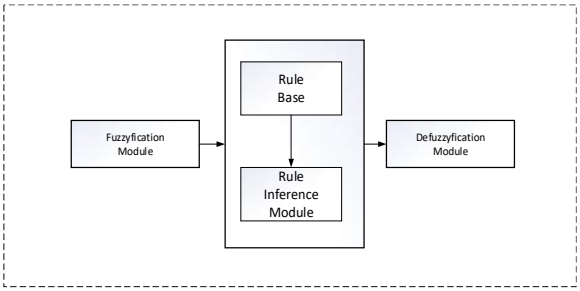


Fig. 4. General sketch for a fuzzy controller

– Second source: the mathematician Lofti A. Zadeh used in 1960 for the first time the term "Fuzzy Sets" and continued to develop this idea for the next 40 years, in his first paper published in 1965 on Fuzzy Sets [28], it was the beginning of a new stage of his scientific career, in the publication of this first article, the answer generated in the scientific community a lot of controversy.

From 1965 onwards, all published articles focused on the process and use of the fuzzy set thesis [19]. Professor Richard Bellman, renowned mathematician, was its main advocate and one of its most important contributors to the analysis and control of systems, in general this theory "Fuzzy Sets" was received with hostility and skepticism.

5.1 Basic Theory

The idea of Fuzzy Logic is not to determine whether the variable X is true or false, but to determine to what degree ∈ [0,1] it is true. We call this degree of certainty the degree of membership, although in some texts it is called possibility and, in this case, special emphasis is usually made on the difference between probability (empirical measure of the frequency with which an observation is repeated in a set of measurements) and possibility (degree of membership of an observation to a fuzzy set), we also speak of confidence level since it is the degree to which we are sure that the observation belongs to the defined set. The degree of membership is assigned by the membership function (f: X → R).

Fuzzy set theory allows us to gradually evaluate the membership of elements relative to a set. The

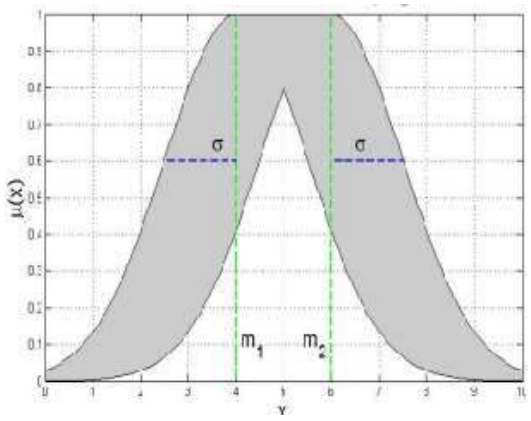


Fig. 5. Membership Function for IT2

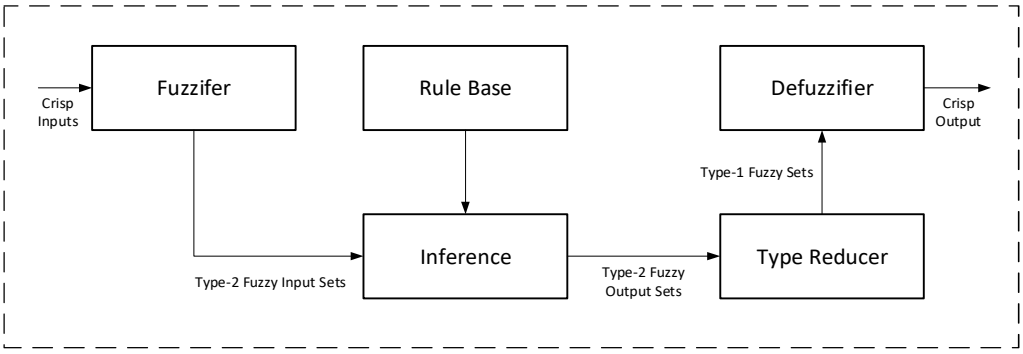


Fig. 6. Scheme of a Type-2 FLC

fuzzy set 'A' in a nonempty space X(A⊆X) can be defined as [19]:

$$A = \{(x, u_A(x)) | X \in U\}, \quad (4)$$

where $u_A: X \rightarrow [0,1]$ is a function of each element of *X* that establishes the extent to which it belongs to the set *A*. This function is called the membership function of the fuzzy set A.

Figure 4 shows us the basic structure of a fuzzy control system [96, 97, 98], which are detailed below:

– Fuzzification: Fuzzifies the system inputs.
– Rule base: Contains the selection of fuzzy rules.
– Mechanism of inference (Rule Inference Module): It contains a database that defines the membership functions used in the rules and a reasoning mechanism that performs the inference procedure (fuzzy reasoning).
– Defuzzification: Converts the (fuzzy) result of the inference process to a real value (crisp) within the domain of the output variable [31].

5.2 Interval Type-2 Fuzzy Logic System

It is known that Type-2 fuzzy systems (T2FLS) allow us to model and minimize the effects of uncertainty in Type-1 Fuzzy Systems (T1FLS) [32, 49]. We also understand that Type-2 fuzzy systems are more difficult to use and understand, so their use is still not very common [48, 54, 56, 98].

Figure 6 shows structure of Type-2 Fuzzy Control System T2FLS [1, 33, 34, 35, 57, 92]:

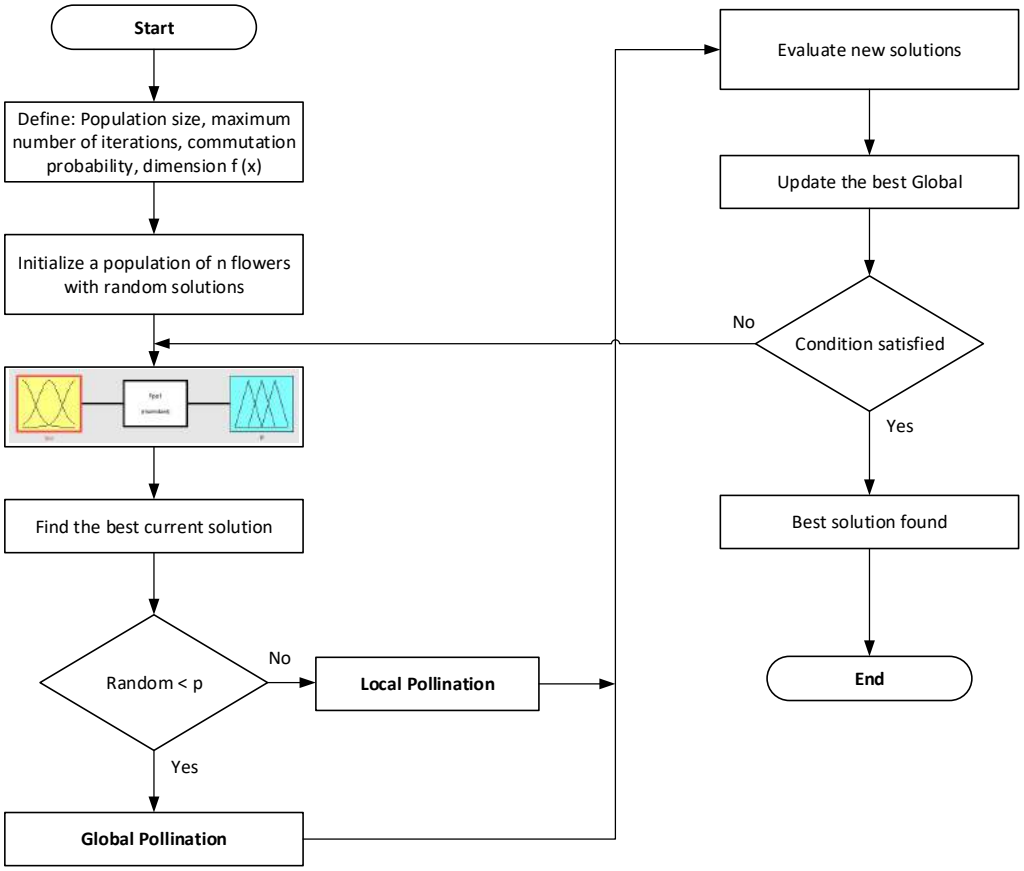


Fig. 7. Fuzzy Flower Pollination Algorithm Flowchart

$$\tilde{A} = \{(x, u_{\tilde{A}}(x))|\forall x \in X\}, \quad (5)$$

The interval type 2 fuzzy sets proposed by Zadeh [91] and continued by Liang and Mendel [35], provide the mathematical approach to handle uncertainty by means of a secondary domain describing the uncertainty of the data. Mathematical equation of IT2FLS (6):

$$\tilde{A} = \{((x, u), 1)|\forall x \in X, \forall u \in Jx \subseteq [0,1]\}, \quad (6)$$

X is the primary domain representing the degree of membership of the fuzzy set and Jx is the secondary domain related to the uncertainty and is always equal to 1. An IT2 MF can be defined from two limiting T1 MFs, and they are known as the upper MF and the lower MF and the Footprint of Uncertainty (FOU) (Mendel and John, 2002), which is the area between the two, and Figure 5 illustrates these concepts. In an IT2 FIS, the inference is very similar to that of a T1 FIS.

6 Mathematical Modeling of Fuzzy Flower Pollination Algorithm (FFPA)

Figure 7 shows the FPA flow diagram where the type-2 fuzzy system is included in the algorithm process by intervals.

7 Parameter Adaptation

This research uses the optimization algorithm inspired by the nature FPA, the method to optimize the parameters applies small adjustments in the optimization process, for the adjustment of the

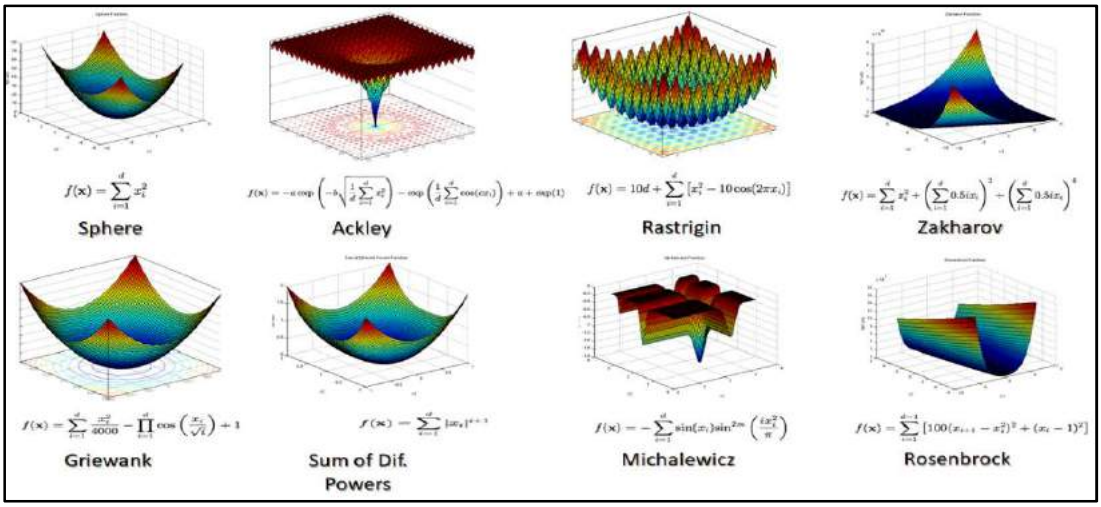


Fig. 8. Benchmark functions that were used for the experiments

parameters, it uses Interval Type-2 Fuzzy Logic System (IT2FLS) to verify the value of one or more parameters in each iteration of the algorithm.

The fuzzy system uses as input the percentage of iterations in which p is evaluated, to know the new values of the parameters and thus to know if it is a global or local pollination.

To evaluate the error of these metrics, the parameter E (epsilon) is used, which represents the flower constancy, all these parameters are used as input for the fuzzy system defined by equations (7) and (8):

$$Iteration = \frac{Current\ iteration}{Maximum\ of\ iteration}, \tag{7}$$

Mean Square Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2, \tag{8}$$

where:

$Y_i = Current\ result\ at\ time\ i.$

$\hat{Y}_i = Forecast\ of\ the\ value\ at\ instant\ i.$

$n = Total\ number\ of\ samples\ considered.$

8 Simulation Results

All the experiments carried out in this investigation were done with 8 mathematical functions, Figure 8, shows the 8 Benchmark functions: Sphere, Ackley, Rastrigin, Zakharov, Griewank, Sum of Different Powers, Michalewickz and Rosenbrock, FPA-T1FLS and FPA-IT2FLS as indicated in the table.

Tables 1 and 2 show the results: Best, Worse, Mean, and Standard Deviation for 30 and 100 dimensions of the FPA-T1FLS.

Tables 3 and 4 show the results: Best, Worse, Mean, and Standard Deviation for 30 and 100 dimensions of the non-optimized FPA-IT2FLS.

Tables 5 and 6 show the results: Best, Worse, Mean, and Standard Deviation for 30 and 100 dimensions of the FPA-IT2FLS optimized for the FPA13T2330 architecture.

Tables 7 and 8 show the results: Best, Worse, Mean, and Standard Deviation for 30 and 100 dimensions of the FPA-IT2FLS optimized for the FPA13T2B130 architecture.

Table 9 shows the results: Best, Worse, Mean, Standard Deviation and Z-Test, for 30 dimensions, hypothesis tests were performed with non-optimized FPA-T1FLS and FPA-IT2FLS, it can be observed that only in 4 mathematical functions there were significant evidence that FPA-IT2FLS is better than FPA-T1FLS.

Table 10 shows the results: Best, Worse, Mean, Standard Deviation and Z-Test, for 100

Table 1. Experiments with FPA and T1FLS

30 – Dimensions – FPA-T1FLS				
Function	**Best**	**Worse**	**Mean**	**Std**
1-Sphere	1.650E-04	2.080E-02	4.350E-03	5.000E-03
2-Ackley	1.690E-02	2.210E+00	7.620E-01	9.830E-01
3-Rastrigin	2.630E-02	1.070E+02	9.800E+00	2.530E+01
4-Zakharov	2.790E-03	1.380E-01	3.890E-02	3.670E-02
5-Griewank	2.530E-06	6.310E-04	1.410E-04	1.320E-04
6-Sum of Dif Powers	5.010E-17	2.510E-11	1.120E-12	4.550E-12
7-Michalewicz	-9.520E+00	-1.310E+01	-1.180E+01	8.360E-01
8-Rosenbrock	2.370E+01	4.690E+01	3.230E+01	4.370E+00

Table 2. Experiments with FPA and T1FLS

100 – Dimensions – FPA- T1FLS				
Function	**Best**	**Worse**	**Mean**	**Std**
1 Sphere	3.570E-01	1.490E+00	7.520E-01	2.830E-01
2 Ackley	3.960E-01	7.430E-01	5.890E-01	7.740E-02
3 Rastrigin	5.890E-01	1.030E+00	7.910E-01	1.060E-01
4 Zakharov	1.110E+00	4.100E+00	2.080E+00	6.090E-01
5 Griewank	2.660E-03	6.530E-03	4.020E-03	8.870E-04
6 Sum of Dif Powers	6.630E-14	1.090E-08	7.840E-10	2.170E-09
7 Michalewicz	-1.400E+01	-2.350E+01	-1.910E+01	2.050E+00
8 Rosenbrock	1.270E+02	2.170E+02	1.680E+02	2.090E+01

Table 3. Experiments with FPA and non-optimized IT2FLS

30 – Dimensions – FPA-IT2FLS				
Function	**Best**	**Worse**	**Mean**	**Std**
1-Sphere	3.910E-04	2.150E-02	5.100E-03	5.150E-03
2-Ackley	9.390E-02	1.220E-01	7.270E-02	3.390E-02
3-Rastrigin	3.170E+01	3.840E+01	3.500E+01	1.880E+00
4-Zakharov	1.010E-02	1.240E-01	5.230E-02	3.010E-02
5-Griewank	2.880E-05	6.800E-04	1.820E-04	1.520E-04
6-Sum of Dif Powers	1.210E-17	3.570E-11	1.650E-12	6.450E-12
7-Michalewicz	-8.510E+00	-1.170E+01	-9.760E+00	7.620E-01
8-Rosenbrock	7.470E+01	1.110E+02	8.920E+01	9.260E+00

dimensions, hypothesis tests were performed with non-optimized FPA-T1FLS and FPA-IT2FLS, it can be observed that only in 5 mathematical functions there were significant evidence that FPA-IT2FLS is better than FPA-T1FLS.

Table 11 shows the results: Best, Worse, Mean, Standard Deviation and Z-Test, for 30 dimensions, hypothesis tests were performed with FPA-T1FLS and optimized FPA-IT2FLS (FPA13T2330), it can be observed the following.

Table 4. Experiments with FPA and non-optimized IT2FLS

100 – Dimensions – FPA-IT2FLS				
Function	**Best**	**Worse**	**Mean**	**Std**
1-Sphere	5.710E-01	1.640E+00	1.010E+00	2.900E-01
2-Ackley	5.950E-01	4.950E-01	5.700E-01	1.090E-01
3-Rastrigin	2.260E+02	2.540E+02	2.360E+02	6.470E+00
4-Zakharov	1.300E+00	3.450E+00	2.190E+00	4.740E-01
5-Griewank	5.760E-03	1.440E-02	9.690E-03	2.270E-03
6-Sum of Dif Powers	6.690E-15	4.510E-08	3.010E-09	8.940E-09
7-Michalewicz	-1.260E+01	-1.640E+01	-1.460E+01	9.380E-01
8-Rosenbrock	1.080E+03	1.320E+03	1.210E+03	5.390E+01

Table 5. Experiments with FPA and optimized IT2FLS

30 – Dimensions – FPA-IT2FLS – FPA13T2330				
Function	**Best**	**Worse**	**Mean**	**Std**
1.Sphere	1.410E-03	5.229E-02	7.968E-03	9.250E-03
2.Ackley	4.223E-02	1.671E-01	8.219E-02	3.312E-02
3.Rastrigin	2.900E+01	3.840E+01	3.434E+01	2.194E+00
4.Zakharov	6.703E-03	1.502E-01	5.799E-02	3.535E-02
5.Griewank	4.354E-05	1.100E-03	3.038E-04	2.270E-04
6.Sum of Dif Powers	2.488E-19	8.314E-09	3.116E-10	1.491E-09
7.Michalewicz	-1.129E+01	-8.452E+00	-9.932E+00	7.098E-01
8.Rosenbrock	9.051E+01	1.379E+02	1.077E+02	1.099E+01

Table 6. Experiments with FPA and optimized IT2FLS

100 – Dimensions – FPA-IT2FLS – FPA13T2330				
Function	**Best**	**Worse**	**Mean**	**Std**
1 Sphere	4.395E-01	1.486E+00	9.833E-01	2.426E-01
2 Ackley	4.748E-01	1.030E+00	7.034E-01	1.275E-01
3 Rastrigin	2.273E+02	2.624E+02	2.476E+02	8.315E+00
4 Zakharov	1.210E+00	3.469E+00	2.106E+00	5.325E-01
5 Griewank	6.794E-03	2.323E-02	1.362E-02	3.384E-03
6 Sum of Dif Powers	3.800E-14	3.892E-07	2.653E-08	8.098E-08
7 Michalewicz	-1.686E+01	-1.313E+01	-1.474E+01	1.008E+00
8 Rosenbrock	1.412E+03	1.714E+03	1.549E+03	6.847E+01

Only in 7 mathematic functions there was significant evidence that the FPA-IT2FLS (FPA13T2330) is better than the FPA-T1FLS. Table 12 shows the results: Best, Worse, Mean, Standard Deviation and Z-Test, for 100 dimensions, hypothesis tests were performed with FPA-T1FLS and optimized FPA-IT2FLS (FPA-IT2330), it can be observed that only in For 7

Table 7. Experiments with FPA and optimized IT2FLS

30 – Dimensions – FPA-IT2FLS – FPA13T2B130				
Function	**Best**	**Worse**	**Mean**	**Std**
1.Sphere	5.676E-04	2.330E-02	6.731E-03	5.871E-03
2.Ackley	2.583E-02	1.895E-01	8.226E-02	3.306E-02
3.Rastrigin	2.839E+01	3.940E+01	3.478E+01	2.408E+00
4.Zakharov	1.476E-02	1.722E-01	5.126E-02	3.528E-02
5.Griewank	2.525E-05	9.736E-04	2.843E-04	2.096E-04
6.Sum of Dif Powers	1.491E-17	4.638E-10	1.915E-11	8.446E-11
7.Michalewicz	-1.075E+01	-8.854E+00	-9.916E+00	4.734E-01
8.Rosenbrock	9.574E+01	1.376E+02	1.132E+02	1.028E+01

Table 8. Experiments with FPA and optimized IT2FLS

100 – Dimensions – FPA-IT2FLS – (FPA13T2B130)				
Function	**Best**	**Worse**	**Mean**	**Std**
1-Sphere	5.798E-01	1.327E+01	9.797E+00	4.478E+00
2-Ackley	4.947E-01	8.192E-01	6.573E-01	9.349E-02
3-Rastrigin	2.315E+02	2.643E+02	2.521E+02	7.748E+00
4-Zakharov	8.883E+01	1.077E+02	9.939E+01	4.088E+00
5-Griewank	7.521E-03	2.346E-02	1.283E-02	3.657E-03
6-Sum of Dif Powers	5.444E-15	3.533E-08	1.470E-09	6.361E-09
7-Michalewicz	-1.828E+01	-1.275E+01	-1.489E+01	1.142E+00
8-Rosenbrock	1.474E+03	1.802E+03	1.638E+03	8.748E+01

Table 9. Comparison FPA-T1FLS with non-optimized FPA-IT2FLS

COMPARATIVE 30 Dim	**FPA-T1FLS**		**FPA-IT2FLS**		**Z-Test**
Function	**Mean**	**Std**	**Mean**	**Std**	
1.Sphere	4.350E-03	5.000E-03	5.100E-03	5.150E-03	**N**
2.Ackley	7.620E-01	9.830E-01	7.270E-02	3.390E-02	**Y**
3.Rastrigin	9.800E+00	2.530E+01	3.500E+01	1.880E+00	**Y**
4.Zakharov	3.890E-02	3.670E-02	5.230E-02	3.010E-02	**N**
5.Griewank	1.410E-04	1.320E-04	1.820E-04	1.520E-04	**N**
6.Sum of Dif Powers	1.120E-12	4.550E-12	1.650E-12	6.450E-12	**N**
7.Michalewicz	-1.180E+01	8.360E-01	-9.760E+00	7.620E-01	**Y**
8.Rosenbrock	3.230E+01	4.370E+00	8.920E+01	9.260E+00	**Y**

mathematical functions, there was significant evidence that the FPA-IT2FLS (FPA-IT2330) is better than the FPA-T1FLS.

Table 13 shows the results: Best, Worse, Mean, Standard Deviation and Z-Test, for 30 dimensions, hypothesis tests were performed with FPA-T1FLS and optimized FPA-IT2FLS (FPA-IT2B130), it can be observed that only in For 6 math functions, there was significant evidence that the FPA-IT2FLS (FPA-IT2B130) is better than the FPA-T1FLS. Table 14 shows the results: Best, Worse, Mean, Standard Deviation and Z-Test, for 100

Table 10. Comparison FPA-T1FLS with non-optimized FPA-IT2FLS

COMPARATIVE 100 Dim	FPA-T1FLS		FPA-IT2FLS		Z-Test
Function	Mean	Std	Mean	Std	
1 Sphere	7.520E-01	2.830E-01	1.010E+00	2.900E-01	**Y**
2 Ackley	5.890E-01	7.740E-02	5.700E-01	1.090E-01	**N**
3 Rastrigin	7.910E-01	1.060E-01	2.360E+02	6.470E+00	**Y**
4 Zakharov	2.080E+00	6.090E-01	2.190E+00	4.740E-01	**N**
5 Griewank	4.020E-03	8.870E-04	9.690E-03	2.270E-03	**Y**
6 Sum of Dif Powers	7.840E-10	2.170E-09	3.010E-09	8.940E-09	**N**
7 Michalewicz	-1.910E+01	2.050E+00	-1.460E+01	9.380E-01	**Y**
8 Rosenbrock	1.680E+02	2.090E+01	1.210E+03	5.390E+01	**Y**

Table 11. Comparison FPA-T1FLS with optimized FPA-IT2FLS

COMPARATIVE 30 Dim	FPA-T1FLS		FPA-IT2330		Z-Test
Function	Mean	Std	Mean	Std	
1 Sphere	4.350E-03	5.000E-03	7.968E-03	9.250E-03	**Y**
2 Ackley	7.620E-01	9.830E-01	8.219E-02	3.312E-02	**Y**
3 Rastrigin	9.800E+00	2.530E+01	3.434E+01	2.194E+00	**Y**
4 Zakharov	3.890E-02	3.670E-02	5.799E-02	3.535E-02	**Y**
5 Griewank	1.410E-04	1.320E-04	3.038E-04	2.270E-04	**Y**
6 Sum of Dif Powers	1.120E-12	4.550E-12	3.116E-10	1.491E-09	**N**
7 Michalewicz	-1.180E+01	8.360E-01	-9.932E+00	7.098E-01	**Y**
8 Rosenbrock	3.230E+01	4.370E+00	1.077E+02	1.099E+01	**Y**

Table 12. Comparison FPA-T1FLS with optimized FPA-IT2FLS

COMPARATIVE 100 Dim	FPA-T1FLS		FPA-IT2330		Z-Test
Function	Mean	Std	Mean	Std	
1.Sphere	7.520E-01	2.830E-01	9.833E-01	2.426E-01	**Y**
2.Ackley	5.890E-01	7.740E-02	7.034E-01	1.275E-01	**Y**
3.Rastrigin	7.910E-01	1.060E-01	2.476E+02	8.315E+00	**Y**
4.Zakharov	2.080E+00	6.090E-01	2.106E+00	5.325E-01	**N**
5.Griewank	4.020E-03	8.870E-04	1.362E-02	3.384E-03	**Y**
6.Sum of Dif Powers	7.840E-10	2.170E-09	2.653E-08	8.098E-08	**Y**
7.Michalewicz	-1.910E+01	2.050E+00	-1.474E+01	1.008E+00	**Y**
8.Rosenbrock	1.680E+02	2.090E+01	1.549E+03	6.847E+01	**Y**

dimensions, hypothesis tests were performed with FPA-T1FLS and optimized FPA-IT2FLS (FPA-IT2B130), it can be observed that only in 7 mathematical functions there was significant evidence that the FPA-IT2FLS (FPA-IT2B130) is better than the FPA-T1FLS.

9 Conclusions and Further Research

We have seen in other research that when we use a Type-1 Fuzzy Logic System (T1FLS) in parameter optimization of a bio-inspired algorithm,

Table 13. Comparison FPA-T1FLS with optimized FPA-IT2FLS

COMPARATIVE 30 Dim	FPA-T1FLS		FPA-IT2B130		Z-Test
Function	Mean	Std	Mean	Std	
1 Sphere	4.350E-03	5.000E-03	6.731E-03	5.871E-03	**Y**
2 Ackley	7.620E-01	9.830E-01	8.226E-02	3.306E-02	**Y**
3 Rastrigin	9.800E+00	2.530E+01	3.478E+01	2.408E+00	**Y**
4 Zakharov	3.890E-02	3.670E-02	5.126E-02	3.528E-02	**N**
5 Griewank	1.410E-04	1.320E-04	2.843E-04	2.096E-04	**Y**
6 Sum of Dif Powers	1.120E-12	4.550E-12	1.915E-11	8.446E-11	**N**
7 Michalewicz	-1.180E+01	8.360E-01	-9.916E+00	4.734E-01	**Y**
8 Rosenbrock	3.230E+01	4.370E+00	1.132E+02	1.028E+01	**Y**

Table 14. Comparison FPA-T1FLS with optimized FPA-IT2FLS

COMPARATIVE 100 Dim	FPA-T1FLS		FPA-IT2B130		Z-Test
Function	Mean	Std	Mean	Std	
1.Sphere	7.520E-01	2.830E-01	9.797E+00	4.478E+00	**Y**
2.Ackley	5.890E-01	7.740E-02	6.573E-01	9.349E-02	**Y**
3.Rastrigin	7.910E-01	1.060E-01	2.521E+02	7.748E+00	**Y**
4.Zakharov	2.080E+00	6.090E-01	9.939E+01	4.088E+00	**Y**
5.Griewank	4.020E-03	8.870E-04	1.283E-02	3.657E-03	**Y**
6.Sum of Dif Powers	7.840E-10	2.170E-09	1.470E-09	6.361E-09	**N**
7.Michalewicz	-1.910E+01	2.050E+00	-1.489E+01	1.142E+00	**Y**
8.Rosenbrock	1.680E+02	2.090E+01	1.638E+03	8.748E+01	**Y**

good results are obtained, but when we use a Type-2 Fuzzy Logic System (T2FLS) for parameter optimization, better results are obtained.

In this research, we used the bio-inspired algorithm FPA and an Interval Type-2 Fuzzy Logic System (IT2FLS). The experiments were performed with 8 benchmark functions: Sphere, Ackley, Rastrigin, Zakharov, Griewank, Sum of different powers, Michalewicz and Rosenbrock for 30 and 100 dimensions.

Once the hypothesis tests are done, we can observe that the methods that use interval type-2 fuzzy systems are better than type-1 fuzzy systems and even better results are obtained when interval type-2 fuzzy systems are optimized, in this research the FPA-IT2330 architecture was the best architecture obtained for an interval type-2 fuzzy system, of the 8 membership functions in 7 the IT2FLS was better for 30 and 100 dimensions (Tables 11 and 12).

As future work, we can perform experiments with more mathematical functions CEC2013 and CEC2017, we can also perform experiments with other dimensions: 5, 10, 50, 50, 200 and 500 with these last two surely the computational cost will be high, we can also perform experiments with other architectures of interval type-2 fuzzy systems and finally we can perform experiments with generalized type-2 fuzzy systems.

References

1 **Castillo, O., Melin, P., Ontiveros, E., Peraza, C., Ochoa, P., Valdez, F., Soria, J. (2019).** A high-speed interval type 2 fuzzy system approach for dynamic parameter adaptation in metaheuristics. Engineering Applications of Artificial Intelligence, Vol. 85, pp. 666–680. DOI: 10.1016/j.engappai.2019.07.020.

2 **Kirsch, U. (1993).** Structural optimization: fundamentals and applications. Springer-Verlag, pp. 57–124.

3 **Faturechi, R., Miller-Hooks, E. (2014).** A mathematical framework for quantifying and optimizing protective actions for civil infrastructure systems. Computer-Aided Civil and Infrastructure Engineering, Vol. 29, No. 8, pp. 572–589. DOI: 10.1111/mice.12027.

4 **Aldwaik, M., Adeli, H. (2014).** Advances in optimization of highrise building structures. Structural and Multidisciplinary Optimization, Vol. 50, No. 6, pp. 899–919. DOI: 10.1007/s00158-014-1148-1.

5 **Gao, H., Zhang, X. (2013).** A Markov-based road maintenance optimization model considering user costs. Computer-Aided Civil and Infrastructure Engineering, Vol. 28, No. 6, pp. 451–464. DOI: 10.1111/mice.12009.

6 **Zhang, G., Wang, Y. (2013).** Optimizing coordinated ramp metering—A preemptive hierarchical control approach. Computer-Aided Civil and Infrastructure Engineering, Vol. 28, No. 1, pp. 22–37. DOI: 10.1111/j.1467-8667.2012.00764.x.

7 **Carreon, H., Valdez, F., Castillo, O. (2020).** Fuzzy Flower Pollination Algorithm to Solve Control Problems. Hybrid Intelligent Systems in Control, Pattern Recognition and Medicine, Studies in Computational Intelligence, Vol. 827, pp. 119–154. DOI: 10.1007/978-3-030-34135-0_10.

8 **Carvajal, O., Castillo, O., Soria, J. (2018).** Optimization of Membership Function Parameters for Fuzzy Controllers of an Autonomous Mobile Robot Using the Flower Pollination Algorithm. Journal of Automation, Mobile Robotics and Intelligent Systems, Vol. 12, No. 1, pp. 44–49. DOI: 10.14313/JAMRIS_1-2018/6.

9 **Sharma, K.R., Honc, D., Dušek, F. (2015).** Predictive control of differential drive mobile robot considering dynamics and kinematics. 30th European Conference on Modelling and Simulation, pp. 354–360. DOI: 10.7148/2016-0354.

10 **Umoh, U.A., Inyang, U.G., Nyoho, E.E. (2019).** Interval Type-2 Fuzzy Logic for Fire Outbreak Detection. International Journal on Soft Computing, Artificial Intelligence and Applications, Vol. 8, No. 3, pp. 27–46. DOI: 10.5121/ijscai.2019.8303.

11 **Rai, D., Tyagi, K. (2013).** Bio-inspired optimization techniques: a critical comparative study. ACM SIGSOFT Software Engineering Notes, Vol. 38, No. 4, pp. 1–7. DOI: 10.1145/2492248.2492271.

12 **Valdez, F. (2015).** Bio-Inspired Optimization Methods. Springer Handbook of Computational Intelligence, pp. 1533–1538. DOI: 10.1007/978-3-662-43505-2_81.

13 **Holland, J.H. (1992).** Adaptation in Natural and Artificial Systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT Press.

14 **Kennedy, J., Eberhart, R. (1995).** Particle swarm optimization. International Conference on Neural Networks (ICNN), Vol. 4, pp. 1942-1948. DOI: 10.1109/ICNN.1995.488968.

15 **Kirkpatrick, S. Gelatt, C.D., Vecchi, M.P. (1983).** Optimization by Simulated Annealing. Science, Vol. 220, No. 4598, pp. 671–680. DOI: 10.1126/science.220.4598.671.

16 **Hooke, R., Jeeves, T.A. (1961).** Direct search solution of numerical and statistical problems. Journal of the ACM (JACM), Vol. 8, No. 2, pp. 212–229. DOI: 10.1145/321062.321069.

17 **Yang, X.S. (2012).** Flower pollination algorithm for global optimization. 11th International Conference on Unconventional Computation and Natural Computation, Lecture Notes in Computer Science, Vol. 7445, pp. 240–249. DOI: 10.1007/978-3-642-32894-7_27.

18 **Sabahi, F., Akbarzadeh-T, M.R. (2013).** A qualified description of extended fuzzy logic. Information Sciences, Vol. 244, pp. 60–74. DOI: 10.1016/j.ins.2013.03.020.

19 **Zadeh, L.A. (1965).** Fuzzy sets. Information and Control, Vol. 8, pp. 338–353.

20 **Nikravesh, M. (2007).** Evolution of fuzzy logic: From intelligent systems and computation to human mind. Studies in Fuzziness and Soft Computing, Vol. 217, pp. 37–54.

21 **Black, M. (1937).** Vagueness, an exercise in logical analysis. Philosophy of Science, Vol 4, No. 4, pp. 427–455.

22 **Russell, B. (1923).** Vagueness. The Australian Journal of Psychology and Philosophy, Vol. 1, No. 2, pp. 84–92. DOI: 10.1080/00048402308540623.

23 **Brock, J. (1979).** Principle themes in Peirce's logic of vagueness. Peirce Studies, Vol. 1, No. 1, pp. 41–50.

24 **Nadin, M. (1982).** Consistency, completeness and the meaning of sign theories: The Semiotic Field. The American Journal of Semiotics, Vol. I, No. 3, pp. 79–98. DOI: 10.5840/ajs1982135.

25 **Nadin, M. (1980).** The logic of vagueness and the category of synechism. The Monist, Library of Philosophy, Vol. 63, No. 3, pp. 351–366.

26 **Engel-Tiercelin, C. (1992).** Vagueness and the unity of C.S. Peirce's Realism. Transactions of the Charles S. Peirce Society, Vol. 28, No. 1, pp. 51–82.

27 **Merrell, F. (1995).** Semiosis in the Postmodern Age. Purdue University Press.

28 **Merrell, F. (1996).** Signs Grow: Semiosis and Life Processes. University of Toronto Press.

29 **Kaveh, A. (2017).** Applications of Metaheuristic Optimization Algorithms in Civil Engineering. Springer, Cham.

30 **Surjanovic, S., Bingham, D. (2013).** Virtual Library of Simulation Experiments: Test Functions and Datasets. Simon Fraser University.

31 **Reznik, L. (1997).** Fuzzy Controllers. Victoria University of Technology.

32 **Mendel, J.M., John, R.I. (2002).** Type-2 fuzzy sets made simple. IEEE Transactions on Fuzzy Systems, Vol. 10, No. 2, pp. 117–127. DOI: 10.1109/91.995115.

33 **Mendel, J.M., Hagras, H., John, R.I. (2006).** Standard background material about interval type-2 fuzzy logic systems that can be used by all authors. IEEE Computational Intelligence Society.

34 **Hagras, H.A. (2004).** A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. IEEE Transactions on Fuzzy Systems, Vol. 12, No. 4, pp. 524-539. DOI: 10.1109/TFUZZ.2004.832538.

35 **Liang, Q., Mendel, J.M. (2000).** Interval type-2 fuzzy logic systems: theory and design. IEEE Transactions on Fuzzy Systems, Vol. 8, No. 5, pp. 535-550. DOI: 10.1109/91.873577.

36 **Khursheed, M., Nadeem, M.F., Khalil, A., Sajjad, I.A., Raza, A., Iqbal, M.Q., Bo, R., Rehman, W.U. (2020).** Review of Flower Pollination Algorithm: Applications and Variants. International Conference on Engineering and Emerging Technologies (ICEET), pp. 1–6. DOI: 10.1109/ICEET48479.2020.9048215.

37 **Madasu, S.D., Kumar, M.L.S.S., Singh, A.K. (2018).** A flower pollination algorithm based automatic generation control of interconnected power system. Ain Shams Engineering Journal, Vol. 9, No. 4, pp. 1215–1224. DOI: 10.1016/j.asej.2016.06.003.

38 **Kaur, G., Singh, D. (2012).** Pollination Based Optimization or Color Image Segmentation. International Journal of Computer Engineering and Technology (IJCET), Vol. 3, No. 2, pp. 407–414.

39 **Kumar, S., Singh, A. (2012).** Pollination based optimization. 6th International Multi Conference on Intelligent Systems, Sustainable, New and Renewable Energy Technology and Nanotechnology (IISN), pp. 269–273.

40 **Waser, N.M. (1986).** Flower constancy: definition, cause and measurement. The American Naturalist, Vol. 127, No. 5, pp. 593–603. DOI: 10.1086/284507.

41 **Abdel-Raouf, O., Abdel-Baset, Mohamed, El-Henawy, I. (2014).** A Novel Hybrid Flower Pollination Algorithm with Chaotic Harmony Search for Solving Sudoku Puzzles. International Journal of Engineering Trends and Technology (IJETT), Vol. 7, No. 3, pp. 126–132. DOI: 10.14445/22315381/IJETT-V7P225.

42 **Kalra, S., Arora, S. (2016).** Firefly algorithm hybridized with flower pollination algorithm for multimodal functions. Proceedings of the International Congress on Information and Communication Technology, Advances in Intelligent Systems and Computing (AISC), Vol. 438, pp. 207–219. DOI: 10.1007/978-981-10-0767-5_23.

43 Pavlyukevich, I. (2007). Lévy flights, non-local search and simulated annealing. Journal of Computational Physics. Vol. 226, No. 2, pp. 1830-1844. DOI: 10.1016/j.jcp.2007.06.008.

44 Bell, A.D., Bryan, A. (2008). Plant form: an illustrated guide to flowering plant morphology. Timber Press.

45 Glover, B. (2007). Understanding flowers and flowering: An integrated approach. Oxford University Press.

46 Pavlyukevich. (2007). Lévy flights, non-local search and simulated annealing. Journal of Computational Physics, Vol. 226, No.2, pp. 1830–1844. DOI: 10.1016/j.jcp.2007.06.008.

47 Dinkar, S.K., Deep, K. (2018). An efficient opposition based Lévy Flight Antlion optimizer for optimization problems. Journal of Computational Science, Vol. 29, pp. 119–141. No. 10.1016/j.jocs.2018.10.002.

48 Castillo, O., Aguilar, L.T. (2019). Fuzzy Lyapunov Synthesis for Nonsmooth Mechanical Systems. Type-2 Fuzzy Logic in Control of Nonsmooth Systems, Studies in Fuzziness and Soft Computing, Vol. 373, pp. 43-54. DOI: 10.1007/978-3-030-03134-3_3.

49 Sadat Asl, A.A., Zarandi, M.H.F. (2018). A Type-2 Fuzzy Expert System for Diagnosis of Leukemia. Fuzzy Logic in Intelligent System Design, NAFIPS, Advances in Intelligent Systems and Computing, Vol. 648, pp. 52–60. DOI: 10.1007/978-3-319-67137-6_6.

50 Umoh, U.A., Inyang U.G., Nyoho E.E. (2020). A Hybrid Framework for Fire Outbreak Detection Based on Interval Type-2 Fuzzy Logic and Flower Pollination Algorithm. Soft Computing for Problem Solving 2019, Advances in Intelligent Systems and Computing, Vol. 1139, pp. 27–145. DOI: 10.1007/978-981-15-3287-0_3.

51 Fouad, A., Gao, X.Z. (2019). A novel modified flower pollination algorithm for global optimization. Neural Computing & Applications, Vol. 31, No. 8, pp. 3875–3908. DOI: 10.1007/s00521-017-3313-0.

52 Kayabekir, A.E., Bekdaş, G., Nigdeli, S.M., Yang, X.S. (2018). A Comprehensive Review of the Flower Pollination Algorithm for Solving Engineering Problems. Nature-Inspired Algorithms and Applied Optimization, Studies in Computational Intelligence, Vol. 744, pp. 171–188. DOI: 10.1007/978-3-319-67669-2_8.

53 Okwu, M.O., Tartibu, L.K. (2021). Metaheuristic Optimization. Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications, Studies in Computational Intelligent, Vol. 927, pp. 1–4.

54 Ontiveros-Robles, E., Melin, P., Castillo, O. (2021). An Efficient High-Order α-Plane Aggregation in General Type-2 Fuzzy Systems Using Newton–Cotes Rules. International Journal of Fuzzy Systems, Vol. 23, No. 4, pp. 1102–1121. DOI: 10.1007/s40815-020-01031-4.

55 Olivas, F., Valdez, F., Melin, P., Sombra, A., Castillo, O. (2019). Interval type-2 fuzzy logic for dynamic parameter adaptation in a modified gravitational search algorithm. Information Sciences, Vol. 476, pp. 159–175. DOI: 10.1016/j.ins.2018.10.025.

56 Ontiveros, E., Melin, P., Castillo, O. (2018). High order α-planes integration: A new approach to computational cost reduction of General Type-2 Fuzzy Systems. Engineering Applications of Artificial Intelligence, Vol. 74, pp. 186–197. DOI: 10.1016/j.engappai. 2018.06.013.

57 Liang, Q., Mendel, J.M. (2000). Interval type-2 fuzzy logic systems: theory and design. *IEEE Transactions on Fuzzy Systems*, Vol. 8, No. 5, pp. 535-550. DOI: 10.1109/91.873577.

58 Valdez, F., Castillo, O., Melin, P. (2021). Bio-Inspired Algorithms and Its Applications for Optimization in Fuzzy Clustering. Algorithms, Vol. 14, No. 4, pp. 1–21. DOI: 10.3390/a14040122.

59 Dubey, H.M., Pandit, M., Panigrahi, B.K. (2015). A biologically inspired modified flower pollination algorithm for solving economic dispatch problems in modern power systems. Cognitive Computation, Vol. 7, No. 5, pp. 594–608. DOI: 10.1007/s12559-015-9324-1.

60 Yamany, W., Zawbaa, H.M., Emary, E., Hassanien, A.E. (2015). Attribute reduction approach based on modified flower pollination algorithm. IEEE International Conference on

Fuzzy Systems (FUZZ-IEEE), pp. 1–7. DOI: 10.1109/FUZZ-IEEE.2015.7338111.

61 Zhou, Y., Wang, R., Luo, Q. (2016). Elite opposition-based flower pollination algorithm. Neurocomputing, Vol. 188, pp. 294–310. DOI: 10.1016/j.neucom.2015.01.110.

62 Sarjiya, Putra, P.H., Saputra, T.A. (2015). Modified flower pollination algorithm for nonsmooth and multiple fuel options economic dispatch. 8th International Conference on Information Technology and Electrical Engineering (ICITEE), pp. 1–5. DOI: 10.1109/ICITEED.2016.7863285.

63 Regalado, J.A., Emilio, B.E., Cuevas, E. (2015). Optimal power flow solution using modified flower pollination algorithm. IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), pp. 1–6. DOI: 10.1109/ROPEC.2015.7395073.

64 Dubey, H.M., Pandit, M., Panigrahi, B.K. (2015). Hybrid flower pollination algorithm with time-varying fuzzy selection mechanism for wind integrated multi-objective dynamic economic dispatch. Renewable Energy, Vol. 83, pp. 188–202. DOI: 10.1016/j.renene.2015.04.034.

65 Zainudin, A., Sia, C.K., Ong, P., Narong, O.L.C., Nor, N.H.M. (2017). Taguchi design and flower pollination algorithm application to optimize the shrinkage of triaxial porcelain containing palm oil fuel ash. IOP Conference Series: Materials Science and Engineering, Vol. 165. DOI: 10.1088/1757-899X/165/1/012036.

66 Abdel-Baset, M., Hezam, I. (2016). A hybrid flower pollination algorithm for engineering optimization problems. International Journal of Computer Applications, Vol. 140, No. 12, pp. 10–23. DOI: 10.5120/ijca2016909119.

67 Lenin, K., Ravindhranath, R.B., Surya, K.M. (2014). Shrinkage of active power loss by hybridization of flower pollination algorithm with chaotic harmony search algorithm. Control Theory and Informatics, Vol. 4, No. 8, pp. 31–38.

68 Jensi, R., Jiji, G.W. (2015). Hybrid data clustering approach using K-means and flower pollination algorithm. Advanced Computational Intelligence: An International Journal (ACII), Vol. 2, No. 2, pp. 15–25. DOI: 10.48550/arXiv.1505.03236.

69 Merzougui, A., Labed, N., Hasseine, A., Bonilla-Petriciolet, A., Laiadi, D., Bacha, O. (2016). Parameter identification in liquid-liquid equilibrium modeling of food-related thermodynamic systems using flower pollination algorithms. The Open Chemical Engineering Journal Vol. 10, No. 1, pp. 59–73. DOI: 10.2174/1874123101610010059.

70 Shehata, M.N., Fateen, S.E.K., Bonilla-Petriciolet, A. (2016). Critical point calculations of multi-component reservoir fluids using nature-inspired metaheuristic algorithms. Fluid Phase Equilibria, Vol. 409, pp. 280–290. DOI: 10.1016/j.fluid.2015.10.002.

71 Zainudin, A., Sia, C.K., Ong, P., Narong, O.L.C., Nor, N.H.M. (2017). Taguchi design and flower pollination algorithm application to optimize the shrinkage of triaxial porcelain containing palm oil fuel ash. International Conference on Applied Science (ICAS), IOP Conference Series: Materials Science and Engineering, Vol. 165. DOI: 10.1088/1757-899X/165/1/012036.

72 Narong, L.C., Sia, C.K., Yee, S.K., Ong, P., Zainudin, A., Nor, N.H.M., Kasim, N.A. (2017). Optimization of the EMI shielding effectiveness of fine and ultrafine POFA powder mix with OPC powder using flower pollination algorithm. International Conference on Applied Science (ICAS), IOP Conference Series: Materials Science and Engineering, Vol. 165. DOI: DOI: 10.1088/1757-899X/165/1/012035.

73 Nigdeli, S.M., Bekdaş, G., Yang, X.S. (2016). Application of the flower pollination algorithm in structural engineering. **Yang, X.S., Bekdaş G., Nigdeli S.M. (eds.)**, Metaheuristics and Optimization in Civil Engineering, Modeling and Optimization in Science and Technologies, Vol. 7, pp. 25–42. DOI: 10.1007/978-3-319-26245-1_2.

74 Meng, O.K., Pauline, O., Kiong, S.C., Wahab, H.A., Jafferi, N. (2017). Application of modified flower pollination algorithm on mechanical engineering design problem.

International Conference on Applied Science (ICAS), IOP Conference Series: Materials Science and Engineering, Vol. 165. DOI:10.1088/1757-899X/165/1/012032.

75 **Bekdaş, G., Nigdeli, S.M., Yang, X.S. (2015).** Sizing optimization of truss structures using flower pollination algorithm. Applied Soft Computing, Vol. 37, pp. 322–331. DOI: 10.1016/j.asoc.2015.08.037.

76 **Kavirayani, S., Kumar, G.V. (2017).** Flower pollination for rotary inverted pendulum stabilization with delay. Telecommunication Computing Electronics and Control (TELKOMNIKA), Vol. 15, No. 1, pp. 245–253. DOI: 10.12928/TELKOMNIKA.v15i1.3403.

77 **Xu, S., Wang, Y., Huang, F. (2017).** Optimization of multi-pass turning parameters through an improved flower pollination algorithm. International Journal of Advanced Manufacturing Technology, Vol. 89, No. 1, pp. 503–514. DOI: 10.1007/s00170-016-9112-4.

78 **Acherjee, B., Maity, D., Kuar, A.S. (2017).** Parameters optimisation of transmission laser welding of dissimilar plastics using RSM and flower pollination algorithm integrated approach. International Journal of Mathematical Modelling and Numerical Optimisation, Vol. 8, No. 1, pp. 1–22. DOI: 10.1504/IJMMNO.2017.083656.

79 **Chakravarthy, V., Rao, P.M. (2015).** On the convergence characteristics of flower pollination algorithm for circular array synthesis. 2nd International Conference on Electronics and Communication Systems (ICECS), pp. 485–489. DOI: 10.1109/ECS.2015.7124953.

80 **Chakravarthy, V., Paladuga, S.R., Prithvi, M.R. (2015).** Synthesis of Circular Array Antenna for Sidelobe Level and Aperture Size Control Using Flower Pollination Algorithm. International Journal of Antennas and Propagation, Vol. 2015, pp. 1–9. DOI: 10.1155/2015/819712.

81 **Singh, U., Salgotra, R. (2017).** Pattern Synthesis of Linear Antenna Arrays Using Enhanced Flower Pollination Algorithm. International Journal of Antennas and Propagation, Vol. 2017, pp. 1–11. DOI: 10.1155/2017/7158752.

82 **Prathiba, R., Moses, M.B., Sakthivel, S. (2014).** Flower pollination algorithm applied for different economic load dispatch problems. International Journal of Engineering and Technology (IJET), Vol. 6, No. 2, pp. 1009–1016.

83 **Kaur, G., Singh, D., Kaur, M. (2013).** Robust and efficient 'RGB' based fractal image compression: flower pollination-based optimization. International Journal of Computer Applications, Vol. 78, No. 10, pp. 11–15. DOI: 10.5120/13524-1215.

84 **Ouadfel, S., Taleb-Ahmed, A. (2016).** Social spiders optimization and flower pollination algorithm for multilevel image thresholding: a performance study. Expert Systems with Applications, Vol. 55, pp. 566–584. DOI: 10.1016/j.eswa.2016.02.024.

85 **Rodrigues, D., Silva, G.F.A., Papa, J.P., Marana, A.N., Yang, X.S. (2016).** EEG-based person identification through binary flower pollination algorithm. Expert Systems with Applications, Vol. 62, pp. 81–90. DOI: 10.1016/j.eswa.2016.06.006.

86 **Alyasseri, Z.A.A., Khader A.T., Al-Betar M.A., Awadallah, M.A., Yang, X.S. (2018).** Variants of the Flower Pollination Algorithm: A Review. **Yang, X.S., eds.**, Nature-Inspired Algorithms and Applied Optimization, Studies in Computational Intelligence, Vol. 744, pp. 91–118. DOI: 10.1007/978-3-319-67669-2_5.

87 **Balasubramani, K., Marcus, K. (2014).** A Study on Flower Pollination Algorithm and Its Applications. International Journal of Application or Innovation in Engineering & Management, Vol. 3, No. 11, pp 230–235. DOI: 10.2648/IJAIEM.303.609.

88 **Chiroma, H., Shuib, N.L.M., Muaz, S.A., Abubakar, A.I., Ila, L.B., Maitama, J.Z. (2015).** A Review of the Applications of Bio-Inspired Flower Pollination Algorithm. Procedia Computer Science, Vol. 62, pp. 435–441. DOI: 10.1016/j.procs.2015.08.438.

89 **Himanshukumar, R.P., Vipul, A.S. (2020).** Comparative Study of Interval Type-2 and Type-1 Fuzzy Genetic and Flower Pollination Algorithms in Optimization of Fuzzy Fractional Order PIλD μ Controllers. **Yang, Yi,** editor,

Intelligent System and Computing, IntechOpen. DOI: 10.5772/intechopen.90359.

90 Umoh, U., Abayomi, A., Udoh, S., Abdulazeez, A. (2021). Flower Pollination Algorithm in Optimization of Interval Type-2 Fuzzy for Telemedical Problem. **Abraham, A., Sasaki, H., Rios, R., Gandhi, N., Singh, U., Ma, K. (eds.)**, Innovations in Bio-Inspired Computing and Applications (IBICA), Advances in Intelligent Systems and Computing, Vol. 1372, pp. 43–54. DOI: 10.1007/978-3-030-73603-3_4.

91 Zadeh, L.A. (1975). The concept of a linguistic variable and its application to approximate reasoning—I. Information Sciences, Vol. 8, No. 3, pp. 199–249. DOI: 10.1016/0020-0255(75)90036-5.

92 Olivas, F., Valdez, F., Castillo, O., Melin, P. (2016). Dynamic parameter adaptation in particle swarm optimization using interval type-2 fuzzy logic. Soft Computing, Vol. 20, No. 3, pp. 1057–1070. DOI: 10.1007/s00500-014-1567-3.

93 Roeva, O., Zoteva, D., Castillo, O. (2021). Joint set-up of parameters in genetic algorithms and the artificial bee colony algorithm: an approach for cultivation process modelling. Soft Computing, Vol. 25, pp. 2015–2038. DOI: 10.1007/s00500-020-05272-1.

94 Lagunes, M. L., Castillo, O., Soria, J., Valdez, F. (2021). Optimization of a fuzzy controller for autonomous robot navigation using a new competitive multi-metaheuristic model. Soft Computing, Vol. 25, pp. 11653–11672. DOI: 10.1007/s00500-021-06036-1.

95 Hidalgo, D., Cervantes, L., Castillo, O., Melin, P., Martinez-Soto, R. (2020). Fuzzy Parameter Adaptation in Genetic Algorithms for the Optimization of Fuzzy Integrators in Modular Neural Networks for Multimodal Biometry. Computación y Sistemas, Vol. 24, No. 3, pp. 1093–1105. DOI: 10.13053/CyS-24-3-3329.

96 Valdez, F., Castillo, O., Peraza, C. (2020). Fuzzy Logic in Dynamic Parameter Adaptation of Harmony Search Optimization for Benchmark Functions and Fuzzy Controllers. International Journal of Fuzzy Systems, Vol. 22, pp. 1198–1211. DOI: 10.1007/s40815-020-00860-7.

97 Castillo, O., Amador-Angulo, L.A. (2018). A generalized type-2 fuzzy logic approach for dynamic parameter adaptation in bee colony optimization applied to fuzzy controller design. Information Sciences, Vol. 460–461, pp. 476-496. DOI: 10.1016/j.ins.2017.10.032.

98 Ontiveros, E., Melin, P., Castillo, O. (2018). High order α-planes integration: A new approach to computational cost reduction of General Type-2 Fuzzy Systems. Engineering Applications of Artificial Intelligence, Vol, 74, pp. 186–197. DOI: 10.1016/j.engappai. 2018.06.013.

Article received on 08/06/2021; accepted on 17/11/2021.
Corresponding author is Oscar Castillo.

ISSN 2007-9737

An Experimental Study of Grouping Crossover Operators for the Bin Packing Problem

Stephanie Amador-Larrea, Marcela Quiroz-Castellanos,
Guillermo-de-Jesús Hoyos-Rivera, Efrén Mezura-Montes

Universidad Veracruzana,
Instituto de Investigación de Inteligencia Artificial,
Mexico

emezura@gmail.com

Abstract. The one-dimensional Bin Packing Problem (1D-BPP) is a classical NP-hard problem in combinatorial optimization with an extensive number of industrial and logistic applications, considered intractable because it demands a significant amount of resources for its solution. The Grouping Genetic Algorithm with Controlled Gene Transmission (GGA-CGT) is one of the best state-of-the-art algorithms for 1D-BPP. This article aims to highlight the impact that the crossover operator itself can have on the final performance of the GGA-CGT. We present a comparative experimental study of four state-of-the-art crossover operators for 1D-BPP: Uniform, Exon Shuffling, Greedy Partition and Gene-level; this is the first time that the Uniform, Exon Shuffling and Greedy Partition operators are adapted and studied as a part of the GGA-CGT; moreover, the Uniform crossover has never been used before for solving the 1D-BPP. We measure the performance of the GGA-CGT by replacing its original crossover operator (Gene-level) with each of the other three state-of-the-art operators. Furthermore, we propose a new version of the Uniform crossover and examine two replacement strategies for the Gene-level crossover. Experimental results indicate that the Gene-level crossover operator is shown to have a greater impact in terms of the number of optimal solutions found, outperforming the other operators for the class of Hard28 instances, which has shown the greatest degree of difficulty for 1D-BPP algorithms.

Keywords. Bin packing problem, group oriented crossover operators, evolutionary computation, grouping genetic algorithm.

1 Introduction

The off-line one-dimensional Bin Packing Problem (1D-BPP) is a well-known grouping optimization problem with many applications in logistics, industry, telecommunications, transports, among several others. Given an unlimited number of bins with a fixed capacity $c > 0$ each, and a set of n items, each one with a specific weight $0 < w_i \leq c$, 1D-BPP consists of storing all of the items into the minimum number of bins without exceeding the capacity of any bin. 1D-BPP belongs to the NP-hard class, i.e., the problem complexity grows exponentially as the problem size increases.

It implies that there is no efficient algorithm to find an optimal solution for every instance of 1D-BPP. Searching for the best possible solutions to 1D-BPP, a wide variety of algorithms have been designed. The proposals range from simple heuristics to hybrid strategies, including branch and bound techniques [7], metaheuristics [20] and special neighbourhood searches [4].

However, despite the efforts of the scientific community to develop new strategies, there is not yet an efficient algorithm capable of finding the best solution for all possible 1D-BPP instances, so it is then important to try to identify the characteristics that define the behavior of the algorithms to understand and improve their performance.

One of the suggested methods to solve BBP is the Grouping Genetic Algorithm (GGA) proposed by Falkenauer in 1996 [10], who presented

a design of three new components: (1) a representation scheme for solutions in which groups are seen as genes; (2) a fitness function that evaluates the exploitation of bins' capacity; and, (3) grouping variation operators to modify and re-combine the group-based solutions, including a Segment-level crossover. Later, in 2015, Quiroz-Castellanos et al. [20] proposed the algorithm known as Grouping Genetic Algorithm with Controlled Gene Transmission (GGA-CGT).

Unlike Falkenauer, Quiroz-Castellanos et al. proposed the application of the variation operators in a controlled way, inducting the fullest-bin pattern. GGA-CGT is one of the best algorithms found in the state-of-the-art to solve 1D-BPP; it focuses on the transmission of the best genes on the chromosomes (the fullest-bin pattern), keeping a balance between selective pressure and diversity in the population, in order to favor the generation and evolution of high-quality solutions. GGA-CGT includes an intelligent grouping crossover operator, called Gene-level crossover, which gives the best genes (the fullest bins) a higher probability of being preserved.

The experimental results presented by Quiroz-Castellanos et al. [20] exposed that the Gene-level crossover showed an effectiveness improvement of 30% when compared with the Segment-level crossover proposed by Falkenauer [10]. Despite the success of the Gene-level crossover, the performance of the GGA-CGT is related mainly to the mutation operator, which alone is capable of finding quality solutions.

In the present work, three well-known grouping crossover operators are implemented to solve 1D-BPP: Uniform crossover, Exon Shuffling crossover and Greedy Partition crossover. These operators have never been implemented within the GGA-CGT algorithm before. Furthermore, the Uniform crossover has not been used to solve the 1D-BPP. The goal of the implementation is to measure the performance of these crossover operators whit respect to the predefined Gene-level crossover operator. The performance of the GGA-CGT is studied by replacing the original Gene-level crossover operator with each of these state-of-the-art crossovers as well as new versions of the Gene-level and the Uniform crossover operators.

The paper structure is as follows. Section 2 presents the most relevant state-of-the-art algorithms for 1D-BPP; Section 3 comprises a brief definition of the components of the GGA-CGT; Section 4 includes an explanation for the state-of-the-art grouping crossover operators that will be implemented afterwards; Section 5 contains the experimental proposal to analyze the performance of the mentioned grouping crossover operators; finally, Section 6 summarizes the conclusions and future research paths.

2 Releated Work

In the last three decades, different techniques have been implemented in order to find the best solution for the BPP, as it is one of the most interesting problems of the optimization field. Among the techniques that stand out the most are hybrid algorithms and heuristics. For the 1D-BPP study, most algorithms proposed in the literature have been evaluated using a well-studied trial benchmark [8]; it includes 1615 instances in which the number of items n varies within $[50, 1000]$, the bin capacity c is within $[100, 100000]$ and the ranges of the weights are within $(0, c]$.

The specialized literature includes approximation algorithms, which had their performances mathematically analyzed, being the most successful ones: (1) First-Fit Decreasing (FFD), (2) Best-Fit Decreasing (BFD) and (3) Minimum Bin Slack (MBS) [13]. The proposals also include many exact algorithms using dynamic programming, LP relaxation, branch-and-bound, branch-and-price and constraint programming methods [7, 17]. The most relevant results have been obtained by means of metaheuristic and hybrid algorithms covering proposals based on local search [2, 4], evolutionary algorithms [3, 10, 20, 15, 5] and swarm intelligence algorithms [1, 16, 18, 11]. The most exploited approaches, which have allowed obtaining the best results, consist mainly of: (1) the use of simple 1D-BPP heuristics; (2) the application of search space reduction methods; (3) the inclusion of local search techniques based on the dominance criterion; (4) the use of lower

bounding strategies; and (5) the induction of the fullest-bin pattern.

The review of the results obtained by the best 1D-BPP solution algorithms revealed that there are still instances of the literature that present a high degree of difficulty and the strategies included in the procedures do not seem to lead to better solutions. After the literature analysis, it was observed that none of the state-of-the-art strategies had been analyzed to explain the reason for its high-grade or poor performance. Few studies have centered on the analysis of the relationships between the effectiveness of the algorithms and the structure and complexity of the 1D-BPP instances [6]. It is important to understand the algorithms' behavior and evaluate the strategies that allow them to achieve their performance. This work aims at studying the performance of different grouping crossover operators trying to identify the strategies that they use and that positively impact their performance.

3 Grouping Genetic Algorithm with Controlled Gene Transmission (GGA-CGT)

The GGA-CGT proposed in 2015 by Quiroz-Castellanos et al. [20] uses a group-based representation in which each gene represents a group of items or bin. The GGA-CGT is aimed at maximizing the fitness of the individuals in the population. The fitness function is described as follows:

$$F_{BPP} = \frac{\Sigma_{i=1}^{m}(S_i/c)^2}{m}, \tag{1}$$

where m is the number of bins in the solution, S_i is the total weight of the items in the bin i and c corresponds to the capacity of the bins. The GGA-CGT algorithm generates an initial population using the FF-n heuristic, in which the n objects of weight greater than 50 percent of the bin capacity are packed in n separate bins, then the remaining objects are accommodated using the well-known First Fit heuristic on a random permutation of this subset.

GGA-CGT uses a controlled selection regarding the choice of individuals to cross and mutate. The strategy consist of an elitist approach together with two inverted rankings to give all the solutions a chance to contribute to the next generation but forcing the survival of the best solutions. For the crossover, n_c parents are selected to generate n_c children. Two sets of parents G and R are generated each with $n_c/2$ individuals, one set being randomly selected from the best n_c individuals with uniform probability (B) and the other set randomly selected from the whole population without considering the elite solutions with uniform probability (R). For mutation, n_m individuals are taken from the best individuals in the population.

GGA-CGT includes a new crossover operator that is referred to as Gene-level crossover, which generates two children c_1 y c_2 from two parents p_1 y p_2. In this operator, both parents are first sorted in descending order with respect to how full each gene (bin) is. Then, the genes of both parents are compared in parallel, whereby the fuller gene is inherited first. If both genes are equally full, then, for the first child, preference is given to the first parent's gene and, for the second child, preference is given to the second parent's gene. If any parent has more genes than the other, these are inherited directly from this solution. Genes with repeated items are eliminated from the children, and the missed items are reinserted with the FFD heuristic.

Regarding the mutation, it consists of an Elimination operator which works at the gene level, promoting the transmission of the best genes on the chromosome. The Adaptive mutation operator, which considers the bins in descending order of their filling, eliminating the n_b least full bins of the solution and reinserting their items with the Rearrangement by Pairs heuristic. The number of bins n_b to be eliminated from the individual, unlike traditional methods, is calculated in relation to the size of the solution and the number of incomplete bins. The equation is defined below:

$$n_b = [\imath \cdot \epsilon \cdot p_\epsilon], \tag{2}$$

where $\imath$ corresponds to the number of incomplete bins in the solution, ϵ corresponds to the elimination proportion defined by Eq. 3, p_ϵ is the

elimination probability defined by Eq. 4, and k is a parameter that defines the rate of change of ϵ and p_ϵ with respect to $\imath$ $(k > 0)$:

$$\epsilon = \frac{(2 - (\imath/m))}{\imath^{(1/k)}}, \tag{3}$$

$$p_\epsilon = 1 - uniform(0, \frac{1}{\imath^{1/k}})). \tag{4}$$

The replacement strategy preserves the population diversity and the best solutions by replacing duplicated fitness individuals and the worst fitness solutions with new offspring. The controlled replacement method for the crossover consists of introducing the n_c children such that $n_c/2$ replace the individuals in the set of random parents R and the other $n_c/2$ replace the individuals with repeated fitness first, if there are still un-reinserted children and no solutions with repeated fitness, they are added by replacing the solutions with the worst fitness solutions.

When the mutation operator is applied, some of the elite solutions whose age is less than a predefined $life\_span$ parameter are cloned. Every clone can be entered into the population in two ways; first, by replacing solutions with repeated fitness, then if there are no solutions with repeated fitness, they are added by replacing the worst fitness solutions.

The details of the heuristics used to generate the population, the rearrangement heuristics to repair solutions, as well as the remaining mechanisms and the parameter settings can be consulted in the work of Quiroz-Castellanos et al. [20].

In order to identify the impact of the Gene-level crossover operator (GLX) on the GGA-CGT performance, an experimental study was performed by using nine different values for the crossover rate, to vary the number of individuals selected for the crossover process (n_c). Fig. 1 and Fig. 2 present the number of optimal solutions found and the average number of generations with different configurations for the GLX. The figures allow observing how the crossover operator seems to have a low impact on the performance of the GGA-CGT. As it can be seen from figures the inclusion of the GLX operator improves the performance of the GGA-CGT, however an increase in the crossover rate does not seem to contribute to the effectiveness of the algorithm.

The following sections will present a series of studies consisting of: (1) implementing the state-of-the-art crossover operators in the GGA-CGT algorithm; (2) performing experimentation with different crossover percentages within the algorithm; and (3) analyzing the results of the GGA-CGT algorithm with each operator.

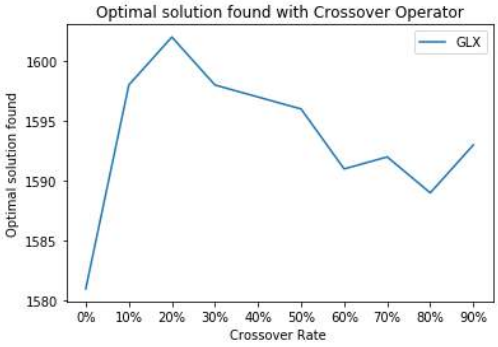


Fig. 1. Number of optimal solutions obtained by the GGA-CGT with the original Gene-level crossover (GLX) for different crossover rates

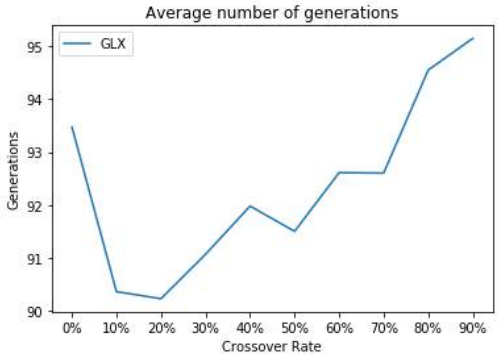


Fig. 2. Average number of generations executed by the GGA-CGT with the original Gene-level crossover (GLX) for different crossover rates

4 Grouping Crossover Operators

The crossover is one the most frequently used genetic operators. This operator combines the

information of two or more solutions (called parents) to produce descendants (called children or offspring). The state-of-the-art algorithms includes different grouping crossover operators (see Ramos-Figueroa et al. [21] for a survey of grouping genetic operators). Grouping crossover operators perform the transmission of the genetic material considering the characteristics of every gene, performing a more controlled combination process, by giving the best genes a higher probability of being preserved.

The state-of-the-art algorithms present four grouping crossover operators that work with the group-based encoding: Gene-level crossover (GLX), Uniform crossover (UX), Exon Shuffling crossover (ESX), and Greedy Partition crossover (GPX). GLX was described in Section 3. The following sections describe the general procedure of the other three operators.

4.1 Uniform Crossover

The Uniform crossover (UX) variation operator generates two children c_1 and c_2 by combining the genes of the parents solutions p_1 and p_2. In this operator the parents are considered in parallel, i.e. gene 1 of $p_1$taken on a par with gene 1 of p_2. For every pair of genes of the parents, a random value with uniform distribution in the interval (0,1) is generated; a value less than or equal to 0.5 indicates that child c_1 and c_2 receive the gene from the parent p_1 and p_2 respectively, otherwise child c_1 receives the gene from p_2 and offspring c_2 receives the gene from p_1. During this process, there is the possibility of creating infeasible solutions, which is why heuristics are used to repair them. The performance of the UX operator has only been tested in a Grouping Genetic Algorithm for the Multiple Knapsack problem in 2008 [12]. In our implementation of UX for 1D-BPP, the genes with repeated items are eliminated from the children, and the missed items are reinserted with the FFD packing heuristic. Figure 3a depicts an example of the crossover process followed by UX.

4.2 Exon Shuffling Crossover

The Exon Shuffling crossover (ESX), was proposed by Kolkman and Stemmer (2001) [14]. It has been often used to tackle 1D-BPP [9, 25, 19]. This is an operator that generates a single child c from two parent solutions p_1 and p_2. The first step consists in joining the parents. Then their genes are ordered from best to worst with respect to their fullness. Finally, the genes are inherited to the child as long as none of the items of the respective gene exist previously in the child [21]. In our implementation of ESX for 1D-BPP, the missed items are reinserted with the FFD packing heuristic. Figure 3b depicts an example of the crossover process followed by ESX.

4.3 Greedy Partition Crossover

The Greedy Partition crossover (GPX), is also among the state-of-the-art grouping oriented crossover operators that have been used to tackle 1D-BPP in other Grouping Genetic Algorithm [23]. This particular operator has two versions and this paper focuses on the group oriented one. Given two parent solutions p_1 and p_2, GPX generates two children c_1 and c_2 using a greedy heuristic. Here the first step is to order the genes of the parent solutions p_1 and p_2 from most to the least filled. Then, for each child, a vector of probabilities with uniform distribution of the size of the parent solution with more genes is generated. The probability defines from which parent the offspring will receive the gene; if the value generated is less than or equal to 0.5, it indicates that the child c_1 receives the gene from p_1; otherwise, it will receive the one from p_2. The same process is employed to create the child c_2. Like in the previous cases, in our implementation of GPX for 1D-BPP, the genes with repeated items are eliminated from the children and the missed items are reinserted with the FFD packing heuristic. Figure 4c depicts an example of the crossover process followed by GPX.

4.4 Gene-Level Crossover

The operator is described in Section 3. This operator was proposed by Quiroz-Castellanos et al. [20] for the GGA-CGT algorithm and has been

used several times to solve the 1D-BPP due to its performance [15, 22, 24]. Figure 4d depicts an example of the crossover process followed by GLX.

Since the above operators, are considered the best grouping crossover operators amongst the state-of-the-art, the next section comprises an analysis of these operators to determine which one enables the GGA-CGT to reach the best performance for 1D-BPP. The experiments cover: (1) the integration of the operators in the GGA-CGT algorithm; (2) the study of different crossover percentages; and, (3) the robustness analysis of the results of the best crossover operators inside the GGA-CGT algorithm.

5 Experimentation and Results

This section presents the experiments to analyze the way the different crossover operators can impact on the performance of GGA-CGT. The experimental design consists of three phases. The first one covers the analysis of the state-of-the-art grouping crossover operators (UX, ESX, GPX and GLX) to determine which ones have the best impact on the effectiveness of GGA-CGT. The second one comprises an analysis of the best operators to observe the influence of the number of children generated and their reinsertion to the population. Finally, the third one studies the robustness of the GGA-CGT with the best crossovers, comparing with the original GLX operator.

The performance assessment of each operator involves solving the 1615 standard instances [8], which are distributed among nine sets: data set 1 (720 instances), data set 2 (480 instances), data set 3 (10 instances), triplets (80 instances), uniform (80 instances), hard28 (28 instances), was 1 (100 instances), was 2 (100 instances), and gau 1 (17 instances). Sets data set 1, data set 3, gau 1 and hard28, have shown to have test cases with a high degree of difficulty. Standing out hard28, where there is a higher number of instances that the algorithms cannot solve optimally.

To analyze the performance of each operator on the GGA-CGT, for each instance, a single execution of the algorithm GGA-CGT was run, with the initial seed for the random number generation set to 1. For each operator ten different crossover rates were explored, from 0% to 90% of the population. For all the parameters, different from the crossover rate, we used the configuration proposed by Quiroz-Castellanos et al. [20].

5.1 State-of-the-Art Grouping Crossover Operators

The GGA-CGT employs the controlled reproduction technique proposed by Quiroz-Castellanos et al. [20], for the state-of-the-art crossover operators we used the same strategy. Both, the UX operator and the GPX operator, generate a set C of n_c children from n_c parents. Like in the original GGA-CGT, the first $n_c/2$ children are introduced to the population replacing the individuals in the set of random parents R. The other $n_c/2$ children are introduced replacing individuals with repeated fitness and replacing the worst solutions. On the other hand, concerning the ESX operator, where $n_c/2$ children are generated from n_c parents (since only one child is generated for every two parents) the reintegration into the population is done in one way: the $n_c/2$ children are introduced replacing the individuals in the set of random parents R.

Table 1, Table 2, Table 3, and Table 4 show the results obtained by the GGA-CGT with each of the state-of-the-art crossover operator (UX, ESX, GPX and GLX, respectively). For every class of instances, each table first shows the number of test cases (Inst.), followed by the number of optimal solutions found by the GGA-CGT with each crossover rate. The last row of each table shows the total number of optimal solutions obtained by each configuration. Moreover, Fig. 5 and Fig. 6 show the number of optimal solutions that are found and the average number of generations executed, when we explore ten different crossover rates (from 0% to 90%) over the four different crossover operators.

The results obtained from implementing the UX variation operator in the GGA-CGT algorithm, are shown in Table 1. As can be seen, the highest number of instances solved is 1592, with a crossover rate of 30%, however, with 90% of crossover rate, the performance of the GGA-CGT is affected since it only solves 1574 of the

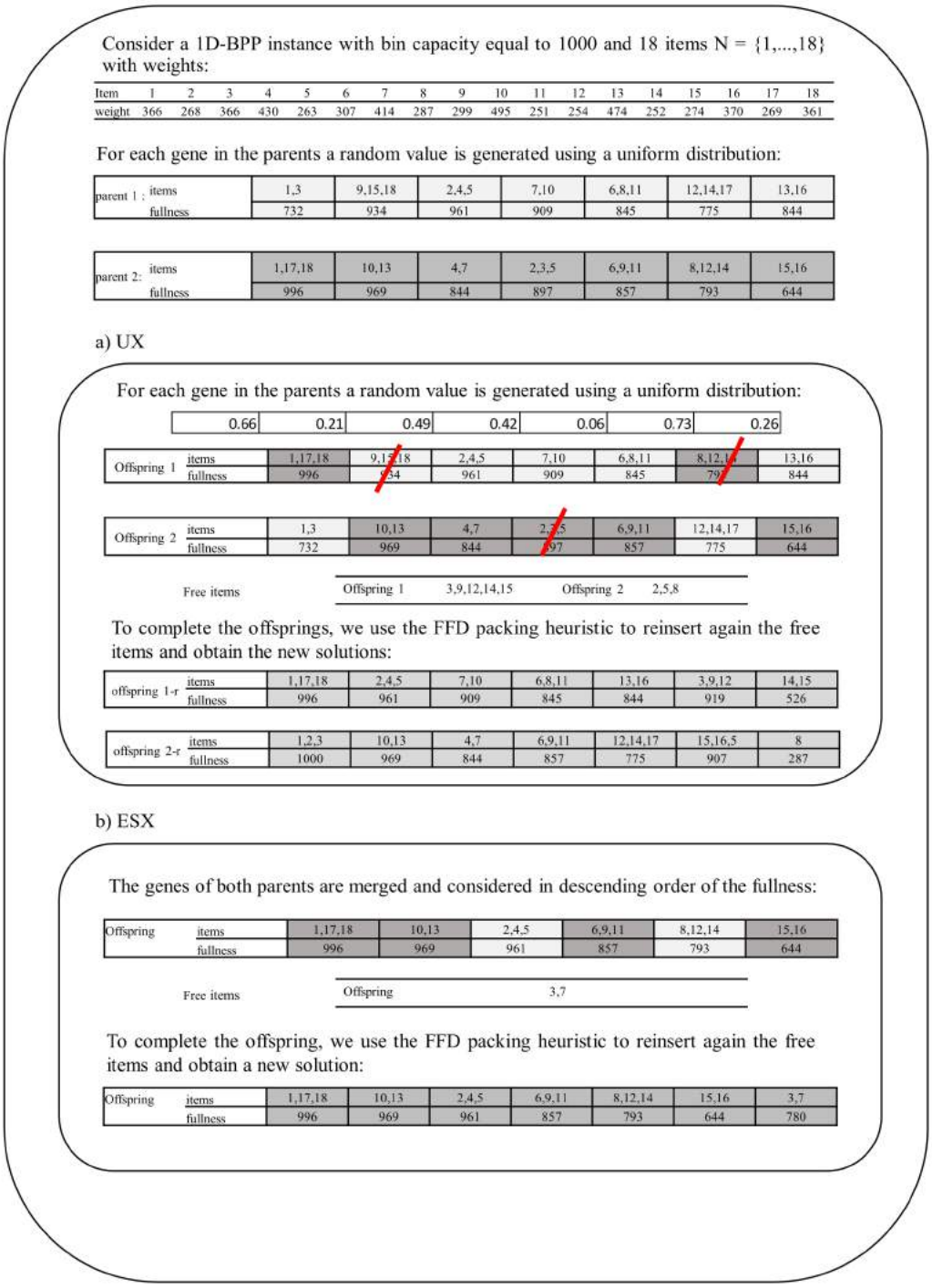


Fig. 3. An example of each state-of-the-art crossover operator for a 1D-BPP instance: UX (Uniform Crossover), ESX (Exon shuffling Crossover), GPX (Greedy partition Crossover), and GLX (Gene-level Crossover) (Part 1)

ISSN 2007-9737

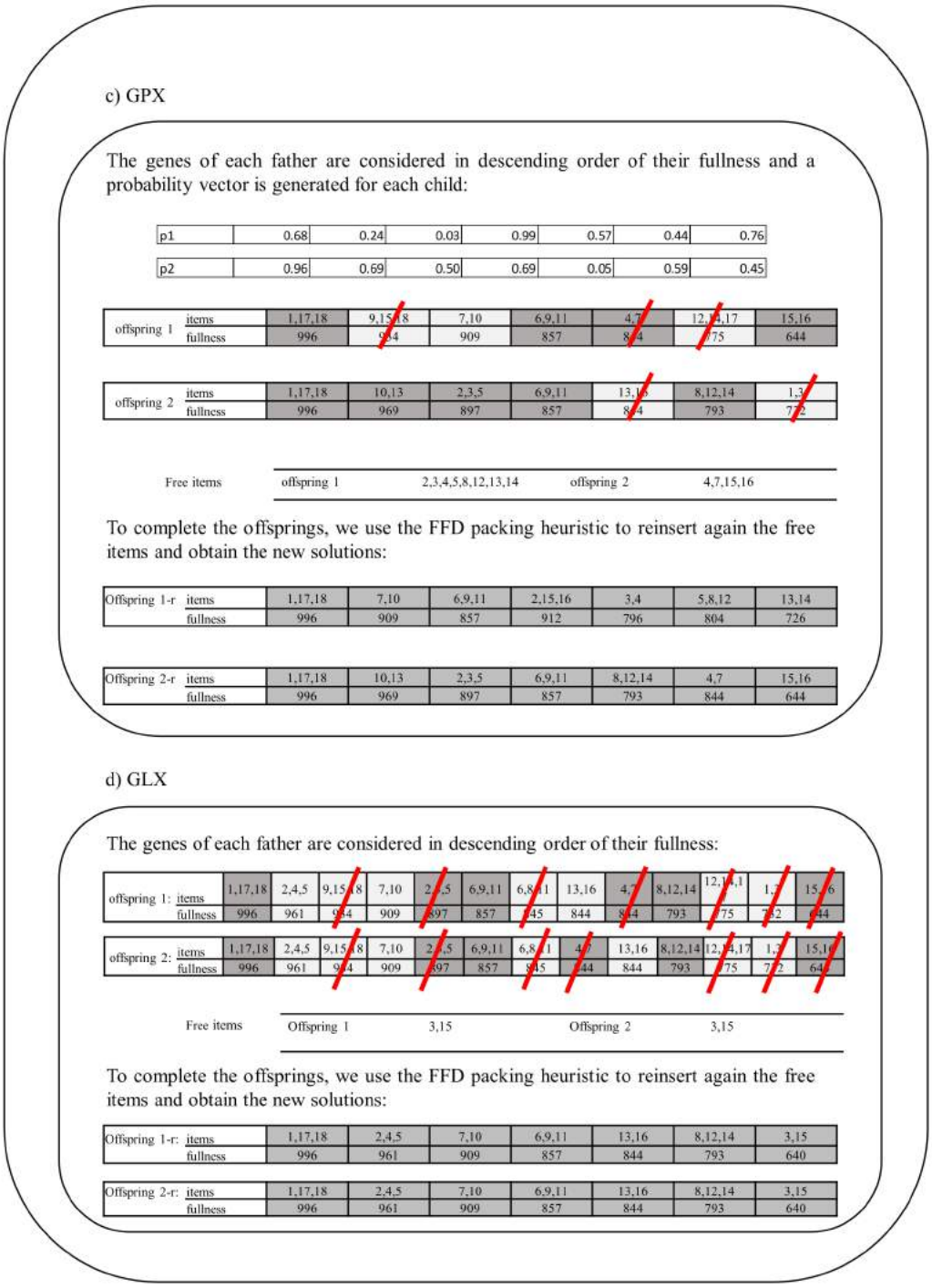


Fig. 4. An example of each state-of-the-art crossover operator for a 1D-BPP instance: UX (Uniform Crossover), ESX (Exon shuffling Crossover), GPX (Greedy partition Crossover), and GLX (Gene-level Crossover) (Part 2)

1615 instances less than those solved with 0% crossover rate. In terms of the average number of generations, with a 90% of crossover rate, the algorithm iterates 100.22 generations per instance, as it is shown in Fig. 6.

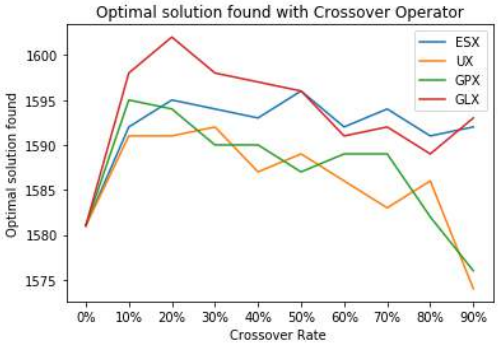


Fig. 5. Number of optimal solutions obtained by the GGA-CGT with each state-of-the-art crossover for different crossover rates

The results obtained from implementing the ESX variation operator in the GGA-CGT algorithm are shown in Table 2. As can be seen, the highest number of the optimally solved instances is 1596, with a 50% crossover rate. Something interesting in this operator is that the optimally solved instances are between 1592 and 1596, and it is not affected with any crossover rate. Concerning to the average number of generations, the algorithm performs a maximum of 91.6 generations, corresponding to a crossover rate of 80%, as it is shown in Fig. 6.

The results obtained from implementing the GPX operator in the GGA-CGT algorithm, are shown in Table 3. The operator solves optimally at most 1595 instances with a 10% crossover rate. As the UX operator, the performance of the GGA-CGT is affected with a 90% crossover rate, since it only finds the optimal solution of 1576 instances less than those solved with 0% crossover rate. It is important to mention that it performs 99.56 generations on average with a 90% crossover rate, as it is shown in Fig. 6.

The same experiment was performed for the crossover operator of the GGA-CGT algorithm, GLX [20]. The results are presented in the Table 4, which with a 20% crossover rate solves optimally

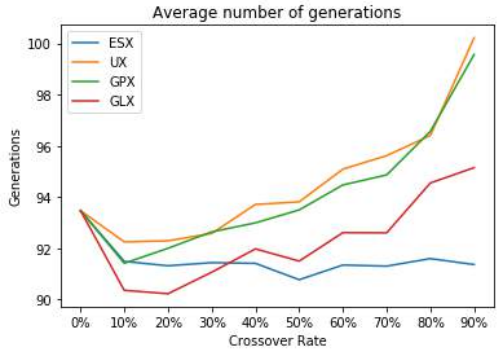


Fig. 6. Average number of generations executed by the GGA-CGT with each state-of-the-art crossover for different crossover rates

1602 instances out of 1615. The minimum number of optimal solutions is found with a crossover rate of 80%, being 1589, and it indeed benefits the algorithm, since without crossover, it finds only 1581 optimal solutions.

From Fig. 5 and Fig. 6 it can be observed that ESX and GLX outperformed the performance of the other two crossover operators, presenting a more stable behavior with different crossover rates.

After the execution of the GGA-CGT with the four crossover operators it was concluded that the ESX operator with a configuration of a 50% of crossover rate and the GLX operator with a 20% crossover rate, found a higher number of optimal solutions, which is why a series of experiments were performed, based on these results and they are described below.

5.2 Reinsertion of Children to the Population

Considering that the GLX algorithm generates two offspring for each pair of crossed parents, for the following experiments, the replacement criteria within the GGA-CGT algorithm were considered. Two versions of the GGA-CGT algorithm were implemented, where the GLX crossover operator generates only one child for each pair of parents.

For the implementation of the following experimentation, the following three cases were addressed: (1) when the ESX algorithm reinserts offspring into the population by replacing individuals with repeated fitness and, if there are still

Table 1. Results obtained by the GGA-CGT with the Uniform crossover (UX) for different crossover rates

Class	Inst.	Crossover rate									
		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
data set 1	720	708	718	719	719	718	718	718	718	719	718
data set 2	480	480	480	480	480	480	480	480	479	480	479
data set 3	10	9	9	10	10	10	10	10	9	10	10
triplets	80	80	80	80	80	79	80	78	78	78	69
uniform	80	80	80	80	80	79	80	80	80	79	80
hard28	28	9	11	9	10	8	8	7	6	7	5
was1	100	100	100	100	100	100	100	100	100	100	100
was 2	100	100	100	100	100	100	100	100	100	100	100
gau 1	17	15	13	13	13	13	13	13	13	13	13
Total	1615	1581	1591	1591	**1592**	1587	1589	1586	1583	1586	1574

Table 2. Results obtained by the GGA-CGT with the Exon Shuffling crossover (ESX) for different crossover rates

Class	Inst.	Crossover rate									
		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
data set 1	720	708	719	719	719	718	718	719	719	718	720
data set 2	480	480	480	480	480	480	480	480	480	480	480
data set 3	10	9	9	10	9	9	10	9	10	9	9
triplets	80	80	80	80	80	80	80	80	80	80	80
uniform	80	80	80	80	80	80	80	80	80	80	80
hard28	28	9	9	10	10	10	12	9	10	9	8
was1	100	100	100	100	100	100	100	100	100	100	100
was 2	100	100	100	100	100	100	100	100	100	100	100
gau 1	17	15	15	16	16	16	16	15	15	15	15
Total	1615	1581	1592	1595	1594	1593	**1596**	1592	1594	1591	1592

offspring without reinsertion, by replacing the worst solutions; (2) when the GLX algorithm reintegrates the created offspring into the population by replacing individuals within the set of random parents R; and, (3) as in the case of the ESX operator, when the GLX algorithm reinserts the offspring into the population by replacing individuals with repeated fitness and, if there are still offspring without reintegration, by replacing the worst solutions.

The case where ESX reinserts the children into the population by replacing individuals within the set of random parents R was presented in Table 2, Fig. 5 and Fig. 6. These results are used in Fig. 7 and Fig. 8 for the ESX-1 operator.

The results obtained by the ESX variation operator, corresponding to the case when using reinsertion into the population by replacing the solutions within the repeated fitness group and worst solutions (ESX-2), are shown in Table 5. The GGA-CGT algorithm manages to solve optimally a maximum of 1593 instances with crossover rates of 30% and 40%. The lowest number of instances that it solves optimally is 1589 with a 10% crossover rate.

On the other hand, the results of the implementation of the GLX operator with the reinsertion into the population by replacing the parents of the random group (GLX-1) are presented in Table 7. As it can be seen, the operator implemented within

Table 3. Results obtained by the GGA-CGT with the Greedy Partition crossover (GPX) for different crossover rates

Class	Inst.	Crossover rate									
		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
data set 1	720	708	720	719	718	719	718	719	719	718	718
data set 2	480	480	480	480	480	480	480	480	480	480	478
data set 3	10	9	9	10	10	10	10	10	10	10	9
triplets	80	80	80	79	80	79	80	79	80	75	73
uniform	80	80	80	80	80	80	80	79	80	80	79
hard28	28	9	11	12	8	9	6	8	6	6	6
was1	100	100	100	100	100	100	100	100	100	100	100
was 2	100	100	100	100	100	100	100	100	100	100	100
gau 1	17	15	15	14	14	13	13	14	14	13	13
Total	1615	1581	**1595**	1594	1590	1590	1587	1589	1589	1582	1576

Table 4. Results obtained by the GGA-CGT with the Gene-level crossover (GLX) for different crossover rates

Class	Inst.	Crossover rate									
		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
data set 1	720	708	719	720	719	718	718	718	718	718	719
data set 2	480	480	480	480	480	480	480	480	480	480	480
data set 3	10	9	9	10	9	9	9	9	9	8	9
triplets	80	80	80	80	79	79	78	76	77	74	78
uniform	80	80	80	80	80	79	80	79	79	79	80
hard28	28	9	14	16	15	16	15	13	14	14	12
was1	100	100	100	100	100	100	100	100	100	100	100
was 2	100	100	100	100	100	100	100	100	100	100	100
gau 1	17	15	16	16	16	16	16	16	15	16	15
Total	1615	1581	1598	**1602**	1598	1597	1596	1591	1592	1589	1593

the algorithm finds a maximum number of 1599 optimal solutions with a 20% crossover rate, while the lowest number of optimal solutions is 1591 with a 80% crossover rate.

The last experiments, corresponding to the GLX with the reinsertion of the offspring in the population replacing the solutions with repeated fitness and worst fitness (GLX-2), are shown in Table 6. This operator allows the GGA-CGT to find a maximum of 1598 optimal solutions with crossover rates of 40%, 60% and 80%, while the minimum number of instances optimally solved is 1595 for 70% and 90% crossover rates, without taking into account the results when there is no crossover.

Based on the results discussed earlier in this section, with respect to the implementation of the two versions of the ESX operator, it is concluded that the ESX operator using a reinsertion of the children in the group of random parents (ESX-1), has a better performance, since it solves on average 1593.22 instances in an optimal way out of 1615. Moreover, the highest number of optimal solutions found by the ESX operator are solved with this reinsertion (1596). On the other hand, the other reinsertion, replacing parents with repeated and worst fitness solves an average of 1591.55 instances in an optimal way.

Regarding the results of the implementation of the GLX operator in the version that only

Table 5. Results obtained by the GGA-CGT with the Exon Shuffling crossover with replacement to individuals with duplicated fitness and the worst solutions (ESX-2) for different crossover rates

Class	Inst.	Crossover rate									
		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
data set 1	720	708	718	719	719	718	718	718	718	718	718
data set 2	480	480	480	480	480	480	480	480	480	480	480
data set 3	10	9	9	9	10	10	9	10	9	9	10
triplets	80	80	80	80	80	80	79	79	79	80	79
uniform	80	80	80	80	80	80	80	80	80	80	80
hard28	28	9	8	9	9	10	9	10	10	10	9
was1	100	100	100	100	100	100	100	100	100	100	100
was 2	100	100	100	100	100	100	100	100	100	100	100
gau 1	17	15	14	15	15	15	16	15	15	15	15
Total	1615	1581	1589	1592	**1593**	**1593**	1591	1592	1591	1592	1591

Table 6. Results obtained by the GGA-CGT with the Gene-level crossover with replacement to individuals with duplicated fitness and the worst solutions (GLX-2) for different crossover rates

Class	Inst.	Crossover rate									
		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
data set 1	720	708	719	719	719	718	719	719	719	720	719
data set 2	480	480	480	480	480	480	480	480	480	480	480
data set 3	10	9	10	9	9	10	10	10	9	10	9
triplets	80	80	80	80	80	80	80	80	80	80	79
uniform	80	80	80	80	80	80	80	80	80	80	80
hard28	28	9	12	13	13	14	12	13	11	12	13
was1	100	100	100	100	100	100	100	100	100	100	100
was 2	100	100	100	100	100	100	100	100	100	100	100
gau 1	17	15	15	16	16	16	16	16	16	16	15
Total	1615	1581	1596	1597	1597	**1598**	1597	**1598**	1595	**1598**	1595

creates one child for every two parents, when the reinsertion into the population is performed by replacing the solutions with repeated and worst fitness (GLX-2), it has better results, finding on average 1596.77 optimal solutions. In the case of the GLX operator with reinsertion to the population by replacing the set of random parents (GLX-1), an average of 1595.44 optimal solutions are found. Moreover, by means of this replacement the highest number of optimal solutions are found by the GLX operator (1599).

5.3 Uniform Crossover with Ordered Genes

After analyzing the different features involved in the crossover operators, we observe that the UX operator was the only one that did not sort the parent solutions in descending order before performing the crossover process. A final implementation of the UX operator was done, where the genes of the parents were first sorted in descending order of their filling, and then the usual crossover process of the operator was performed (UX-sorted). The results are shown in the Table 8. As we can see, the algorithm with a 10% crossover rate solves a total of 1596 instances in

Table 7. Results obtained by the GGA-CGT with the Gene-level crossover with replacement to individuals in the set of the random parents R (GLX-1) for different crossover rates

Class	Inst.	Crossover rate									
		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
data set 1	720	708	719	720	719	720	719	718	718	718	719
data set 2	480	480	480	480	480	480	480	480	480	480	480
data set 3	10	9	10	10	9	9	9	10	9	9	9
triplets	80	80	79	79	79	78	79	76	78	75	79
uniform	80	80	80	80	80	80	80	80	79	79	79
hard28	28	9	13	14	14	15	13	14	15	15	11
was1	100	100	100	100	100	100	100	100	100	100	100
was 2	100	100	100	100	100	100	100	100	100	100	100
gau 1	17	15	16	16	16	16	16	15	16	15	16
Total	1615	1581	1597	**1599**	1597	1598	1596	1593	1595	1591	1593

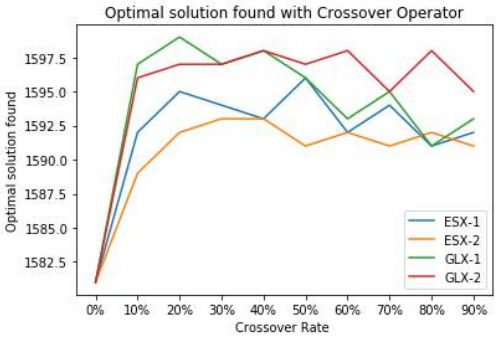


Fig. 7. Number of optimal solutions with different replacements. ESX-1 GLX-1: children replace the individuals in the set of the random parents. ESX-2, GLX-2: children replace the individuals with duplicated fitness and also the worst solutions

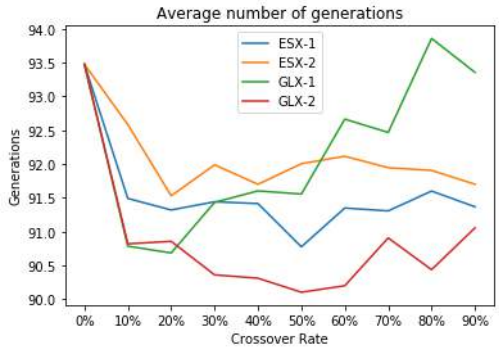


Fig. 8. Average number of generations with different replacements. ESX-1 GLX-1: children replace the individuals in the set of the random parents. ESX-2, GLX-2: children replace the individuals with duplicated fitness and also the worst solutions

an optimal way, while with a 90% crossover rate it affects the performance of the algorithm as it only solves 1574 instances. In addition to the above, a comparison was made between the UX operator and UX-sorted. The results are shown in two tables, one presenting the maximum number of instances solved per class in the (Table 9) and the other with respect to the minimum number of instances solved per class in the (Table 10).

5.4 Robustness of Crossover Operators

To evaluate all of the potential of the best crossover operators, we performed a robustness test by executing three versions of the GGA-CGT thirty times with different seeds of random numbers. These last experiments arose from the comparison of the ESX and GLX operators.

For the GLX operator we used the version in which the children are reinserted in the population by replacing the random parents (GLX-1) with a 50% crossover rate, as well as the original version of the GLX included in the the GGA-CGT

Table 8. Results obtained by the GGA-CGT with the UX-sorted crossover for different crossover rates

Class	Inst.	Crossover rate									
		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
data set 1	720	708	719	720	719	719	719	719	718	718	718
data set 2	480	480	480	480	480	480	480	479	480	480	478
data set 3	10	9	10	10	10	10	10	10	10	10	9
triplets	80	80	80	80	80	78	79	77	77	76	71
uniform	80	80	80	80	80	80	80	79	79	79	79
hard28	28	9	13	10	8	9	7	7	7	7	6
was1	100	100	100	100	100	100	100	100	100	100	100
was 2	100	100	100	100	100	100	100	100	100	100	100
gau 1	17	15	14	13	13	13	13	13	14	14	14
Total	1615	1581	**1596**	1593	1590	1589	1588	1584	1585	1584	1575

Table 9. Results of the highest number of optimally solved instances by the GGA-CGT with the UX and UX-sorted crossover operators

Class	Inst.	UX	UX-sorted
		Crossover rate	
		30%	10%
data set 1	720	719	719
data set 2	480	480	480
data set 3	10	10	10
triplets	80	80	80
uniform	80	80	80
hard28	28	10	13
was1	100	100	100
was 2	100	100	100
gau 1	17	13	14
Total	1615	1592	1596

Table 10. Results of the lowest number of optimally solved instances by the GGA-CGT with the UX and UX-sorted crossover operators

Class	Inst.	UX	UX-sorted
		Crossover rate	
		90%	90%
data set 1	720	718	718
data set 2	480	479	478
data set 3	10	10	9
triplets	80	69	71
uniform	80	80	79
hard28	28	5	6
was1	100	100	100
was 2	100	100	100
gau 1	17	13	14
Total	1615	1574	1575

algorithm by Quiroz-Castellanos et al. [20], that generates two children with a 20% crossover rate. On the other hand, for the ESX operator we also used the version in which the children are reinserted in the population by replacing the random parents (ESX-1) with a crossover rate of 50%. These operators were selected as they are within the group of operators that showed the higher effectiveness. The experiment consists of 30 executions of the GGA-CGT algorithm (48,450 runs) with each crossover operator.

The results of the experiments are shown in Table 11, Table 12 and Table 13. The second column (Inst.), corresponds to the total number of instances belonging to the class in the first column. The column “Optimal solutions in the 30 executions” includes the total number of optimal solutions found in the 30 executions.

The column “Average number of optimal” corresponds to the average number of instances optimally solved in each set. Finally, the column

Table 11. Results for the 30 executions of the GGA-CGT with the ESX when it generates only one child

Class	Inst.	Optimal solutions in the 30 executions	Average number of optimal	%
data set 1	720	21562	718.73	99.82%
data set 2	480	14400	480.00	100.00%
data set 3	10	282	9.40	94.00%
triplets	80	2396	79.87	99.83%
uniform	80	2400	80.00	100.00%
hard28	28	299	9.97	35.60%
was 1	100	3000	100.00	100.00%
was 2	100	3000	100.00	100.00%
gau 1	17	467	15.57	91.57%
	1615	47806	1593.53	98.67%

Table 12. Results for the 30 executions of the GGA-CGT with the GLX when it generates only one child

Class	Inst.	Optimal solutions in the 30 executions	Average number of optimal	%
data set 1	720	21562	718.73	99.82%
data set 2	480	14400	480.00	100.00%
data set 3	10	281	9.37	93.67%
triplets	80	2370	79.00	98.75%
uniform	80	2393	79.77	99.71%
hard28	28	394	13.13	46.90%
was 1	100	3000	100.00	100.00%
was 2	100	3000	100.00	100.00%
gau 1	17	476	15.87	93.33%
	1615	47876	1595.87	98.82%

"%" corresponds to the percentage of optimal solutions found.

It is observed that the GGA-CGT algorithm with the GLX operator, which generates two children, found on average a higher number of optimal solutions, with a 98.99% percentage.

While the GLX and the ESX that generate one child found 98.8% and 98.67% of the optimal solutions, respectively. Furthermore, on average, the original GGA-CGT with the GLX operator that generates two children resolves in an optimal way an average of 1598.67 instances out of 1615. Regarding to the GLX and the ESX that generate one child, they found on average 1595.87 and 1593.53 optimal solutions, respectively.

5.5 Statistical Analysis of the Effectiveness of GGA-CGT with the Best Crossover Operators

A series of statistical tests were applied to the results of the 30 executions of the GGA-CGT algorithm with the ESX, GLX and GLX-V1

Table 13. Results for the 30 executions of the original GGA-CGT with the GLX that generates two children

Class	Inst.	Optimal solutions in the 30 executions	Average number of optimal	%
data set 1	720	21571	719.03	99.87%
data set 2	480	14400	480.00	100.00%
data set 3	10	291	9.70	97.00%
triplets	80	2391	79.70	99.63%
uniform	80	2400	80.00	100.00%
hard28	28	430	14.33	51.19%
was 1	100	3000	100.00	100.00%
was 2	100	3000	100.00	100.00%
gau 1	17	477	15.90	93.53%
	1615	47960	1598.67	98.99%

operators. The statistical test applied was the Wilcoxon Rank-sum test with a 95% confidence. These tests were performed on two result sets. First, the error of the results was calculated for each instance in each execution, which was determined as the relative difference defined as: $(y - x)/x$, where x corresponds to the optimal number of bins and y corresponds to the number of bins obtained.

For the first set of experiments, the average error of the 30 executions was obtained for each instance.

Next, the Wilcoxon test was applied to the average errors of the algorithm per class of instances with the three operators: ESX vs GLX-V1, ESX vs GLX and GLX-V1 vs GLX, obtaining 9 p-values (one for each class). The results are shown in the Table 15.

For the second set of tests, all instances were considered separately. A Wilcoxon test was performed for each instance with the results obtained from the algorithm with the different operators, i.e.: ESX vs GLX-V1, ESX vs GLX and GLX-V1 vs GLX. The Table 16 shows the results of the instances where a significant difference was observed in the results obtained with the different operators.

The Wilcoxon test, at the class level, shows no significant difference between the operators, which would indicate that the performance is the same among the three. However, the test at the instance level does show a difference, in some instances of the Hard28 and Gau 1 classes, between the ESX operator and the GLX-V1 and GLX operators, indicating that for these instances the ESX operator is the worst operator.

5.6 Difficult 1D-BPP Instances

Finally, with the objective to identify the features of the 1D-BPP instances that present the highest degree of difficulty for the four state-of-the-art grouping crossovers, Table 16 shows the set of instances for which the optimum was not found. These results were obtained from the set of experiments performed with the different crossover rates, selecting those instances that were not solved optimally in at least one of the configurations.

The first column includes the name of the sets of instances that include cases for which the optimal solutions could not been found for some of the configurations of the four crossover operators. The second column contains the name of the difficult instances, and the following columns include an X for each operator that fails to find the optimal solutions for each instance in one of its configurations. The instances belong to the classes: data set 1, data set 2, data set 3, triplets,

Table 14. Instances for which the GGA-CGT does not find the optimal solution with each operator for any of the crossover rates

Class	Inst.	UX	ESX	GPX	GLX	Class	Inst.	UX	ESX	GPX	GLX
	N3c1w2_r	X					t501_17				X
data set 1	N3c3w4_c	X	X	X	X	triplets	t501_18	X			X
	N4c3w4_s	X	X	X	X		t501_19				X
data set 2	N2w1b2r5	X		X		Uniform	u250_12	X		X	X
	N2w1b2r8			X			h1D-BPP832	X		X	X
data set 3	Hard2			X			h1D-BPP40	X	X	X	X
	t60_00	X					h1D-BPP360	X	X	X	
	t60_01	X		X			h1D-BPP645	X	X	X	X
	t60_05	X					h1D-BPP742	X	X	X	
	t60_06	X		X			h1D-BPP766	X	X	X	X
	t60_07	X					h1D-BPP60	X	X	X	X
	t60_08	X					h1D-BPP13	X	X	X	X
	t60_10			X			h1D-BPP195	X	X	X	X
	t60_11	X					h1D-BPP709	X	X	X	X
	t60_12	X					h1D-BPP785	X	X	X	X
	t60_13	X				hard28	h1D-BPP47	X	X	X	X
	t60_14				X		h1D-BPP181	X	X	X	X
	t60_15				X		h1D-BPP485	X	X	X	X
triplets	t60_17	X					h1D-BPP640	X	X	X	X
	t60_18	X					h1D-BPP144	X	X	X	X
	t60_19						h1D-BPP561	X		X	X
	t120_03	X					h1D-BPP781	X	X	X	X
	t120_14	X					h1D-BPP900	X	X	X	X
	t249_04				X		h1D-BPP178	X	X	X	X
	t501_00			X			h1D-BPP419	X	X	X	X
	t501_04				X		h1D-BPP531	X	X	X	
	t501_05	X		X	X		h1D-BPP814	X	X	X	
	t501_08				X		TEST0058	X		X	
	t501_09				X	gau1	TEST0014	X	X	X	X
	t501_14			X	X		TEST0030	X	X	X	X
	t501_16			X			TEST0005	X		X	X
							Total	46	25	39	36

uniform, hard28 and gau 1. The highest number of instances where the optimum is not found belong to the triplets class. With respect to the class data set 3 and uniform, only one instance is not optimally solved.

As it can be seen, the instances belonging to the hard28 class are the most complicated for the 4 operators. The operator with the lowest number of optimally solved instances is the ESX operator. Although the GLX operator optimally solves the highest number of instances, these instances tend to be more variable with different crossover percentages.

The UX operator, on the other hand, is the one that has the greatest diversity with respect to the number of instances not optimally resolved, being a total of 46. With respect to the instances belonging to the triplets class, the ESX operator

Table 15. *p*-value of the Wilcoxon test for the error of the GGA-CGT with the ESX, GLX-V1 and GLX crossovers in the different classes of instances

Class	*p*-value for the average error		
	ESX vs GLX-V1	ESX vs GLX	GLX-VI vs GLX
data set 1	0.9637	0.9636	1
data set 2	1	1	1
data set 3	0.9698	0.9698	0.9698
triplets	0.2459	0.1557	0.8618
uniform	0.8914	0.8914	0.9986
hard 28	0.5552	0.5121	0.9151
was 1	1	1	1
was 2	1	1	1
gau 1	0.7828	0.7828	0.9862

is the only operator that has no problem in finding the optimum, besides being the only one to solve optimally all the instances of the Uniform class.

Although the GGA-CGT algorithm finds the optimal solution of most of the well-known benchmark instances within the state-of-the-art, it has been shown that it does not perform well with the 2800 new difficult instances, referred to as $BPPv_{u}\_c$, proposed by Carmona-Arroyo et al. [6]. The results that were obtained by the GGA-CGT demonstrated that for most of these instances, the operators included in the GGA-CGT do not appear to lead to better solutions.

6 Conclusion and Future Work

In this research we propose an experimental study about the GGA-CGT for the 1D-BPP, in which different grouping crossover operators were used to compare and mesure the performance. The conclusions of this research are presented as follows.

From the experiments carried out in Section 5, it is concluded that the original Gene-level crossover operator (GLX), when generating two children, outperforms the other state-of-the-art grouping crossovers, achieving a better performance with respect to the number of optimal solutions found, being 1602 out of 1615. The second experiment, also explained in Section 5, on the implementation of the ESX and GLX operators in their two versions (generating only one child, with replacement to random parents, and, with replacement to solutions with repeated fitness and worst solutions), shows that the best operator is the GLX operator with the reinsertion to the population through the replacement to random parents, since it solves optimally a total of 1599 instances out of 1615.

The last experiment allow us to validate that the original GLX operator, that generates two children, is the one that achieves the best performance, showing a robust behavior in different runs of the GGA-CGT. Finally, it can be concluded that the GLX operator performs better within the GGA-CGT algorithm.

Although the GLX operator solves a larger number of instances, per class there is no significant difference between the operators. But there is a difference for some instances of the Hard 28 class, where it has been shown that the GLX operator has the best performance of the operators.

For future work, a detailed review of the operators is planned to identify the strategies that allow for them to improve their performance in the different classes instances, with the objective of using this knowledge to design a new crossover operator with a better performance in terms of time, number of generations and number of optimal solutions found for instances of different classes. It is also intended that the GGA-CGT algorithm with the operator to be designed will improve the performance of the algorithm for the $BPPv_{u}\_c$ instances with which it does not perform well.

References

1. **Abdul-Minaam, D. S., Al-Mutairi, W. M. E. S., Awad, M. A., El-Ashmawi, W. H. (2020).** An adaptive fitness-dependent optimizer for the one-dimensional bin packing problem. IEEE Access, Vol. 8, pp. 97959–97974.
2. **Alvim A.C., G. F. e. a., Ribeiro C.C. (2004).** A hybrid improvement heuristic for the one-dimensional bin packing problem. Journal of heuristics, Vol. 10, pp. 205–229.

Table 16. *p*-value of the Wilcoxon test for the error of the GGA-CGT with the ESX, GLX-V1 and GLX crossovers in the most difficult instances

Class	Instance	*p*-value of the error of the 30 executions		
		ESX vs GLX-V1	ESX vs GLX	GLX-V1 vs GLX
hard 28	h1D-BPP360.txt	0.0147	0.147	1
	h1D-BPP742.txt	0.0039	0.0009	0.6574
	h1D-BPP47.txt	3.3853x10-7	2.9537x10-8	0.6574
	h1D-BPP640.txt	0.01471	0.01471	1
	h1D-BPP531.txt	2.9537x10-8	2.9537x10-8	1
	h1D-BPP814.txt	0.0039	0.0039	1
gau 1	TEST0030.txt	0.0358	0.069	0.6574

3. **Borgulya, I. (2020).** A hybrid evolutionary algorithm for the offline bin packing problem. Central European Journal of Operations Research, Vol. 29, No. 2, pp. 425–445.
4. **Buljubašić, M., Vasquez, M. (2016).** Consistent neighborhood search for one-dimensional bin packing and two-dimensional vector packing. Computers & Operations Research, Vol. 76, pp. 12–21.
5. **Cardoso Silva, A., Hasenclever Borges, C. C. (2019).** An improved heuristic based genetic algorithm for bin packing problem. 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), pp. 60–65.
6. **Carmona-Arroyo G., Q.-C. M., Vázquez-Aguirre J.B. (2021).** One-Dimensional Bin Packing Problem: An Experimental Study of Instances Difficulty and Algorithms Performance, volume 940. Springer, Cham.
7. **Delorme, M., Iori, M., Martello, S. (2016).** Bin packing and cutting stock problems: Mathematical models and exact algorithms. European Journal of Operational Research, Vol. 255, No. 1, pp. 1–20.
8. **Delorme, M., Iori, M., Martello, S. (2018).** BPPLIB: a library for bin packing and cutting stock problems. Optim. Lett., Vol. 12, No. 2, pp. 235–250.
9. **Dokeroglu, T., Cosar, A. (2014).** Optimization of one-dimensional bin packing problem with island parallel grouping genetic algorithms. Computers & Industrial Engineering, Vol. 75, pp. 176–186.
10. **Falkenauer, E. (1996).** A hybrid grouping genetic algorithm for bin packing. Journal of heuristics, Vol. 2(1), pp. 5–30.
11. **Feng, L., Xiao, X., Mi, Z., Yi, X., Zhang, H. (2020).** A particle swarm optimization algorithm for layout design of user interfaces in vehicle system. 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT, pp. 365–368.
12. **Fukunaga, A. S. (2008).** A new grouping genetic algorithm for the multiple knapsack problem. 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 2225–2232.
13. **K., F., S., H. K. (2002).** New heuristics for one-dimensional bin-packing. Computers & operations research, Vol. 29(7), pp. 821–839.
14. **Kolkman, S. W., J. (2001).** Directed evolution of proteins by exon shuffling. Nat Biotechnol, Vol. 19, pp. 423–428.
15. **Kucukyilmaz, T., Kiziloz, H. E. (2018).** Cooperative parallel grouping genetic algorithm for the one-dimensional bin packing problem. Computers & Industrial Engineering, Vol. 125, pp. 157–170.
16. **Levine, J., Ducatelle, F. (2004).** Ant colony optimization and local search for bin packing and cutting stock problems. Journal of the Operational Research Society, Vol. 55, No. 7, pp. 705–716.
17. **Lijun Wei, R. B. A. L., Zhixing Luo (2019).** A new branch-and-price-and-cut algorithm for one-dimensional bin-packing problems. INFORMS Journal on Computing, Vol. 32, No. 2, pp. 428–443.
18. **Munien, C., Mahabeer, S., Dzitiro, E., Singh, S., Zungu, S., Ezugwu, A. E.-S. (2020).** Metaheuristic approaches for one-dimensional bin packing problem: A comparative performance study. IEEE Access, Vol. 8, pp. 227438–227465.
19. **Ozcan, S. O., Dokeroglu, T., Cosar, A., Yazici, A. (2016).** A novel grouping genetic algorithm for

the one-dimensional bin packing problem on gpu. **Czachórski, T., Gelenbe, E., Grochla, K., Lent, R.**, editors, Computer and Information Sciences, Springer International Publishing, Cham, pp. 52–60.

20. **Quiroz-Castellanos, M., Cruz-Reyes, L., Torres-Jimenez, J., Gómez S., C., Huacuja, H. J. F., Alvim, A. C. (2015).** A grouping genetic algorithm with controlled gene transmission for the bin packing problem. Comput. Oper. Res., Vol. 55, No. C, pp. 52–64.
21. **Ramos-Figueroa, O., Quiroz-Castellanos, M., Mezura-Montes, E., Kharel, R. (2021).** Variation operators for grouping genetic algorithms: A review. Swarm and Evolutionary Computation, Vol. 60, pp. 100796.
22. **Rivera, G., Cisneros, L., Sánchez-Solís, P., Rangel-Valdez, N., Rodas-Osollo, J. (2020).** Genetic algorithm for scheduling optimization considering heterogeneous containers: A real-world case study. Axioms, Vol. 9, No. 1.
23. **Singh, A., Gupta, A. K. (2007).** Two heuristics for the one-dimensional bin-packing problem. OR Spectr., Vol. 29, No. 4, pp. 765–781.
24. **Tan, B., Ma, H., Mei, Y. (2020).** A group genetic algorithm for resource allocation in container-based clouds. **Paquete, L., Zarges, C.**, editors, Evolutionary Computation in Combinatorial Optimization, Springer International Publishing, Cham, pp. 180–196.
25. **Wilcox, D., McNabb, A., Seppi, K. (2011).** Solving virtual machine packing with a reordering grouping genetic algorithm. 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 362–369.

Article received on 09/07/2021; accepted on 21/11/2021.
Corresponding author is Efrén Mezura-Montes.

ISSN 2007-9737

Ensemble Recurrent Neural Network Design using a Genetic Algorithm applied in Times Series Prediction

Martha Pulido, Patricia Melin

Tijuana Institute of Technology,
Mexico

{martha.pulido, pmelin}@tectijuana.mx

Abstract. This paper shows a new method based on ensemble recurrent neural networks for time series prediction. The proposed method seeks to find the structure of ensemble recurrent neural network and its optimization with Genetic Algorithms applied to the prediction of time series. For this method, two systems are proposed to integrate responses ensemble recurrent neural network that are type-1 and Interval type-2 Fuzzy Systems. The optimization consists of the modules, hidden layer, neurons of the ensemble neural network. The fuzzy system used is of Mamdani type, which has five input variables and one output variable, and the number of inputs of the fuzzy system is according to the outputs of Ensemble Recurrent Neural network. Test are performed with Mackey Glass benchmark, Mexican Stock Exchange, Dow Jones and Exchange Rate of US Dollar/Mexican Pesos. In this way was shown that the method is effective for time series Prediction.

Keywords. Time series prediction, genetic algorithm, ensemble recurrent neural network.

1 Introduction

Recurrent neural networks (RNNs) were already conceived in the 1980s. But these types of neural networks have been very difficult to train due to their computing requirements and until the arrival of the advances of recent years, their use by the industry has become more accessible and popular [2, 3, 4, 5, 10]. A recurrent neural network (RNN) is one that can be connected to any other and its recurring connection is variable. Partially recurring networks are those that your recurring connection fixed [1, 2, 9, 11, 14, 17].

Recurrent Neural Networks are dynamic systems mapping input sequences into output sequences [19, 21, 22, 23].

The calculation of an input, in a step, depends on the previous step and in some cases the future step [34, 35, 40, 44]. RNN are capable of performing a wide variety of computational tasks including sequence processing, one-path continuation, nonlinear prediction, and dynamical systems modeling [38, 47, 49, 50, 52].

The purpose of carrying out a time series analysis of this type is to extract the regularities that are observed in the past behavior of a variable that is, obtain the mechanism that generates it, and know its behavior based on it over time. This is under the assumption that the structural conditions that make up the phenomenon under study remain constant, to predict future behavior by reducing uncertainty in decision-making [6, 7 ,8, 12, 29].

The essence of this work, is propose a new algorithm to design time prediction systems, where recurrent neural networks are applied to analyze the data, also type-1 and interval type-2 fuzzy inference systems to improve the prediction of time series. For this, we apply a search algorithm to obtain the best architecture of the recurrent neural network, and in this way test the efficiency of the proposed hybrid method [6, 7, 8, 12, 29].

Genetic algorithms have been applied to various areas, such as forecasting classification, image segmentation routes for robots stand out, etc. Their hybridation with other techniques improves the results of energy price predictions, raw materials and agricultural products, that is why we apply them to this approach, since they are tools that help use predict a time series and find good solutions, we have previously done work with this metaheuristic and they have given us good results, for this reason we apply it to network optimization neural ensemble for time series.

This work describes the creation of the ensemble recurrent neural network (ERNN), this model is applied to the prediction of the time series [28, 31, 36, 39, 41], the architecture is optimized with genetic algorithms, (GA) [16, 26, 27, 46]. The results of the integration of the Recurrent Neural Network are integrated with type-1 and interval type-2 fuzzy systems (IT2FS), [30, 32, 33, 43, 48]. The essence of this paper is the proposed architecture of the ERNN and the optimization is done using Genetic Algorithms, (GA), applied to the prediction of time series. Two systems are proposed fuzzy type-1 and T2FS, to integrate the responses of the (ERNN), the optimization consists in the number of hidden layer (NL), their number of neurons (NN) and the number of modules (NM),) in the ERNN, the we integrate responses ERNN, with type-1 and IT2FS and in this way we achieve prediction. Mamdani fuzzy inference system (FIS) has five inputs which are Pred1, Pred2, Pred3, Pred4, and Pred5 and one output is called prediction. The number of inputs of the fuzzy system (FS) is according to the outputs of ERNN. Mamdani fuzzy inference system (FIS) is created, this FIS five inputs which are Pred1, Pred2, Pred3, Pred4, and Pred5, (with a range) the range 0 to 1.4, the outputs is called prediction, the range goes from 0 to 1.4 and is granulated into two MFs "Low", "High", as linguistic variables.

This document is constituted by: Section 2 shows the database, in Section 3 the problem to be solved of the proposed model, Section 4 shows results of the proposed model, and Section 5 Conclusions.

2 Related Word

As related work, we can find a comparison was made using recurrent networks for the Puget Electric Demand time series, a learning algorithm was implemented for recurrent neural networks and tests were performed with outliers of data and in this way compare the capacity of loses, as well as the advantages of feedforward neural networks for time series are shown [18].

In [45] a recurrent neural network was developed for prognostic problems; the time series of long memory patterns was used and tests were also carried out with the integrated fractional recurrent neural network (FIRNN) algorithm.

In another study it was shown the performance of the cooperative neuro-evolutionary methods, in this case Mackey-Glass, Lorenz and Sunspot time series, and also two training methods Elman recurrent neural networks [37].

In another study, the prediction of time series was carried out, and the recurrent neural networks were used to make predictions. In this way the effectiveness of recurrent networks for the forecasting of chaotic time series was demonstrated [51].

In the study presented in [20], Recurrent Neural Networks (RNNs), are used to model the seasonality of a series in dataset possess homogeneous seasonal patterns. Comparisons with the autoregressive integrated (ARIMA) and against exponential smoothing (ETS), demonstrate that RNN models are not best solutions, but they are competitive.

In [57], advanced neural networks were used for short term prediction. Also, the exponential smoothed (ES) model for the time series are used, and this allows the equations to capture seasonality and the level more effectively, these networks allow trends non-linear and cross-learning, data is exploited hierarchically and local and global components are used to extract and combine data from or from a data series in order to obtain a good prediction.

This section shows the data set that was used to build the proposed model, in this case the Mackey-Glass time series are used.

3.1 Dataset Proposed

In this case we work the Mackey Glass time series with eight hundred data, we used 70% and 30% data for the training and testing, respectively.

The following equation represents the Mackey-Glass time series:

$$x = \frac{0.2\mathcal{X}(t-\tau)}{1+\mathcal{X}^{10}(\mathrm{t}-\tau)}, \tag{1}$$

where:

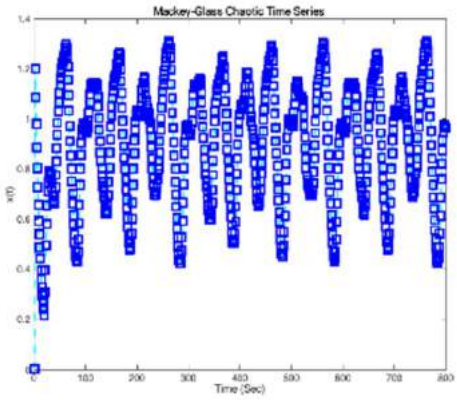


Fig. 1. Plot of the Mackey-Glass Data Set

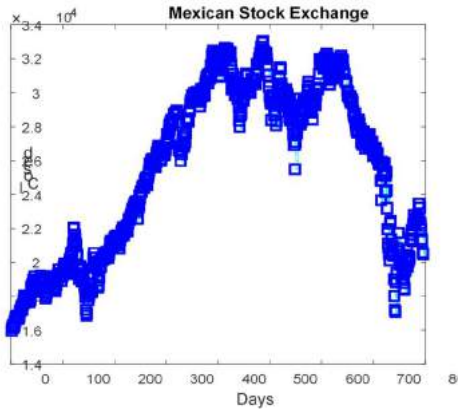


Fig. 2. Plot of the Mexican Stock Exchange Dataset

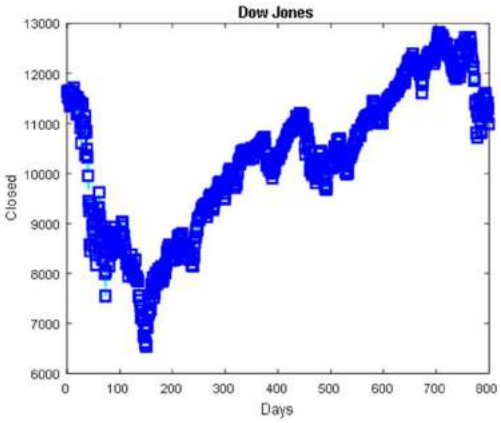


Fig. 3. Plot of the Dow Jones Dataset

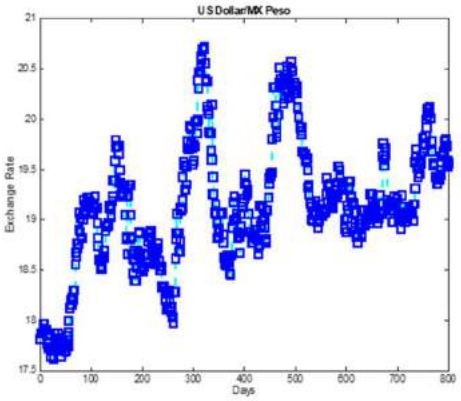


Fig. 4. Plot of the US Dollar/MX Pesos

$$x(t) = 0,$$

$$x(0) = 1.2,$$

$$\mathcal{T} = 17,$$

$$t < 0.$$

The plot of the Mackey Glass for the values mentioned in the equation is presented in Fig. 1. [25, 26].

In Fig. 2, the graph of the Mexican Stock Exchange data [42] is presented. In this case, we use 800 data that correspond to period from 01/04/2016 to 31/12/2019. We used 70% of the data for the RNN trainings and 30% to test the RNN.

Fig. 3 presents the graph of the Dow Jones data [16], where we are using 800 data that correspond to period from 07/01/2017 to 09/05/2019. We used 70% of the data for the RNN trainings and 30% to test the RNN.

In Fig, 4 the graph of the data US Dollar/MX Peso [13] is illustrated, where we use 800 data that correspond to the period from 07/01/2016 to 09/05/2019. We used 70% of the RNN training and 30% to test the RNN.

We trained the ensemble recurrent neural network with 500 data points, and we use the Bayesian regularization backpropagation method

(trainbr) with a set of 300 points for testing, this is for each of the previously mentioned times series.

4 Problem Statement and the Proposed Method

In this Section, it is explained how the ERNN optimization model was created, its integration with type-1 and IT2FS, and we also describe every detail of the technique used for the optimization of the ERNN, as well as type-1 and IT2FS for the prediction of the time series.

4.1 Proposed General Scheme

The first part is to obtain the dataset of the time series, the second part it is determining the number of modules ERNN with the genetic algorithm, and the third part would be to integrate with type-1 and T2FS type-2 the responses of the ERNN to finally achieve time series prediction, as can be observed in Fig. 5.

4.1.1 Creation of the Recurrent Neural Network (RNN)

Recurrent neural networks (RNNs) have all the characteristics and the same operation of simple neural networks, with the addition of inputs that reflect the state of the previous iteration. They are a type of network whose connections form a closed circle with a loop, where the signal is forwarded back to the network, that is, the neural network's own outputs become inputs for later instants. This feature endows them with memory and makes them suitable for time series modeling. The layer that contains the delay units is also called the context layer, as shown (as can be illustrated) in Fig. 6:

The recurrent neural network is made up of several units of a fixed activation function, one for each time step, each unit has a state and it is called the hidden state of the unit and it means that the network has past knowledge and a certain time step. This hidden state is updated and signifies the change in the network's knowledge about the past:

$$y_t = fW(x_t, h_{t-1}), \quad (2)$$

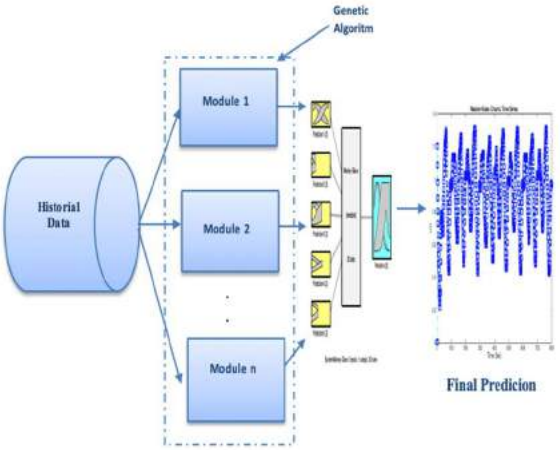


Fig. 5. Proposed General Scheme

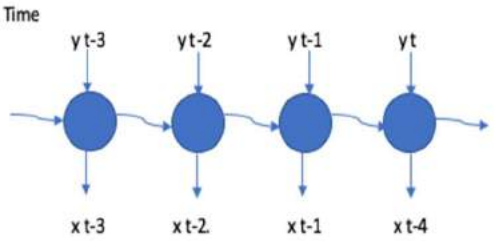


Fig 6. Recurrent Neural Network

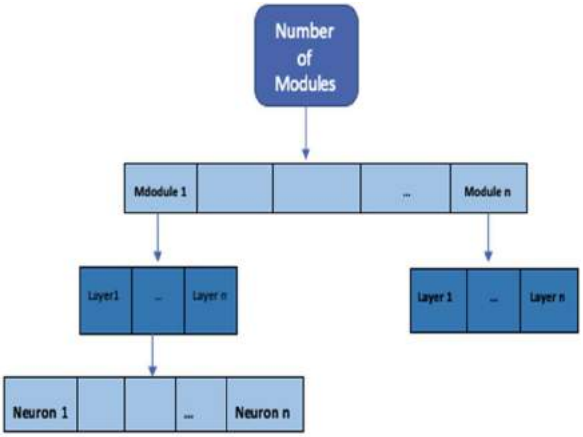


Fig. 7. Chromosome Structure to optimize the RNN

where h_{t-1} the old hidden state, h_t represents the new hidden state, fW the fixed function with trainable weights and x_t is the concurrent input.

The new hidden state is calculated at each time step, and recurrence is used as above and the new

hidden state is used to generate a new state, and so on.

Where the input stream from the previous layer, the weights of the matrix, and bias already seen in the previous layers. RNNs extend this function with a recurring connection in time, where the weight matrix operates on the state of the neural network at the previous time instant. Now, in the training phase through Backpropagation, the weights of this matrix are also updated.

4.1.2 Description of the GA for RNN Optimizations

The parameters of the recurrent neural network that are optimized with the GA are:

1. Number of modules (NM).
2. Number of hidden layers (NL).
3. Number of neurons of each hidden layer (NN).

The following equation represents the objective function that (that is implemented in GA) we used with a genetic algorithm to minimize to prediction error of the time series:

$$ERM = (\sum_{i=1}^{d} |p_i - x_i|)/\mathrm{d}$$
$$Prediction\ Error = (ERM_1 + ERM_2 + \cdots + ERM_N)/N, \quad (3)$$

where p represents the predicted the data for each module of ensemble recurrent network, X corresponds to the real data of time series, d is the number of data used by time series, ERM is the prediction error by module of ERNN, to N corresponds the number of modules determined by the GA and the $Predicion\ Error$ corresponds to average prediction error achieved by the ERNN.

Fig. 7 presents the structure of the GA chromosome.

The main goal to optimize the ERNN architecture, with a GA is to obtain the best prediction error, which seeks to optimize NM, NL, and NN of the ERNN. Table 1 shows the values of the search space of the GA.

In Table 2, the values of the parameters used for each optimization algorithm are presented. The mutation value is variable and shown in Table 2.

Table 1. Table of values for search space

Parameters of RNN	Minimum	Maximum
Modules	1	5
Hidden Layers	1	3
Neurons for each hidden Layer	1	30

Table 2. Table of GA parameters

Parameter	Value
Generations	100
Individuals	100
Crossover	Single Point
Probability	0.85
Selection	Roulette
Mutation	Variable

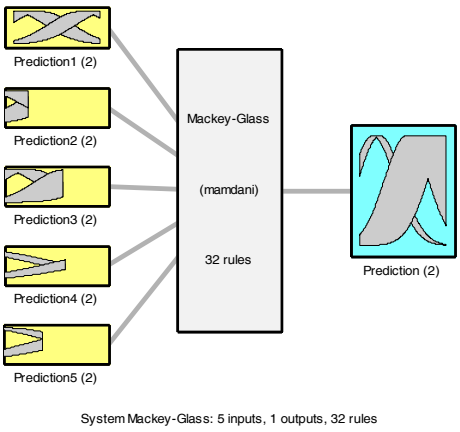


Fig. 8. IT2FS

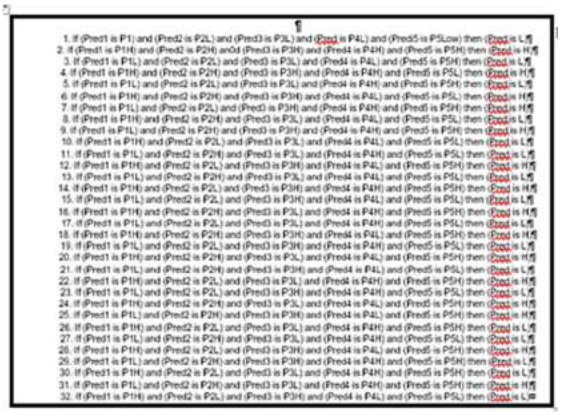


Fig. 9. Rules used for the IT2FS

Table 3. Genetic algorithm results for the RNN of MG

Evolutions	Gen.	Ind.	Pm	Pc	Num. Modules	Num. Layers	Num. Neurons	Duration	Prediction Error
1	100	100	0.07	0.6	3	2	22,23 18,19 17,16	05:10:11	0.0017568
2	100	100	0.05	0.7	5	2	18,22 23,24 25,26 20,21 18,20	06:22:16	0.0019567
3	100	100	0.07	0.5	4	2	25,26 20,22 24,25 21,22	07:24:22	0.0020174
4	100	100	0.03	0.4	5	2	18,22 23,24 25,26 20,21 18,20	07:36:27	0.0016789
5	100	100	0.09	0.9	3	2	18,22 21,22 15,16	06:15:16	0.0017890
6	100	100	0.05	0.5	5	2	19,20 23,24 25,26	09:35:23	0.0020191
7	100	100	0.09	0.9	3	2	24,25 23,22 20,21	0:06:17	0.0015678
8	100	100	0.09	1	5	2	19,18 21,22 27,28 24,24 21,22	06:12:11	0.0018904
9	100	100	0.04	0.7	4	2	19,20 21,22 25,25 18,22	06:13:34	0.0855
10	100	100	0.03	0.7	5	3	20,19,22 18,19,19 22,24,27 21,19,20 26,25,26	08:20:19	0.0016311

T Table 4. Results of type-1 FS for MG

Test	Prediction Error with Type-1 Fuzzy Integration
1	0.1731
2	0.2012
3	0.1965
4	0.2034
5	0.1886
6	0.2898
7	0.2234
8	0.1667
9	0.1945
10	0.2225

Table 5. Results of the IT2FS of MG

Test	Prediction Error 0.3 Uncertainty	Prediction Error 0.4 Uncertainty	Prediction Error 0.5 Uncertainty
1	0.3122	0.2815	0.2512
2	0.3321	0.3017	0.2906
3	0.4256	0.3792	0.4326
4	0.3689	0.3512	0.3891
5	0.5995	0.5725	0.5519
6	0.4912	0.4315	0.4654
7	0.5276	0.5045	0.5618
8	0.3044	0.3426	0.3725
9	0.5122	0.5389	0.5554
10	0.5572	0.5437	0.5215

Table 6. Genetic algorithm results for the RNN of MSE

Evolutions	Gen.	Ind.	Pm	Pc	Num. Modules	Num. Layers	Num. Neurons	Duration	Prediction Error
1	100	100	0.07	0.6	3	3	28,6,24 28,6,24 14,30,26	01:27:18	0.0048872
2	100	100	0.05	0.7	2	2	28,12 28,12	01:16:49	0.0047646
3	100	100	0.07	0.5	2	1	15 15	00:56:20	0.005684
4	100	100	0.03	0.4	2	2	18,2 18,2	01:24:27	0.00488
5	100	100	0.09	0.9	2	2	1,12 1,12	01:05:01	0.004078
6	100	100	0.05	0.5	2	2	22,21 11,12	01:00:19	0.004108
7	100	100	0.09	0.9	2	2	1,3 1,3	01:23:08	0.0053897
8	100	100	0.09	1	5	5	1 1 1 7 8	01:37:40	0.0021431
9	100	100	0.04	0.7	2	1	30 30	01:44:21	0.004596
10	100	100	0.03	0.7	2	3	1,3,28 1,3,28	02:00:22	0.0056895

Table 7. Results of type-1 FS for MSE

Test	Prediction Error With Type-1 Fuzzy Integration
1	0.3272
2	0.3275
3	0.3271
4	0.3270
5	0.3271
6	0.3272
7	0.3271
8	0.3280
9	0.3271
10	0.3273

Table 8. Results of type-2 FS of MS

Test	Prediction Error 0.3 Uncertainty	Prediction Error 0.4 Uncertainty	Prediction Error 0.5 Uncertainty
1	0.3122	0.2815	0.2512
2	0.3321	0.3017	0.2906
3	0.4256	0.3792	0.4326
4	0.3689	0.3512	0.3891
5	0.5995	0.5725	0.5519
6	0.4912	0.4315	0.4654
7	0.5276	0.5045	0.5618
8	0.3044	0.3426	0.3725
9	0.5122	0.5389	0.5554
10	0.5572	0.5437	0.5215

4.1.3. Description of the type-1 and IT2FS

The next step is the description of the type-1 fuzzy system and IT2FS. The following equation shows how the total results of the FS are calculated:

$$y = \frac{\sum_{i=1}^{n} x_i\, u(x_i)}{\sum_{i=1}^{n} u(x_i)_i}, \tag{4}$$

where u represents the MFs and x corresponds to the input data.

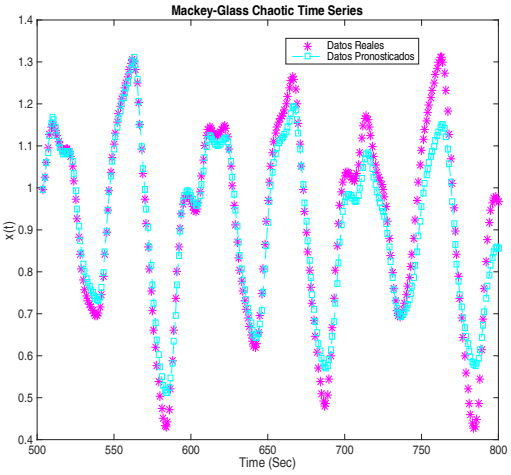


Fig. 9. Graph of real data against predicted data for the type-1 fuzzy system of MG

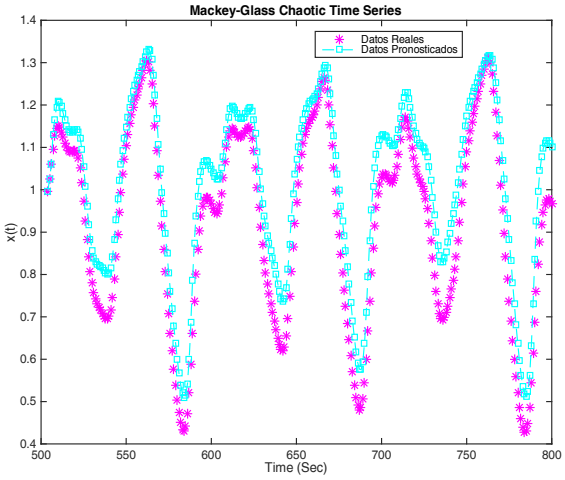


Fig. 10. Plot of real data against predicted data for the T2FS of MG.

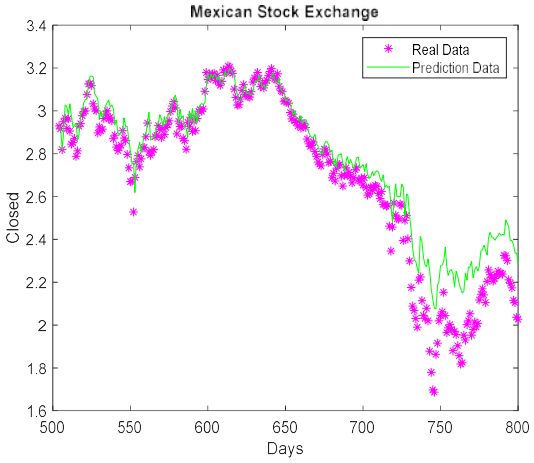


Fig. 11. Plot of real data against predicted data for the type-1 fuzzy system of MSE

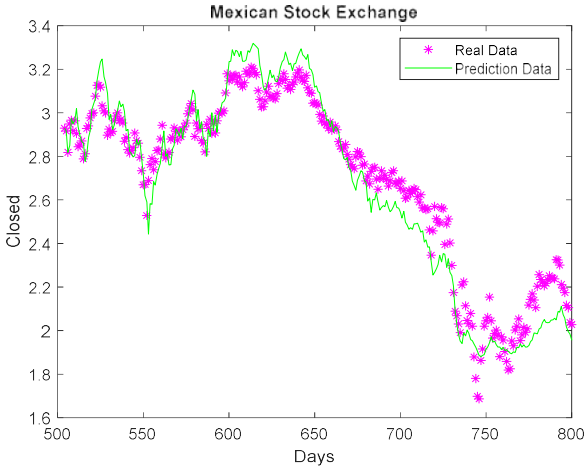


Fig. 12. Plot of the real data against predicted data for the T2FS

Fig. 8 shows a Mamdani fuzzy inference system (FIS) that is created This FIS has five inputs, which are Pred1, Pred2, Pred3, Pred4, and Pred5, the range 0 to 1.4., the output is called prediction and the range goes from 0 to 1.4 and is granulated into two MFs "Low", "High", as linguistic values.

The Fuzzy system rules are as follows (as shown in Figure 9), since the fuzzy system has 5 input variables with two MFs and one output with two MFs, therefore the possible number of rules is 32.

5 Experimentation Results

This part presents the experiments of the optimization of the ERNN with the GA, as well the integration type-1 and IT2FS.

In addition, we present graphs of real data against predicted and results of the prediction for each of the experiments of Mackey Glass benchmark, Mexican Stock Exchange, Dow Jones, and Exchange Rate of US Dollar/Mexican Pesos time series. The following table shows the results

ISSN 2007-9737

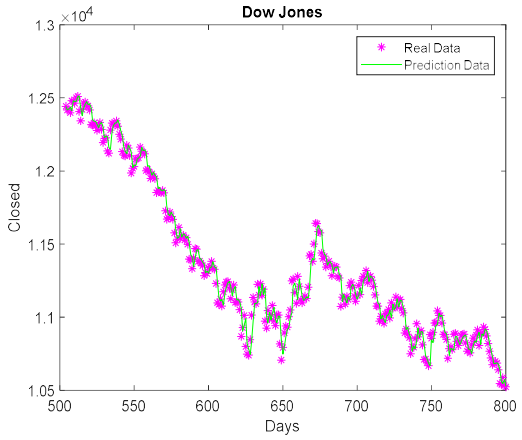


Fig. 13. Plot of real data against predicted data for the type-1 fuzzy system of DJ

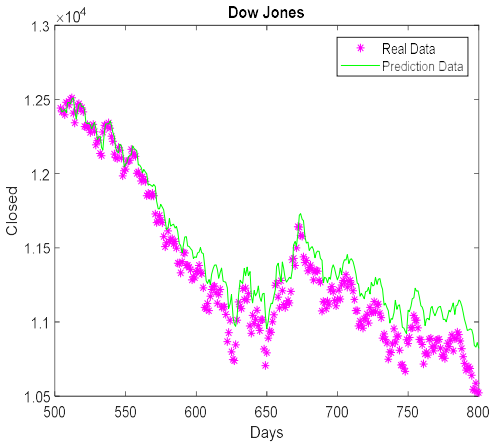


Fig. 14. graph of real data against predicted data for the T2FS of DJ

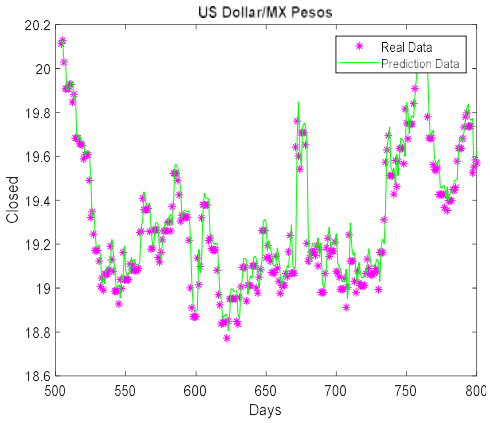


Fig. 15. Plot of real data against predicted data for the type-1 fuzzy system of Dollar

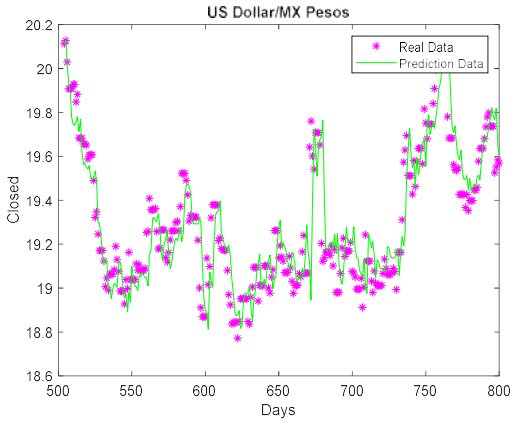


Fig. 16. Plot of real data against predicted data for the T2FS of Dollar data.

of the genetic algorithm, where the best architecture of the ERNN is shown in row number 7 of Table 3 for the Mackey-Glass time series.

Table 4 illustrates the results of the type-1 FS integration for the optimized ERNN, where the obtained result is of experiment number 8, with a prediction error: 0.1667. Figure 9 represents the plot of the real data against predicted data for the type-1 fuzzy system for the Mackey-Glass time series. Table 5 and Figure 10 represent the prediction of the time series using the IT2FS for the Mackey-Glass time series, respectively. Table 6 shows the results of the genetic algorithm, where the best architecture of the ERNN is shown in row number 2 of Table 6 for the Mexican Stock Exchange time series.

Table 7 shows the results of the type-1 FS integration for the optimized ERNN, where the result obtained is of experiment number 4, with a prediction error: 0.3270. Figure 11 represents the plot of the real data against predicted data for the type-1 fuzzy system for the Mexican Stock Exchange time series.

Table 8 and Figure 12 illustrates the prediction of the time series using the IT2FS for the Mexican Stock Exchange time series.

Table 9. Genetic algorithm results for the RNN for the DJ

Evolutions	Gen.	Ind.	Pm	Pc	Num. Modules	Num. Layers	Num. Neurons	Duration	Prediction Error
1	100	100	0.07	0.6	5	3	24,30,9 13,26,15 17,22,25 13,25,30 1,21,30	18:07:07	0.0023472
2	100	100	0.05	0.7	5	5	5,8,1 12,16,16 9,7,12 6,20,15 1,2,7	16:25:18	0.0018525
3	100	100	0.07	0.5	5	3	22,2,1 14,24,25 9,12,23 8,21,10 1,17,22	17:08:49	0.028711
4	100	100	0.03	0.4	5	2	6,29 6,2 5,17 8,9 2,6	19:09:55	0.0022167
5	100	100	0.09	0.9	5	3	15,16,1 5,10,4 3,16,4 6,30,6 7,30,30	18:05:13	0.0021315
6	100	100	0.05	0.5	5	3	30,11,4 30,18,17 13,9,29 7,21,5 2,1,14	17:20:12	0.0026022
7	100	100	0.09	0.9	5	3	8,1,15 6,22,11 4,8,22 6,30,27 1,10,2	17:22:17	0.0025405
8	100	100	0.09	1	5	3	7,8,1 6,30,6 15,30,8 12,30.27,3 3,30,30	18:23:24	0.0022419
9	100	100	0.04	0.7	5	3	9,9,26 9,15,22 13,28,8 1,19,4 2,10,24	19:20:14	0.002269
10	100	100	0.03	0.7	4	3	16,16,1 4,15,4 6,27,7 10,30,20 17,22,29	19:02:31	0.002593

Table 10. Results of type-1 FS of DJ

Test	Prediction Error with Type-1 Fuzzy Integration
1	0.11343
2	0.12376
3	0.26920
4	0.14675
5	0.10567
6	0.22561
7	0.17888
8	0.18886
9	0.26922
10	

Table 11. Results of type-2 FS of DJ

Test	Prediction Error 0.3 Uncertainty	Prediction Error 0.4 Uncertainty	Prediction Error 0.5 Uncertainty
1	0.0188	0.0188	0.0172
2	0.0117	0.0117	0.0145
3	0.0156	0.0156	0.0164
4	0.0138	0.0176	0.0137
5	0.0178	0.0180	0.0185
6	0.0217	0.0224	0.0243
7	0.0169	0.0170	0.0152
8	0.0163	0.0163	0.0165
9	0.0156	0.0154	0.0151
10	0.0208	0.0208	0.0218

Table 9 shows the results of the genetic algorithm, where the best architecture of the ERNN is shown in row number 2 of Table 9 for the Dow Jones time series.

Table 10 shows the results of the integration type-1 FS for the optimized ERNN, where the result obtained is of experiment number 5, with a prediction error: 0.10567 and Figure 13 represents the plot of real data against predicted data for the type-1 fuzzy system.

Table 11 and Figure 14 represent the prediction of the time series using the IT2FS for the Dow Jones time series.

Table 12 shows the results of the genetic algorithm, where the best architecture of the ERNN is shown in row number 4 of Table 12, for the US/Dollar Mexican Pesos time series.

Table 13 illustrates the results of the type-1 FS integration for the optimized ERNN, where the result obtained is of experiment number 4, with a prediction error: 0.113072 and Figure 15 represents the plot of real data against predicted data for the type-1 fuzzy system, for the US/Dollar Mexican Pesos time series.

Table 14 and Figure 16 illustrate the prediction of the time series using the IT2FS for the US/Dollar Mexican Pesos time series.

5.1 Comparison of Results

Comparisons were made with the paper called: “A New Method for Type-2 Fuzzy Integration in Ensemble Neural Networks Based on Genetic Algorithms*”,* where the same data from the series of the Mackey-Glass were used.

In this case, we obtained that recurrent neural networks are better for predicting data from this series since there is a significant difference in the results, as they are better with the recurrent neural than with ensemble neural network,

Table 12. Genetic algorithm results for the RNN of Dollar

Evolutions	Gen.	Ind.	Pm	Pc	Num. Modules	Num. Layers	Num. Neurons	Duration	Prediction Error
1	100	100	0.07	0.6	5	1	1 1 9 6 1	02:01:34	0.00213
2	100	100	0.05	0.7	5	3	11,26,30 4,23,14 1,2,13 12,2,6 1,16,30	07:35:18	0.0018864
3	100	100	0.07	0.5	5	1	1 1 3 11 1	02:21:04	0.0029528
4	100	100	0.03	0.4	5	1	3 6 2 19 3	02:09:55	0.0018685
5	100	100	0.09	0.9	5	1	1 1 1 6 1	02:12:04	0.0030438
6	100	100	0.05	0.5	5	5	5,25,24 7,24,9 1,29,22 4,25,30 1,23,13	02:53:46	0.0020584
7	100	100	0.09	0.8	5	1	2 13 4 6 2	01:54:24	0.0021801
8	100	100	0.09	1	5	1	1 1 1 7 1	01:34:18	0.0021431
9	100	100	0.04	0.7	5	1	1 1 1 2 1	01:38:38	0.0022053
10	100	100	0.03	0.7	5	1	2 12 5 8 1	01:36:38	0.0025446

Therefore, we use a significance of 90% and according to the results obtained and we can say that there is significant improvement with the ensemble neural network, as is summarized in Table 15. Comparisons were also made with the paper called: “Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange”, where the same data from the series of the Mexican stock exchange were used.

Table 13. Results of type-1 FS of Dollar data

Test	Prediction Error with Type-1 Fuzzy Integration
1	0.114981
2	0.113070
3	0.115000
4	0.113072
5	0.114809
6	0.113190
7	0.119767
8	0.115691
9	0.113076
10	0.114352

Table 14. Results of type-2 FS of Dollar data

Test	Prediction Error 0.3 Uncertainty	Prediction Error 0.4 Uncertainty	Prediction Error 0.5 Uncertainty
1	0.2341	0.2215	0.3972
2	0.2217	0.2056	0.3779
3	0.2118	0.2019	0.3888
4	0.1979	0.1845	0.3985
5	0.1722	0.1944	0.3612
6	0.1922	0.2251	0.3758
7	0.2012	0.2252	0.3763
8	0.2212	0.2019	0.3794
9	0.2132	0.2313	0.3590
10	0.2055	0.1903	0.3674

Table 15. Results of comparison of the Mackey-Glass

Time Series	N(RNN)	N(ENN)	Value(T)	Value(P)
Dow Jones	30	30	-0.5091	0.0694

Table 16. Results of comparison of the Mexican Stock Exchange

Time Series	N(RNN)	N(ENN)	Value(T)	Value(P)
Mexican Stock Exchange	30	30	-9.0370	0.000

Table 17. Results of comparison of the Dow Jones

Time Series	N(RNN)	N(ENN)	Value(T)	Value(P)
Mackey-Glass	30	30	1.3732	0.090

We obtained that recurrent neural network is better for predicting data from this series since there is a significant difference in the results are better the recurrent neural that with ensemble neural network, Therefore, we use a significance of 99% and according to the results obtained we can

say that there is significant improvement with the ensemble neural network, as summarizes in Table 16.

Comparisons were made with the paper called: "Optimization of Ensemble Neural Networks with Type-2 Fuzzy Integration of Responses for the Dow Jones Time Series Prediction". where the same data from the series of the Dow Jones were used and we obtained that recurrent neural networks are better for predicting data from this series since there is a no significant difference in the results are better the recurrent neural that with ensemble neural network, Therefore, we use a significance of 90% and according to the results obtained we can say that there is significant improvement with the ensemble neural network, as is summarized in Table 17.

6 Conclusions

In this work the design, implementation, and optimization of ensemble recurrent neural network for the prediction time are presented.

The chosen algorithm for this optimization was the GA, with which a total of 30 different experiments were made.

Comparisons were made with previously carried out works, in this way it can be said that genetic algorithms are an optimization technique that gives good results for the forecast of the time series. The main contribution in this paper was the creation of the new model of recurrent neural networks presented in this document that has shown good results since they are effective for the prediction of time series.

A hierarchical GA was applied to optimize the architecture of the RNN, in terms of parameters (NM, NL NN), to find better architecture and the time series error. The integration of the network responses was done with a type-1 and T2FS, to obtain the prediction error of the proposed time series, such as Mackey Glass benchmark, Mexican Stock Exchange, Dow Jones, and Exchange Rate of US Dollar/Mexican Pesos time series.

Analyzing the results, we can say that the combination of these intelligent computing techniques generates excellent results for this type of problem since the recurrent neural networks analyze the data of time series, the Genetic algorithms perform optimization and they helped us find the best architecture of the RNN, as well as to obtain the best solution to the proposed problem.

As future work we plan to perform optimization of the recurrent neural network with another optimization method, and make comparisons of the type-1 and type-2 fuzzy systems. We will also consider other complex time series to test the ability of our method for predicting complex time series.

Acknowledgments

We would like to express our gratitude to the CONACYT and Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

1. **Apaydin, H., Feizi, H., Sattari, M.T., Colak, M.S., Shamshirband, S., Chau, K.W., (2020).** Comparative Analysis of Recurrent Neural Network Architectures for Reservoir Inflow Forecasting. Water, Vol. 12, No. 5, pp. 1–18, DOI: 10.3390/w12051500.
2. **Chang, B., Chen, M., Haber, E., Chi, E.H. (2019).** Antisymmetric RNN: Adynamical system view on recurrent neural networks. International Conference on Learning Representations, pp. 2–6. DOI: 10.48550/arXiv.1902.09689.
3. **Brockwell, P.D., Davis, R.A. (2002).** Introduction to Time Series and Forecasting. Springer-Verlag, New York, pp. 259316. DOI: 10.1007/0-387-21657-X_8.
4. **Castillo, O., Hidalgo, D., Cervantes, L., Melin, P., Martínez, R. (2020).** Fuzzy Parameter Adaptation in Genetic Algorithms for the Optimization of Fuzzy Integrators in Modular Neural Networks for Multimodal Biometry. Computación y Sistemas, Vol. 24 No. 3, pp. 1093–1105. DOI: 10.13053/cys-24-3-3329.

5. **Castillo, O., Amador-Angulo, L., (2018).** Generalized type-2 fuzzy logic approach for dynamic parameter adaptation in bee colony optimization applied to fuzzy controller design. Information Sciences, Vol. 460-461, pp. 476–496. DOI: 10.1016/j.ins.2017.10.032.
6. **Castillo, O., Melin, P. (2007).** Comparison of Hybrid Intelligent Systems Neural Networks and Interval Type-2 Fuzzy Logic for Time Series Prediction. Proceedings IJCNN 2007, pp. 3086–3091.DOI: 10.1109/IJCNN.2007. 4371453.
7. **Castillo, O., Melin, P. (2002).** Hybrid intelligent systems for time series Prediction using neural networks, fuzzy logic, and fractal theory. Neural Networks, IEEE Transactions on, Vol. 13, No. 6. pp. 1395–1408. DOI: 10.1109/TNN.2002.804316.
8. **Castillo, O., Melin, P. (2001).** Simulation and Forecasting Complex Economic Time Series Using Neural Networks and Fuzzy Logic. Proceedings of the International Neural Networks Conference Vol. 3, pp. 1805–1810. DOI: 10.1109/IJCNN.2001.938436.
9. **Castillo O., Melin, P. (2001).** Simulation and Forecasting Complex Financial Time Series Using Neural Networks and Fuzzy Logic. Proceedings the IEEE the International Conference on Systems, Man and Cybernetics Vol. 4, pp. 2664–2669. DOI: 10.1109/ICSMC. 2001.972967.
10. **Castillo O., Melin, P. (2008).** Type-2 Fuzzy Systems, Type-2 Fuzzy logic Theory and Application. Springer, pp. 30–43. DOI: 10.1109/GrC.2007.118.
11. **Castillo O., Melin, P. (2007).** Comparison of Hybrid Intelligent Systems, Neural Networks and Interval Type-2 Fuzzy Logic for Time Series Prediction. Proceedings IJCNN, pp. 3086–3091. DOI: 10.1109/IJCNN.2007. 4371453.
12. **Castro, J.R., Castillo, O., Melin P., Mendoza, O., Rodríguez-Diaz, A. (2011).** An Interval Type-2 Fuzzy Neural Network for Chaotic Time Series Prediction with cross-validation and akaike test. Soft Computing for Intelligent Control and Robotics, Vol. 318, pp. 269–285. DOI: 10.1007/978-3-642-15534-5_17.
13. **Smith, C., Jin, Y. (2014).** Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction. Neurocomputing, Vol. 143, pp. 1–10. DOI: /10.1016/j.neucom.2014.05.062.
14. **Cowpertwait, P., Metcalfe, A. (2009).** Time Series, Introductory Time Series with R. Springer Dordrecht Heidelberg London New York, pp. 2–5.
15. **Dow Jones Company. (2021).** https://www.dowjones.com.
16. **Fekri, M.N., Patel, H., Grolinger, K., Sharma, V. (2021).** Deep learning for load forecasting with smart meter data: Online Adaptive Recurrent Neural Network. Applied Energy, Vol. 282, pp. 1–17, DOI: 10. 1016/j.apenergy.2020.116177.
17. **Gaxiola, F., Melin, P., Valdez, F., Castillo O., (2014).** Interval Type-2 Fuzzy Weight Adjustment for Backpropagation Neural Networks with Application in Time Series Prediction. Information Sciences Vol. 260, No. 1, pp. 1–14. DOI: 10.1016/j.ins.2013.11.006.
18. **Goldberg, D. (1989).** Genetic Algorithms in search, optimization and machine learning. Addison Wesley.
19. **Hewamalage, H., Bergmeir, C., Bandara, K. (2021).** Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. Vol. 37, No. 1, pp. 388–427. DOI: 10.1016/j.ijforecast.2020.06.008.
20. **Triebe, O., Hewamalage, H., Pilyugina, P., Laptev, N., Bergmeir, C., Rajagopal, R. (2021).** NeuralProphet: Explainable Forecasting at Scale. arXiv:2111.15397.
21. **Jerome, T., Connor, R., Douglas M. (1994).** Recurrent neural networks and robust time series Prediction, IEEE Transactions on Neural Networks, Vol. 5, No. 2, pp. 240–254. DOI: 10.1109/72.279188.
22. **Wen, J., Wu, L., Chai, J. (2020).** Paper Citation Count Prediction Based on Recurrent Neural Network with Gated Recurrent Unit. IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC), pp. 303–306. DOI: 10.1109/ICEIEC49280.2020.9152330.

23. **Jilani, T.A., Burney, S.M.A. (2008).** A refined fuzzy time series model for stock market forecasting. Physica-A-Statistical mechanics and its applications, Vol. 387, No. 12, pp. 2857–2862. DOI: 10.1016/j.physa.2008.01.099.
24. **Wen, J., Wu, L., Chai, J. (2020).** Paper Citation Count Prediction Based on Recurrent Neural Network with Gated Recurrent Unit. IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC), pp. 303–306.
25. **Karnik, N., Mendel, J.M. (1999).** Applications of type-2 fuzzy logic systems to forecasting of time-series. Information Sciences, Vol. 120, No. 1–4, pp. 89-111. DOI: 10.1016/S0020-0255(99)00067-5.
26. **Karnik, N., Mendel, J.M. (2001),** Operations on type-2 set. Fuzzy Set and Systems, Vol. 122, No. 2, pp. 327–348. DOI: 10.1016/S0165-0114(00)00079-8.
27. **Lu, S., Zhang, Q., Chen, G., Seng, D. (2021),** A combined method for short-term traffic flow prediction based on recurrent neural network. Alexandria Engineering Journal. Vol. 60, No. 1, pp. 87–94, DOI: 10.1016/j.aej.2020.06.008.
28. **Mackey, M.C. (2007).** Adventures in Poland: having fun and doing research with Andrzej Lasota. Mat. Stosow, pp. 5–32.
29. **Mackey, M.C., Glass, L. (1997).** Oscillation and chaos in physiological control systems. Science, Vol. 197, No. 4300, pp. 287–289. DOI: 10.1126/science.267326.
30. **Man, K., Tang, K., Kwong, S. (1998).** Genetic Algorithms and Designs, Introduction, Background and Biological Background. Springer-Verlag London Limited, pp. 1–62.
31. **Melin, P., Castillo, O., Gonzalez, S., Cota, J., Trujillo, W.L., Osuna P. (2007).** Design of Modular Neural Networks with Fuzzy Integration Applied to Time Series Prediction. Springer Berlin / Heidelberg, Vol. 41, pp. 265–273. DOI: 10.1007/978-3-540-72432-2_27.
32. **Melin, P., Soto, J., Castillo, O., Soria, J. (2012).** A new Approach for Time Series Prediction Using Ensembles of ANFIS Models. Expert Systems with Applications, (2011).
33. **Mendel, J., (2001).** Uncertain Rule-Based Fuzzy Logic Systems, Introduction of new directions. Prentice-Hall, Inc., Vol. 2, No. 1, pp. 72–73. DOI: 10.1109/MCI.2007.357196.
34. **Mencattini, A., Salmeri, M.,S. Mertazzoni, B, Lojacono, R., Pasero, E., Moniaci, W., (2005).** Local Meteorological Forecasting by Type-2 Fuzzy Systems Time Series Prediction. CIMSA - IEEE International Conference on Computational Intelligence for Measurement Systems and Applications Giardini Naxos, Italy, pp. 20–22.
35. **Mexican Bank Database, (January 10, 2021),** https://www.banxico.org.mx
36. **Min, H.,Jianhui, X., Shiguo X., Fu-Liang Y., (2004),** Pred of chaotic time series based on the recurrent predictor neural network. IEEE, Vol. 52, No. 12, pp. 3409–3416, DOI: 10.1109/TSP.2004.837418.
37. **Mikolov, T., Kombrink, S., Burget, L., Černocký, J., Khudanpur, S. (2011).** Extensions of recurrent neural network language model. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 552–5531. DOI: 10.1109/ICASSP.2011.5947611.
38. **Olivas, F., Valdez, F., Melin, P., Sombra, A., Castillo, O. (2019).** Interval type-2 fuzzy logic for dynamic parameter adaptation in a modified gravitational search algorithm. Information Sciences Vol. 476, pp. 159–175.
39. **Pang, Z., Niu, F., O'Neill, Z. (2020).** Solar radiation prediction using recurrent neural network and artificial neural network: A case study with comparisons. Renewable Energy, Vol. 156, pp. 279–289. DOI: 10.1016/j.renene.2020.04.042.
40. **Valdez, F., Castillo, O., Peraza, C. (2020).** Fuzzy Logic in Dynamic Parameter Adaptation of Harmony Search Optimization for Benchmark Functions and Fuzzy Controller. International Journal of Fuzzy Systems, Vol. 22, pp. 1198–1211 DOI: 10.1007/s40815-020-00860-7.
41. **Petnehazi, G. (2019).** Recurrent Neural Networks for Time Series Forecasting. University of Debrecen, pp. 1–22.

42. Pulido, M., Melin, P. (2021). Ensemble Recurrent Neural Networks for Complex Time Series Prediction with Integration Methods", Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms Theory and Applications. Studies in Computational Intelligence, Vol 940, pp. 71–83 DOI: 10.1007/978-3-030-68776-2_4 pp. 71-83.

43. Pulido, M., Mancilla, A., Melin, P. (2009). An Ensemble Neural Network Architecture with Fuzzy Response Integration for Complex Time Series Prediction. In: **Castillo, O., Pedrycz, W., Kacprzyk, J.,** editors, Evolutionary Design of Intelligent Systems in Modeling, Simulation and Control. Studies in Computational Intelligence, Springer, Berlin, Heidelberg. Vol 257. 85–110. DOI: 10.1007/978-3-642-04514-1_6.

44. Pulido, P., Melin, P., Castillo, O. (2014). Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange. Information Sciences, Vol. 208, pp. 188–204. DOI: 10.1016/j.ins.2014.05.006.

45. Melin, P., Puldo, M. (2014). Optimization of Ensemble Neural Networks with Type-2 Fuzzy Integration of Responses for the Dow Jones Time Series Prediction. Intelligent Automation & Soft Computing, Vol. 20, No. 3, pp, 403–418. DOI: 1 0.1080/10798587.2014.893047.

46. Pulido, M., Melin, P. (2021). Comparison of Genetic Algorithm and Particle Swarm Optimization of Ensemble Neural Networks for Complex Time Series Prediction. In: **Melin, P., Castillo, O., Kacprzyk, J.,** editors, Recent Advances of Hybrid Intelligent Systems Based on Soft Computing. Studies in Computational Intelligence, Vol 915. Springer, Cham. DOI: 10.1007/978-3-030-58728-4_3.

47. Rohitash, C., Mengjie, Z. (2012). Cooperative coevolution of Elman recurrent neural networks for chaotic time series Prediction. Neurocomputing, Vol. 86, pp. 116–123. DOI: 10.1016/j.neucom.2012.01.014.

48. Sharkey, A.J.C. (1999). Combining artificial neural nets: ensemble and modular multi-net systems. Perspectives in Neural Computing, Springer-Verlag, London.

49. Sharkey, A.J.C. (1996). One combining Artificial of Neural Nets. Department of Computer Science University of Sheffield, U.K. Vol. 8, No. 3-4, DOI: 10.1080/09540099611 6785.

50. Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena, Vol. 404, pp.1–28. DOI: 0.1016/j.physd.2019.132306.

51. Slawek, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. International Journal of Forecasting, Vol. 36, No. 1, pp. 75–85. DOI: 10.1016/j.ijforecast. 2019.03.017.

52. Sollich, P., Krogh, A. (1996). Learning with ensembles: how over-fitting can be useful, **D.S. Touretzky, M.C. Mozer, M.E. Hasselmo (Eds.),** Advances in Neural Information Processing Systems 8, Denver, CO, MIT Press, Cambridge, MA, pp.190–196.

53. Soto, J., Melin, P., Castillo, O. (2018). A New Approach for Time Series Prediction Using Ensembles of IT2FNN Models with Optimization of Fuzzy Integrators. International Journal of Fuzzy Systems, Vol. 20, pp. 701–728. DOI: 10.1007/s40815-017-0443-6.

54. Soto J., Melin, P., Castillo, O. (2014). Time series prediction using ensembles of ANFIS models with genetic optimization of interval type-2 and type-1 fuzzy integrators. International Journal of Hybrid Intelligent system Vol. 11, No. 3, pp. 211–226. DOI: 10.3233/HIS-140196.

55. Sudipto, S., Raghava, G.P.S. (2006). Prediction of continuous B-cell epitopes in an antigen using recurrent neural network. National Library of Medicine, Vol. 65, No. 1, pp. 40–48. DOI: 10.1002/prot.21078

56. Walid, A. (2016), Recurrent Neural Network for Forecasting Time Series with Long Memory Pattern. Journal of Physics: Conference Series, Vol. 824, pp. 1–8. DOI: 10.1088/1742-6596/824/1/012038.

57. Wei, X., Zhan, L., Yang, H.Q., Zhang, L., Yao Y.P. (2020). Machine learning for pore-water

pressure time-series prediction: Application of recurrent neural networks, Geoscience Frontiers, Vol. 12, No. 1, pp. 453–467. DOI: 10.1016/j.gsf.2020.04.011.

58. **Whitley, L.D. (2017).** Foundations of Genetic Algorithms 2. Morgan Kaufman Publishers, (1993), pp.332.
59. **Yao, Q., Dongjin, S., Haifeng, C., Wei, C., Guofei, J., Garrison, C. (2017).** A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction, Computer science, cornell University, pp. 1–7.
60. **Zadeh, L.A. (1965).** Fuzzy Sets. Information and control. Vol. 8, pp. 338–353.
61. **Zhan, J., Man, K.F. (1998).** Time series Prediction using recurrent neural network in multi-dimension embedding phase space. IEEE International Conference on Systems, Man and Cybernetics, Vol. 2 pp. 1868–1873 DOI: 10.1109/ICSMC.1998.728168.
62. **Zhang, D., Peng, Q., Lin, J., Wang, D., Liu, X., Zhuang, J. (2019).** Simulating Reservoir Operation Using a Recurrent Neural Network Algorithm. Water, Vol. 11 No. 4, pp. 1–18, DOI: 10.3390/w11040865.
63. **Zhang, J.S., Xio, X.C. (2000).** Predicting Chaotic Time Series Using Recurrent Neural Network. Published under licence by IOP Publishing Ltd, pp. 88–90.
64. **Zhou, Y., Guo, S., Xu, C.Y., Chang, F.J., Yin, J., (2020),** Improving the Reliability of Probabilistic Multi-Step-Ahead Flood Forecasting by Fusing Unscented Kalman Filter with Recurrent Neural Network. Water, Vol. 12, No. 2, pp. 1–15, DOI: 10.3390/w12020578.

Article received on 10/06/2021; accepted on 16/11/2021.
Corresponding author is Patricia Melin.

ISSN 2007-9737

Hierarchical Decision Granules Optimization Through The Principle of Justifiable Granularity

Raúl Navarro-Almanza, Mauricio A. Sanchez, Juan R. Castro,
Olivia Mendoza, Guillermo Licea

Universidad Autónoma de Baja California,
Facultad de Ciencias Químicas e Ingeniería,
Mexico

{rnavarro, mauricio.sanchez, jrcastror, omendoza, glicea}@uabc.edu.mx

Abstract. Interpretable Machine Learning (IML) aims to establish more transparent decision processes where the human can understand the reason behind the models' decisions. In this work a methodology to create intrinsically interpretable models based on fuzzy rules is proposed. There is a selection to identify the rule structure by extracting the most significant elements from a decision tree by the principle of justifiable granularity. There are defined hierarchical decision granules and their quality metrics. The proposal is evaluated with ten publicly available datasets for classification tasks. It is shown that through the principle of justified granularity, rule-based models can be greatly compressed through their fuzzy representation, not only without significantly losing performance but even with compression of 40% it manages to exceed the performance of the initial model.

Keywords. Granular computing, neuro-fuzzy, Sugeno, hierarchical decision granules, interpretable machine learning.

1 Introduction

Machine learning models are ubiquitous nowadays; They are involved in many activities of daily life in which people are aware or unaware of their use. It is essential to know how and why the models provide a particular output and how they can be made more fair and secure, especially in critical applications. Interpretable Machine Learning (IML) strives to construct a bridge between the learned model and human understanding.

There are various ways to achieve model interpretability by applying: i) intrinsic interpretable models, ii) regularization techniques, and iii) posthoc explanation techniques. One trend is to build surrogate models (lower complexity) than the original one and more interpretable to understand the decision process, such as rule-based models. [7, 9, 48].

This work takes advantage of the inherent interpretability that brings Fuzzy Inference Systems (FIS). FIS models are interpretable since they are rule-based models and are described linguistically. The antecedent is represented by fuzzy variables and sets, proposed by Zadeh [59]. The inference process of those systems is performed through fuzzy reasoning, which aims to represent human perception and their inference mechanism under uncertainty.

An interesting characteristic of FIS is that their knowledge representation is composed of IF-THEN rules, where their antecedents and consequents are in natural language. For example, a proposition can be "`Temperature` is `hot`", where `Temperature` is related to an input attribute and `hot` to the set which partially belong. As was described, this formal notation potentially brings an intrinsic high interpretability degree if their components are well-defined [32].

The FIS are used in a variety of application domains in ML context, such as medical [14, 19, 24, 41, 57, 35], robotics [15, 1], decision making [60, 12, 3].

Often this FIS modeling is data-driven, usually by some unsupervised technique (e.g., clustering) to discover their input partition, membership functions parameters, and rule structure. An important issue is that as the number of attributes (input dimension) and fuzzy sets increases, it becomes more susceptible to presenting a combinatorial problem in the rule-finding process. In this approach, we construct the initial rule structure through a decision tree model, which is further transformed to the fuzzy space domain and adapted to a Sugeno-type architecture described in the further sections.

As the input dimension space and feature interaction complexity increases, the resulted decision tree gets deeper; therefore, it becomes pruned to be hardly interpretable. For each *leaf* node or decision node in the tree, it can extract an IF-THEN decision rule. To tackle the problem of a high number of decision rules is conducted a post-pruning process. The pruning process is carried on by selecting the most relevant elements (antecedents and decision rules).

Granular Computing (GrC) aims to form meaningful Information Granules, which represent a collection of objects abstraction and relate them by some similarity in a hierarchical manner [5], which allows creating semantically richer structures [67]. GrC is inspired by how the human brain works, processing information abstractly at the required level to resolve a given task. The data is organized by some of their characteristics in a hierarchical way to form a granule, a formal representation of this structure with two essential properties, specificity, and coverage.

Specificity is related to the granule representation; the higher specificity is, the less ambiguous it is, and humans can easily understand it. On the other hand, coverage is related to the proportion of individuals designated by the granule.

Intuitively, a well-formed information granule should have higher values in both specificity and coverage. However, these properties are commonly in conflict, as the higher the specificity is, the lower is the coverage; an optimization process usually conducts the granule allocation.

This work proposes a data-driven method to construct a fuzzy rule-based system using the principle of justifiable granularity for selecting the most relevant knowledge base elements, according to the trade-off between specificity and coverage values. This discrimination is conducted over the rules formed by decision tree models, allowing the construction of a variable model complexity useful in IML.

The main contributions of this work are:

— Definition of a Sugeno-type neuro-fuzzy model for classification tasks that leads to the straightforward interpretation of the decision process.

— Characterization of hierarchical Information Granules in decision-set context.

— Definition of hierarchical specificity and coverage metrics for optimization of graph-based entities.

— Data-driven methodology for establishing fuzzy inference system structure by decision tree rule extraction and selecting the most relevant elements by following the principle of justifiable granularity.

In the following sections correspond to: a brief description of the relevant theory, section 2. In section 3 is reviewed the related work. In section 4 is described the hierarchical Information Granules and their generation. In section 5, the neuro-fuzzy model is described. In the section 6 is shown the followed methodology and experiment setup to evaluate the proposal. The results and conclutions are in sections 7 and 8 respectively.

2 Background

The proposed methodology comprehends three main areas: i) Fuzzy systems for establishing natural language interface in the form of fuzzy rules; ii) Decision tree-based models for initial structure rule discovery in the data space domain. iii) Granular computing, aiming at the well-formed Information Granule for rule-based knowledge model representation following the principle of justifiable granularity.

ISSN 2007-9737

2.1 Fuzzy Systems

Fuzzy Systems are rule-based models that use fuzzy logic to conduct the reasoning process. Fuzzy logic was proposed by Zadeh [58] as an approach to represent computable human perception through words.

This reasoning system type offers greater explainability and is widely used in the Machine Learning field due to its ability to process linguistic information [13].

These systems are broadly used on various domain applications such as control, medical, aerospace and environmental applications [15,30,54,49,29,9].

The knowledge base modeling can be built either manually by experts, or designed automatically, usually by clustering techniques.

Zadeh's fuzzy rule has the following structure: IF *Temperature* is *hot* THEN *Cooling* is *high*.

The antecedents and consequents are as shown formed by linguistic variables (*Temperature* and *Cooling*), the values of these variables are linguistic values (*hot* and *high*), which their meanings are easily understanding by humans.

The Zadeh's linguistic variable [59] is characterized by a quintuple $(x, T(x), X, G, M)$, in which x is the name of the variable; $T(x)$ is the term set of x, linguistic terms; X is the universe of discourse; G is a synthetic rule which generates linguistic terms in $T(x)$; M is a semantic rule which associates each linguistic value A its meaning $M(A)$, where $M(A)$ denotes a fuzzy set in A.

A fuzzy set A in X domain is defined as a set of ordered pairs (equation 1):

$$A = \{(x, \mu_A(x)) | x \in X\}, \tag{1}$$

where $\mu_A(x) \in [0, 1]$ is the membership function that represents the human perception in form of membership degree in de universe of discourse X.

2.2 Decision Trees

Decision trees are graph-based Machine Learning models for classification and regression tasks. The model constructs a tree in which their non-terminal nodes perform splits in the input data space (in the context of Machine Learning). The splitting process is sequential until it reaches the leaf nodes (terminal nodes). The evidence provides an output label or probability of belongingness to some class (in classification tasks).

This tree construction relies on subsequent partitioning in the data inputs space by selecting the best feature and value to split. There are many criteria for select the best split candidate. The main idea is to achieve the best purity, which means that the residual data after each split belongs to only one class; until this goal is not reached, new nodes are added to the tree.

The relevant hyper-parameters in this context for regularizing the model are the number of features searched in each partition node; minimum elements belonging to a node to consider creating a branch; criteria to measure the quality of a split; the maximum depth that can have the decision tree.

Once the tree is created, it can be traverse through its branches until each lead node is reached; every node condition in the path can be extracted to form an antecedent and consequent part of an IF-THEN rule. Thus, for every leaf node, a rule can be constructed.

In this work, the performed input space partitioning by the decision tree model is used to define the initial structure of the knowledge base of the FIS. The selection of the most relevant rules elements is conducted via optimization by the principle of justifiable granularity.

2.3 Granular Computing

Granular Computing is a paradigm inspired by how the human brain performs different levels of entities' characteristics abstraction and uses those to make decisions. The main particle in this paradigm is called *Information Granule*. These granules can be regarded as a collection of objects hierarchically that exhibit similarities among them.

There is not an specific formalization to define an Information Granule, they can be described by a huge variety of different representation, such as: interval sets [21, 40],rough sets [26, 61, 50, 49, 69], fuzzy sets [37, 4, 36, 63, 62], probabilistic sets [42, 53], possibility sets [65, 46], neural networks [34, 20, 64, 51, 28, 17]. GrC is a unified framework of techniques, methodologies, and theories for the formalization, construction, and manipulation of Information Granules; it brings a coherent environment to work with abstract object representation [5].

Some techniques for building fuzzy information representations are inspired by granular computing, such in [43] where a method is proposed to find the right cluster size concerning the data context; in [8] is proposed a generalized Type-2 fuzzy control model that uses granularity to divide the global model by simpler models.

2.3.1 Principle of Justifiable Granularity

The fundamental idea of principle of justifiable granularity is to form meaninful Information Granules based on experimental evidence (data), following two general criterias: *coverage* and *specificity* [38].

The coverage is the numeric evidence that supports the Information Granule. The intention is to form/discover granules with the more substantial experimental evidence that supports its formulation. On the contrary, the specificity is related to the granule's well-formed; the smaller the Information Granule is, the better. The ideal is to form meaningful Information Granules with the higher coverage and specificity as possible.

These two requirements are in conflict. In a basic formulation, the granule A been represented as an interval $[a, b]$. As higher the range is, the more expected experimental support (cardinality, showed as $card(\cdot)$) it gets ($cov(A) = card(\{x_i | x_i \in [a, b]\})$); at the same time, the specificity decreases, considering the range as the specificity ($sp(A) = |b - a|$).

To find the best meaningful Information Granule, this contrary behavior between the criteria of coverage and specificity can be defined as a multi-objective optimization problem for the maximization of the composite multiplicative index. Given a set of design parameters θ for the Information Granules, it must find the best values for θ that maximizes the equation 2:

$$A^*_\theta = \underset{\theta}{\operatorname{argmax}} \quad cov(A_\theta) * sp(A_\theta). \tag{2}$$

The principle of justifiable granularity allows finding the best well-formed granule. In the fuzzy logic context, it has been used to define fuzzy information granules, such in [33] where is used to construct IT2 Fuzzy Memberships functions.

The proposed method in this work applies the principle of justifiable granularity to compress decision sets and improve their interpretability.

3 Related Work

GrC has been used in Machine Learning problems as a way to define semantic richer data representation to build models with missing information [25], prototype forming for descriptors of facial expressions [55]. To establish initial neural network architecture for further optimization [39]. Furthermore, this framework has been used to overcome the limitations of existing Machine Learning models related to data quality [22, 10, 6], interpretability, domain adaptation for regression tasks [22], and dimensionality reduction [2, 16, 52].

In [31] is discussed the importance of adopting the GrC paradigm in rule-based systems as a way to improve interpretability. The principle of justifiable granularity has been used to discover robust information clusters in the context of data-driven system modeling [68]. There are various works in the context of GrC which support the hypothesis of more robust generation rule-based systems [44, 54], rule reduction using complex fuzzy measures with GrC [47]. Also, the hierarchical representation of granule modeling has been used to solve hierarchical classification problems [23], for building interpretable models in data stream learning environments [27].

The use of GrC in the context of Machine Learning comprehends the discovery of the Information Granules in the data space domain and forms them by some formal description, e.g., intervals, fuzzy sets, rough sets, hyperboxes.

Some recent approaches in GrC adopt cognitive science perspectives and fuzzy logic to support intelligent decision-making [18, 56]. GrC has promoted the adoption of fuzzy logic for data abstraction to be capable of processing various data types in classification tasks using granular decision trees [29, 30]. A top-rated operator in deep learning for extracting relevant features, the convolution, had been adapted to operate with fuzzy sets with a granular perspective for classification problems [11]. In [66] is proposed a polynomial-feature granulation method based on long short-term memory network for oxygen supply network prediction.

In this work, a decision-tree model discovers the rule base before characterizing its elements in a granular paradigm. Given an a priori defined granules collection, this work selects the best ones to create higher-level Information Granules. It then performs the fuzzification to develop a new Sugeno-type fuzzy rule base with learning capability due to their analogous neural network representation.

4 Granules Construction

Different levels of abstractions define the granule construction process. The level of these granules is denoted by the subindex A_i. In the first level of granularity, A_0 corresponds to the original data space, so that a zero-level granule is equivalent to a given instance of the dataset $A_0 \approx x^{(i)}$.

A level 1 Information Granule (A_1) is characterized as the tuple (m, r) that correspond to a range $[m - r, m + r]$, notice that in this particular scenario, it is defined with a symmetric proportion from the median (could be further extended). A level 2 Information Granule (A_2) is described as an implication relation; at this abstraction level, the interaction between different domain Information Granules is considered. A level 3 Information Granule in this work is defined as a decision set.

The notation to denote a granulation process is through eq. 3:

$$\mathcal{G}(\mathcal{A}'_i) = A_{i+1}, \tag{3}$$

where $\mathcal{G}$ is the mapping process to form a higher Information Granule given a set of lower-level granules, $\mathcal{A}'_i \subseteq \mathcal{A}_i$ denote a set that belongs to the i-level granular space, and A_{i+1} is a formed higher level granule. For instance, the first abstraction level starts from the crisp data space, such that $\mathcal{G}(X) = \mathcal{G}(\mathcal{A}_0) = A_1$. For notation purposes $\mathcal{G}^l(A)$ denotes the process of perform l-level abstraction processes for a given granule A.

The notation to denote a degranulation process is through eq. 4:

$$\mathcal{G}^{-1}(A_i) = \{A^{(1)}_{i-1}, \ldots, A^{(n)}_{i-1}\} \subseteq \mathcal{A}_{i-1}, \tag{4}$$

where $\mathcal{G}^{-1}$ is the mapping process to form a lower Information Granule given a granule, $\mathcal{A}_i$ denote the i-level granular space, and A_{i-1} is a formed lower level granule.

It is essential to notice that the raw representation of a granule is a set that is formed by granules of lower levels. Intuitively, a granule should be represented by a model that requires low information.

For a level 1 Information Granule is defined the following metrics to measure the coverage (eq. 5) and specificity (eq. 6):

Coverage

$$cov(A_1) = \frac{card(\{x_k | (m - r) < x_k < (m + r)\})}{N}. \tag{5}$$

Specificity

$$sp(A_1) = 1 - \frac{|r|}{|X_{max} - m|}, \tag{6}$$

where A_1 is an Information Granule, $x_k \in X$, m, and r characterize a range (level 1 granule); m is the median of the range, and r the distance to the rage limits in a symmetric way. $card(\cdot)$ stands for the cardinality of a given set:

$$\mathcal{G}(A'_1) = \bigwedge_i \rho({A_1}^{(i)}; \mathbf{x}) \rightarrow Y = A_2, \tag{7}$$

where $A'_1 \subseteq \mathcal{A}_1$, that in their raw representation is a set that belongs to the $\mathcal{A}_1$ space, namely a set of ranges (denoted by a tuple (m, r)) which conforms an IF-THEN rule.

The operator $\wedge$ represent the conjuntion operation. $\rho : \mathcal{A}_1 \times X$ is a logical function that forms the proposition "x `belongs to the granule` $\mathcal{A}_1^{(i)}$". Y is the target domain:

$$\mathcal{G}^{-1}(A_2) = \mathcal{A}'_1, \quad \mathcal{A}'_1 \subseteq \mathcal{A}_1. \tag{8}$$

In the degranulation process of a level 2 granule, the raw representation of the operation is a set of level 1 granules, and those granules form the antecedent part in the rule structure.

For a level 2 Information Granule are defined the following metrics to measure the coverage (eq. 9) and specificity (eq. 10):

Coverage

$$\begin{aligned} cov(A_2) = &\frac{1}{N}\sum_{i}^{N} card(\{x^{(i)} | \forall x^{(i)} \in X, \\ &\forall A_1^{(i)} \in \mathcal{G}^{-1}(A_2), \rho(A_1^{(i)}; x^{(i)})\}). \end{aligned} \tag{9}$$

Specificity

$$sp(A_2) = \frac{card(\{\mathcal{G}^{-1}(A_2)\})}{dim(X^{(i)})}, \tag{10}$$

where A_2 is an Information Granule formed by a implication relationship $\bigwedge_i \rho({A_1}^{(i)}; \mathbf{x}) \rightarrow Y$, N is the number of instances that correspond to the dataset; and, $dim(X^{(i)})$ is the number of considered features in the dataset. $card(\cdot)$ stands for the cardinality of a given set.

To select a justifiable Information Granule, it is necessary a measure their formation quality. Due to the contradictory behavior of coverage and specificity can form a Pareto front to select the best candidates. Notice that the Pareto front is formed by the product and can be computed to any granule at any abstraction level (eq. 11):

$$Q_{l_i}(A_i) = sp(A_i) * cov(A_i)^{\gamma_i}. \tag{11}$$

There is a parameter that serves to prioritize one of the terms [38], γ_i. If $0 \geq gamma_i < 1$ there is pondered more the coverage, on the contrary, if $\gamma_i > 1$ then the specificity gets more relevant in the calculus. For each abstraction level, a different value γ_i can be defined.

4.1 Hierarchical Quality Measurement

The quality construction of a given of H level granule is measure with the proposed metric (eq. 12), which perform successive degranulation process and multiply their Pareto front values to the lower level granules:

$$V(A_h) = \{Q_{l_h}(A_h) \times V(A'_{h-1}) | \forall A'_{h-1} \in \mathcal{G}^{-1}(A_h)\}. \tag{12}$$

The optimization problem is shown in equation 13, which aims to find the more appropriate level 1 granules. The aptitude function is the hierarchical measure (eq. 12), their restrictions are: 1) The solution set must be the minimum cardinality (showed as the function $card(\cdot)$) 2) The aggregation of the values v_i (hierarchical Pareto) should be equal or less the to regularization parameter α, and 3) The number of elements in the solution set must be equal or less than the regularization parameter ξ. These two regularization parameters allow finding the best level 1 granules smaller set with at least a cumulative value of α and does not have more than ξ elements:

$$\begin{aligned} A_H^* = &\underset{\mathcal{A}'_1 \subseteq \cdots \subseteq \mathcal{A}_H}{\operatorname{argmax}} \; V(\mathcal{G}^2(\mathcal{A}'_1)) \\ &\text{subject to:} \\ &1) \quad \min card(\{\mathcal{A}'_1\}), \\ &2) \quad \sum_{i=1}^{card(\{\mathcal{G}^{-1}(\mathcal{A}_H)\})} V(\mathcal{G}(\mathcal{A}'_1))_i \leq \alpha, \\ &3) \quad card(\{\mathcal{A}'_1\}) \leq \xi. \end{aligned} \tag{13}$$

The resulted Information Granule collection reconstructs the decision set, formally by $G^2(A^*)$. This process can be treated as a post-pruning technique, the bias of the model increases while its variance decreases. The initial structure for building the knowledge base of the FIS to be optimized is formed by the decision set. Figure 1 shows the general block diagram of the proposed methodology for data-driven fuzzy rule base construction. The details of the neuro-fuzzy model characteristics are in section 5.

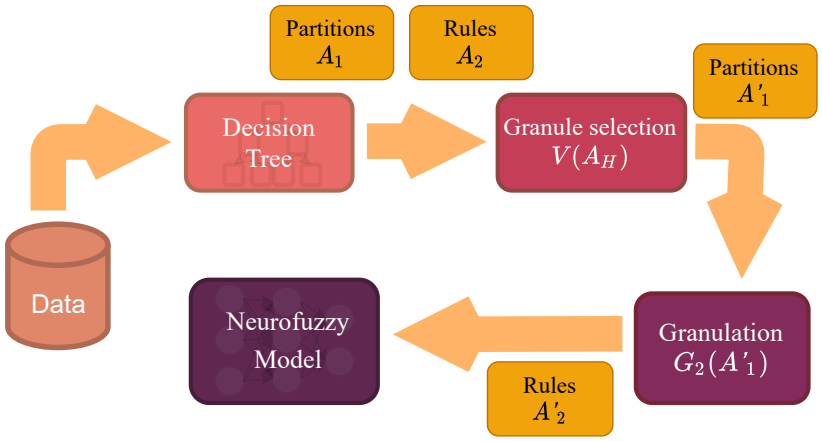


Fig. 1. Block diagram of the proposed methodology for building a fuzzy rule-based system from decision set created by decision-tree model using the principle of justifiable granularity

5 Sugeno-Type Neuro-Fuzzy Model

The Sugeno fuzzy systems allow their construction in a systematic form to generating fuzzy rules in a data-driven manner, they were proposed by Takagi, Sugeno, and Kang [45]. These systems are also composed of IF-THEN rules, but only their antecedent belongs to the fuzzy space. The consequent part is a crisp function that maps the input space. As in the Mandani type fuzzy systems, the rules might fire all at once to get a crisp output value, can compute a simple weighted average of function outputs, which is less time-consuming than defuzzification in Mamdani type fuzzy systems.

The knowledge base can be described as follows:

$$R^1: \text{IF } x_1 \text{ is } low \text{ and } \ldots \text{ and } x_m \text{ is } low \text{ THEN, } y \text{ is } \sigma^{(1)}_{j\in J}(\mathbf{x};\mathbf{w}),$$
$$R^2: \text{IF } x_1 \text{ is } low \text{ and } \ldots \text{ and } x_m \text{ is } high \text{ THEN, } y \text{ is } \sigma^{(2)}_{j\in J}(\mathbf{x};\mathbf{w}),$$
$$\vdots$$
$$R^n: \text{IF } x_1 \text{ is } high \text{ and } \ldots \text{ and } x_m \text{ is } low \text{ THEN, } y \text{ is } \sigma^{(n)}_{j\in J}(\mathbf{x};\mathbf{w}),$$

where x_i is a fuzzy variable (which models the feature space), their fuzzy values are represented by the terms *low*, *high*, etc. y is the output space described by the function $\sigma^{(n)}_{j\in J}(\mathbf{x};\mathbf{w})$, where J represents the output classes, $\mathbf{x}$ are the input crisp values, and $\mathbf{w}$ are the function's coefficients.

Once the knowledge base is designed, some optimization methods can adjust their membership function parameters to fit the data better. This optimization process comprehends the membership function and consequent crisp function parameters. In this approach, to maintain the interpretability characteristic, sigmoid and linear functions are selected.

An analogous neuro-fuzzy architecture carries out the optimization of fuzzy model parameters. This architecture comprises five layers: input, fuzzification, inference, implication, and de-fuzzification layer. The connections between the layers are not fully connected to maintain coherent antecedent relationships in the fuzzy rules. In the fuzzification layer are only connected the membership functions belonging to the input domain. In the implication layer, each crisp function is only related to the rule's output (a given class). Figure 2 shows a visual representation of a Sugeno-type neuro-fuzzy model.

The parameters to optimize the model belong to the antecedent part of the fuzzy rule and the (trainable) parameters in the crip consequent function.

The neural architecture is shown in figure 2. The first layer is non-fully connected among neurons that represent the fuzzification process. Wich maps a crisp input to the fuzzy space:

$$f^{B^j_k}(x_i) = \mu_{B^j_k}(x_i), \tag{14}$$

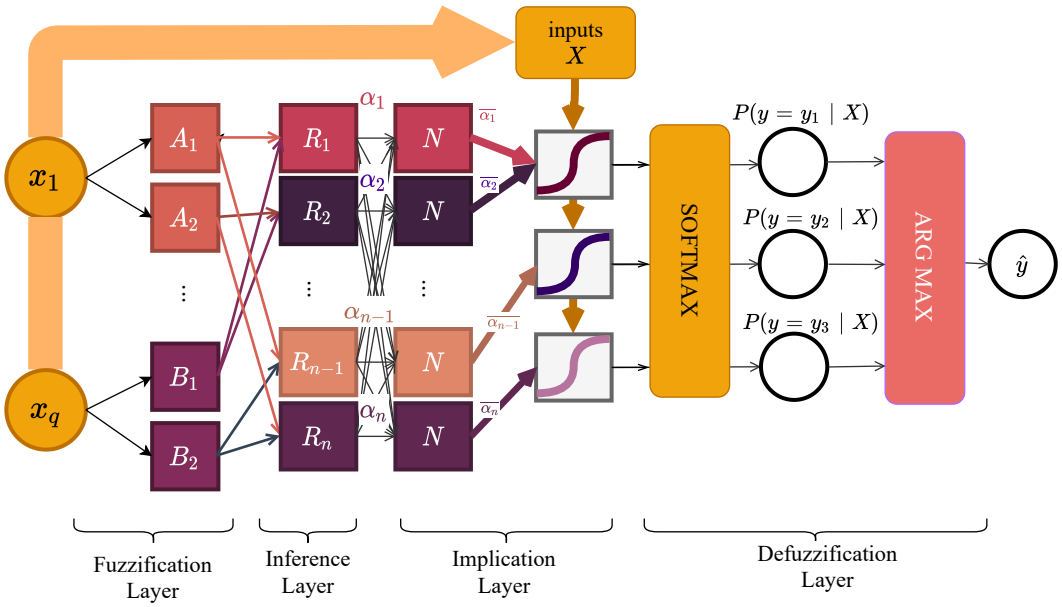


Fig. 2. The Sugeno Neuro-fuzzy representation is composed of a non-fully connected 5-layer artificial neural network. All the computations correspond to the involved operations in a Sugeno-type fuzzy inference system for q inputs, n rules, and crisp consequent functions. Each rule is only linked to one class $j \in J$ target space

where $B_r^j \in V_k$ is a fuzzy set that belongs to the fuzzy variable V_k, the domain of each fuzzy variable is shared by its corresponding attribute domain in the dataset. Only the membership functions directly related to the attribute are evaluated by the input value, which results in a semi-connected layer.

The inference layer is also non-fully connected, fires at a certain strength value in the range $[0, 1]$. The implication operation calculates a *t-norm* ($\tilde{*}$) as a product:

$$\alpha^l(x_i) = \prod_{r=1}^{p} f^{B_r}(x_i). \tag{15}$$

The implication layer performs a normalization operation that conforms a step to the weighted average to the output:

$$\bar{\alpha}^l(x_i) = \frac{\alpha^l(x_i)}{\sum_{j=1}^{L} \alpha^j(x_i)}. \tag{16}$$

After the normalization process, for each rule that has a crisp function consequent related to an output class $j \in J$, a $\tilde{*}$ as the product is calculated:

$$z^j(x_i) = \sum_{l=1}^{M} \{\sigma(x_i; \mathbf{w}^l) \times \bar{\alpha}^l(x_i)\}, \tag{17}$$

where σ corresponds to sigmoid function that transform the input space, $\sigma(x_i; \mathbf{w}^l) = \frac{1}{1+e^{-\mathbf{w}^l x_i}}$. After the sigmoid transformation, all values are computed by the softmax function, which maps proportionally for each class, values in the range $[0, 1]$ and $\sum_{j=1}^{J} h_j = 1$, it commonly interprets those values as a probability output:

$$h_j = P(y = j|\mathbf{x}) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}. \tag{18}$$

The output value dimension corresponds to the number of classes in the problem domain. Getting an actual label as a result value could be computed just as the argument position with the higher probability value:

$$\hat{y} = \arg\max_{j \in J} P(y = j|\mathbf{x}). \tag{19}$$

This neuro-fuzzy model adjusts its parameters to fit a given target better. To measure the error of the prediction is used the Cross-Entropy loss function ($Loss_{CE}$):

$$Loss_{CE}(\mathbf{y}, h(\mathbf{x}; \mathbf{w})) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{J} y_j^{(i)} \log(h_j(x^{(i)})) + \lambda_1 ||\mathbf{w}||_1 + \lambda_2 ||\mathbf{w}||_2, \quad (20)$$

where N is the number of instances on the batch; y is the target value in the form of `one-hot-encoding`, and h_j is the predicted probability of belonging to the class j. In the loss function are defined regularization l_1 and l_2 norms to constrain the weight values and prevent overfitting. These regularizers can help to improve the interpretability by only *relevant* considering relevant features.

The trainable parameters on the neuro-fuzzy model are the design parameters of the fuzzy sets and the set of crisp consequent function parameters; in this setup, the membership functions are defined by Gaussian functions, then the trainable parameters are the mean and σ values, where $\sigma > 0$.

A gradient descent optimization method is applied to find the best parameters. The learning rule is shown in equation 21:

$$\theta^{\text{new}} = \theta^{\text{old}} - \eta \nabla E(\theta^{\text{old}}), \quad (21)$$

where θ are the parameter vector values; $E(\theta)$ is the gradient of the error value of the model with the parameters θ; η is the learning rate value in $0 < \eta < 1$.

The hyper-parameters of the model are:

— *Learning rate value:* this value scales the directional vector generated by gradient calculation, as the lower the value, the better search is but slower. Usually, the default value is set to a value of 0.01.

— *Batch size:* the selection of dataset partition to train the model is set by this value.

— *Epoch number:* an epoch represents an entire iteration overall dataset (train dataset partition).

— *Goal error value:* is a threshold value to consider to stop the training process because is considered acceptable at that value.

— *Output function structure:* is the computation that transforms the input value to a crisp output value to further averaging pondered by the firing strength values.

— *Regularization coefficients λ_1 and λ_2:* those restrict how much the trainable parameters in the output crips functions increases.

6 Experimentation

Ten publicly available datasets [1] are used in order to evaluate the proposed model under different domain applications. All datasets correspond to classification tasks; their characteristics are shown in the table 1.

Table 1. The selected publicly available datasets at UC Irvine Machine Learning Repository[1] for evaluating the proposed methodology

	dataset	instances	features	classes
1	abalone	4177	8	3
2	credit-g	1000	20	2
3	creditcard	284807	29	2
4	diabetes	768	8	2
5	ionosphere	351	34	2
6	iris	150	4	3
7	sonar	208	60	2
8	spambase	4601	57	2
9	wdbc	569	30	2
10	wine	178	13	3

The methodology consists of primary three steps:

1. Decision tree construction to extract and generate a rule-based decision set.
2. The decision set reconstruction following the principle of justifiable granularity for the selection of the more meaningful granules.

[1] `https://archive.ics.uci.edu/ml/index.php`

3. The construction of Sugeno-type neuro-fuzzy architecture for classification tasks then optimized their membership function design parameters and coefficients of the consequent output functions.

6.1 Decision Tree Construction

Given a training dataset $\mathcal{D}_{train} = \{(x, y)\}_{i=1}^{N}$, a decision tree model is train to map the input data patterns to the target domain space, $f_{tree}(\mathbf{x}; \mathrm{T}) \rightarrow \mathbf{y}$. The hyper-parameters set for this experiment are: i) complete search of the feature space in each partition split; ii) one element at minimum belonging to a node to create a branch; iii) Gini impurity criteria to measure the quality of a split; iv) without the maximum depth of the three, that means the nodes are expanded until all leaves are pure.

Creation of a decision set (M_{ds}) by traversing the tree paths from the root to the leaves nodes (decision nodes). Due to the possible repetition of some features for the splitting, it is necessary to simplify the rules by limiting each feature to be clustered only in one range (this process maintains the model's fidelity and does not affect the original representation outcome). Each leaf node creates a rule; therefore, it can be a potentially large number of them. Some criteria must clip all feature ranges. In this experiment, the maximum and minimum values for each feature are taken to replace those undefined boundaries. At this step, every node has been contributed to creating intervals $[m, r]$ where m is the mean value and r is the distance from the mean to the left and right, that are further used to form proposition such as "x `is in` $[m - r, m + r]$".

6.2 The Decision Set Reconstruction by the Principle of Justifiable Granularity

The intervals formed from the learned tree T compounds the first level Information Granules. All ranges created by the decision tree are represented with the tuple (m, r), where m is the median of the range, and r is the distance to some boundary (notice that only represents symmetrical granules).

Once all level 1 Information Granules are generated ($\mathcal{A}_1$ space), then by following the antecedents of the rules, the level 2 Information Granules are generated ($\mathcal{A}_2$ space). At this level, the relationship between lower-level granules are established (see section 4). Next, those level 2 Information Granules are grouped to form a structure-less level 3 Information Granule; namely, it represents the decision set (M_{ds}).

Due to the potentially large number of elements in the decision set M_{ds} (now forming a level 3 granule), it is necessary to prune it. The pruning process follows the principle of justifiable granularity; in this context, instead of selecting a numerical range of values, the best set of level 1 Information Granules that carry out more meaningful information in terms of coverage and specificity. Next, the optimization process (defined in the section 4) is conducted to find the best lower level granules ($\mathcal{A}_1^*$) that contribute the most to create higher quality Information Granules of higher levels.

The set of level 1 granules are clustered by the granulation process to reconstruct a decisition set with fewer information (less variance and more bias) since the lack of some elements (antecedents and rules), $M'_{ds} = \mathcal{G}^2(A_1^*)$, where $M'_{ds} \subseteq M_{ds}$.

6.3 Sugeno-Type Neuro-Fuzzy Optimization

Once the decision set is reconstructed (M'_{ds}), it is converted to a fuzzy inference system using Gaussian membership functions to represent the antecedent ranges($[m - r, m + r]$). In other words, the crisp Information Granules A_1 are fuzzified. Then those are defined by a tuple (m, σ) to represent the Gaussian membership function, where m is the same as the mean of the range in the initial A_1, and σ is the standard deviation, the value approximates this value r, such that $\sigma = r/2$.

The consequent part of each rule is used to define a sigmoid function that is related to the target class $\sigma_{j \in J}^{(r)}(\mathbf{x}; \mathbf{w})$, where r denotes the rule and j the specific class. This sigmoid computes the dot product between the input vector and a set of initially random weights $\sigma(\mathbf{w}^T\mathbf{x})$; each class at least has a function associated with it. The output of this function evaluation is multiplied by

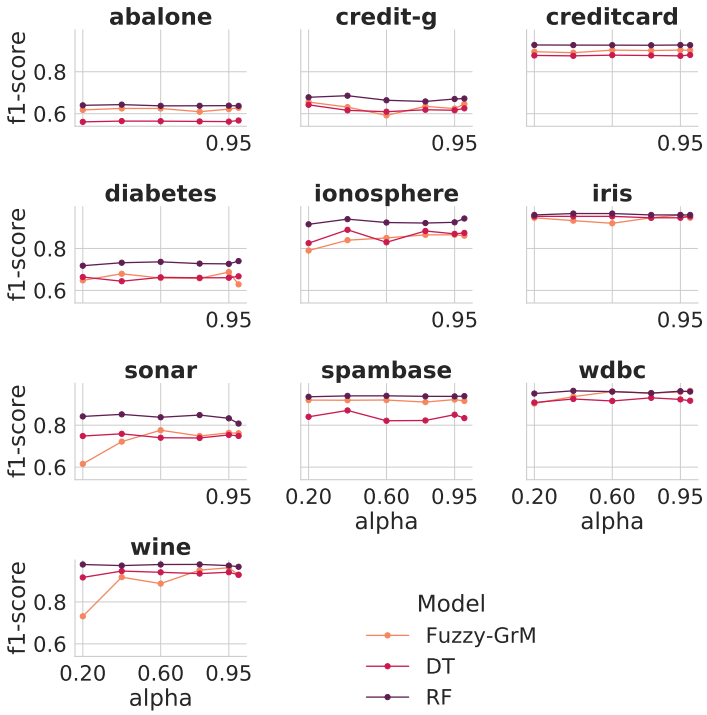


Fig. 3. Performance comparison between the decision tree-based models (Decision Tree and Random Forest) and the proposed granular model (Fuzzy-GrM) at diverse α values. As lower the α value, the higher compression the model is

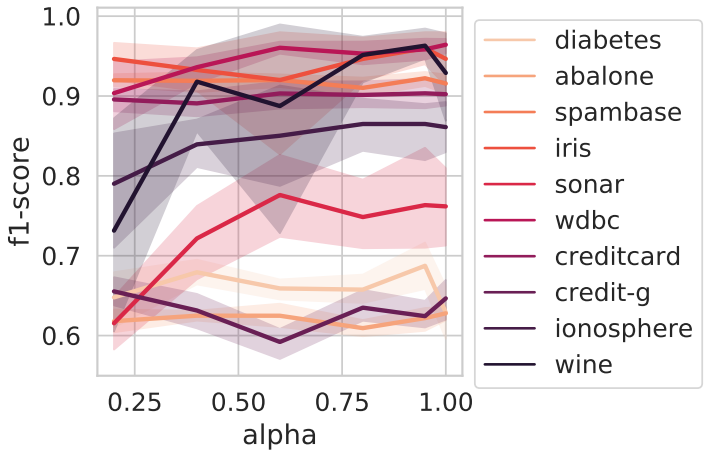


Fig. 4. Overall performance comparison with diverse values of alpha, applied in the ten different classification datasets

Table 2. The proposed model average results with different alpha values applied in 10 publicly available datasets

	alpha	0.20	0.40	0.60	0.80	0.95	1.00
DT	mean	0.79	0.80	0.79	0.80	0.80	0.80
	σ	0.13	0.15	0.14	0.14	0.14	0.14
RF	mean	0.85	0.86	0.86	0.85	0.86	0.86
	σ	0.13	0.13	0.13	0.13	0.13	0.13
Fuzzy-GrM	mean	0.77	0.81	0.81	0.82	0.83	0.82
	σ	0.14	0.13	0.15	0.14	0.14	0.14
Element reduction	mean	93.91%	87.21%	81.02%	71.20%	58.61%	49.09%
Rule reduction	mean	82.51%	71.61%	61.38%	49.09%	41.10%	39.48%

the normalized firing strength of the rules that are linked to the target class consequent (equation 16). At this point, some selection criterion defines the output (e.g., the label of a more significant output rule).

In this work is used a softmax computation, to define the probabilistic output of the FIS (equation 18). In addition, to better interpret inputs, it creates a smooth solution surface space in the training step of the neuro-fuzzy by gradient descent-based optimization algorithms.

7 Results

The proposed model was evaluated by 5-fold cross-validation, in 10 publicly available datasets for classification, with 6 different values for the hyper-parameter α which serve up to select the compression level by selecting the most relevant elements (according to eq. 12). The obtained results for overall Sugeno-type neuro-fuzzy model for classification (showed in table 2) were f1-scores of 0.77, 0.81, 0.81, 0.82, 0.83, 0.82 when α-values were set to 0.2, 0.4, 0.6, 0.8, 0.95, 1 respectively, with the maximimun rules parameter (ξ) set to 50.

Considering all different values of α, in average the model reduction (in terms of elements) was 73.51% with $\sigma = 17.23$. From a rule percentage reduction perspective, the average compression was 57.53% with $\sigma = 17.34$, and relative error concerning the Simple Decision Tree model of -1.4% with $\sigma = 0.021$, and 5.55% with $\sigma = 0.022$ respect to Random Forest model (in this experiment the number of weak learners was set to 100).

Figure 3 shows the performance comparison between the decision tree-based models and the proposed one at diverse α values.

At the global level, considering all analyzed datasets, the minimum elements compression percentage given the following values of the prunning value α were: 81.25% with $\alpha = 0.2$, 58.33% with $\alpha = 0.4$, 50% with $\alpha = 0.6$, 23.07% with $\alpha = 0.8$, 81.25% with $\alpha = 0.2$, 11.11% with $\alpha = 0.95$, 0% with $\alpha = 1$. At rule compression percentage were obtained: 57.14% with $\alpha = 0.2$, 33.33% with $\alpha = 0.4$, 25% with $\alpha = 0.6$, and no rule compression at all for higher values for α.

In dataset results tables (10-7), there are comparisons of the f1-score of the proposed model (F-GrM), Decision Tree (DT), and Random Forest (RF). Figure 5 shows as higher compression is applied to the model, the higher the variance is. Figure 4 shows the overall performance comparison of different hyperparameters.

A paired sample t-test over the mean f1-score values was used to formally validate the results in all experiments with a confident value of 95%. Table 12 shows the comparison between neuro-fuzzy (with different compression rates) and decision tree models.

All mean f1-score values for the proposed model and random forest have a significant difference (RF had substantially better general performance than the proposed model).

Table 3. Proposed model results with different alpha values applied in the sonar dataset

model	alpha		f1-score	ER	RR
F-GrM	0.20	mean	0.62	94.21	82.22
		σ	0.04		
	0.40	mean	0.72	86.61	72.41
		σ	0.06		
	0.60	mean	0.78	74.54	47.13
		σ	0.07		
	0.80	mean	0.75	54.17	21.59
		σ	0.05		
	0.95	mean	0.76	22.45	1.14
		σ	0.08		
	1.00	mean	0.76	0.00	0.00
		σ	0.07		
DT		mean	0.75		
		σ	0.02		
RF		mean	0.84		
		σ	0.02		

Table 4. Proposed model results with different alpha values applied in the wdbc dataset

model	alpha		f1-score	ER	RR
F-GrM	0.20	mean	0.90	95.33	85.26
		σ	0.06		
	0.40	mean	0.94	87.56	73.03
		σ	0.02		
	0.60	mean	0.96	77.53	47.73
		σ	0.01		
	0.80	mean	0.95	60.14	30.00
		σ	0.02		
	0.95	mean	0.96	33.50	10.00
		σ	0.02		
	1.00	mean	0.96	0.00	0.00
		σ	0.01		
DT		mean	0.92		
		σ	0.00		
RF		mean	0.96		
		σ	0.00		

It is important to note that the fuzzy granule model has considerably fewer rule elements even in configurations with no significant difference in the mean f1-scores with respect to the DT model.

Table 5. Proposed model results with different alpha values applied in the credit-g dataset

model	alpha		f1-score	ER	RR
F-GrM	0.20	mean	0.66	95.17	84.78
		σ	0.02		
	0.40	mean	0.63	90.59	71.51
		σ	0.03		
	0.60	mean	0.59	90.85	72.71
		σ	0.02		
	0.80	mean	0.63	90.88	73.57
		σ	0.02		
	0.95	mean	0.62	90.81	73.25
		σ	0.02		
	1.00	mean	0.65	90.82	71.95
		σ	0.03		
DT		mean	0.62		
		σ	0.02		
RF		mean	0.67		
		σ	0.01		

Table 6. Proposed model results with different alpha values applied in the iris dataset

model	alpha		f1-score	ER	RR
F-GrM	0.20	mean	0.95	83.15	61.54
		σ	0.03		
	0.40	mean	0.93	70.51	51.43
		σ	0.03		
	0.60	mean	0.92	55.84	37.14
		σ	0.11		
	0.80	mean	0.95	36.71	8.57
		σ	0.04		
	0.95	mean	0.96	17.33	0.00
		σ	0.03		
	1.00	mean	0.95	0.00	0.00
		σ	0.04		
DT		mean	0.95		
		σ	0.00		
RF		mean	0.96		
		σ	0.00		

According to the validation, there is a significant difference between models where alpha is 0.2, 0.95, and 1; in the first value, which the model compresses the most the rules (around 82%), the

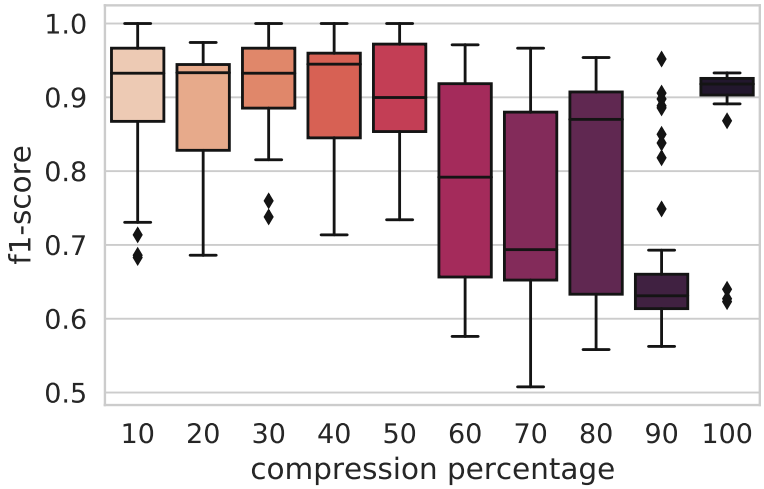


Fig. 5. The relationship between performance and model compression percentage in terms of antecedents

Table 7. Proposed model results with different alpha values applied in the wine dataset

model	alpha		f1-score	ER	RR
F-GrM	0.20	mean	0.73	89.40	67.39
		σ	0.18		
	0.40	mean	0.92	76.86	58.97
		σ	0.07		
	0.60	mean	0.89	67.97	53.66
		σ	0.18		
	0.80	mean	0.95	46.09	19.51
		σ	0.04		
	0.95	mean	0.96	16.92	2.44
		σ	0.02		
	1.00	mean	0.93	0.00	0.00
		σ	0.07		
DT		mean	0.94		
		σ	0.01		
RF		mean	0.97		
		σ	0.01		

initial decision tree model result is higher by 0.02. However, in alpha values 0.95 and 1, which the model comprises the lowest the rules (around 41% and 39%, respectively), the proposed model is higher by 0.03 and 0.02, respectively.

There is no significant difference between model results in intermediate alpha values, although the decision sets were compressed between 71% and 49%.

Table 8. Proposed model results with different alpha values applied in the ionosphere dataset

model	alpha		f1-score	ER	RR
F-GrM	0.20	mean	0.79	94.72	80.00
		σ	0.09		
	0.40	mean	0.84	86.34	57.80
		σ	0.04		
	0.60	mean	0.85	77.94	37.39
		σ	0.08		
	0.80	mean	0.86	59.22	18.42
		σ	0.04		
	0.95	mean	0.86	39.28	3.48
		σ	0.05		
	1.00	mean	0.86	34.21	3.54
		σ	0.04		
DT		mean	0.86		
		σ	0.02		
RF		mean	0.93		
		σ	0.01		

Table 9. Proposed model results with different alpha values applied in the spambase dataset

model	alpha		f1-score	ER	RR
F-GrM	0.20	mean	0.92	97.93	97.00
		σ	0.01		
	0.40	mean	0.92	96.71	95.07
		σ	0.02		
	0.60	mean	0.92	96.63	94.58
		σ	0.01		
	0.80	mean	0.91	96.56	94.71
		σ	0.01		
	0.95	mean	0.92	96.60	94.99
		σ	0.01		
	1.00	mean	0.92	96.56	94.71
		σ	0.02		
DT		mean	0.84		
		σ	0.01		
RF		mean	0.94		
		σ	0.00		

8 Conclusion and Future Work

This work addressed the problem of input space partition to create the base rule structure for a FIS by selecting the meaningful elements from a decision tree model given a compression

Table 10. Proposed model results with different alpha values applied in the diabetes dataset

model	alpha		f1-score	ER	RR
F-GrM	0.20	mean	0.65	95.45	87.90
		σ	0.05		
	0.40	mean	0.68	88.11	71.95
		σ	0.02		
	0.60	mean	0.66	82.19	61.26
		σ	0.02		
	0.80	mean	0.66	81.86	60.84
		σ	0.02		
	0.95	mean	0.69	81.91	62.10
		σ	0.04		
	1.00	mean	0.63	81.82	59.85
		σ	0.04		
DT		mean	0.66		
		σ	0.01		
RF		mean	0.73		
		σ	0.01		

Table 11. Proposed model results with different alpha values applied in the abalone dataset

model	alpha		f1-score	ER	RR
F-GrM	0.20	mean	0.62	96.85	88.57
		σ	0.02		
	0.40	mean	0.62	96.84	87.89
		σ	0.01		
	0.60	mean	0.62	96.84	88.37
		σ	0.02		
	0.80	mean	0.61	96.82	88.50
		σ	0.01		
	0.95	mean	0.62	96.85	88.51
		σ	0.02		
	1.00	mean	0.63	96.85	88.71
		σ	0.01		
DT		mean	0.56		
		σ	0.00		
RF		mean	0.64		
		σ	0.00		

rate value (controlled by the hyperparameter α). The motivation for choosing a subset of the elements in rule-based systems is to maintain their structure as simple as possible, directly

Table 12. Paired sample t-test of mean f1-scores for formal comparison between the decision tree model and the resulting fuzzy rule-based model constructed by the proposed methodology

	alpha	0.20	0.40	0.60	0.80	0.95	1.00
DT	mean	**0.79***	0.80	0.79	0.80	0.80	0.80
	σ	0.13	0.15	0.14	0.14	0.14	0.14
Fuzzy-GrM	mean	0.77	**0.81**	**0.81**	**0.82**	**0.83***	**0.82***
	σ	0.14	0.13	0.15	0.14	0.14	0.14
P-value		0.003	0.069	0.102	0.126	5.68×10^{-4}	4.32×10^{-2}
T-student		2.86	1.50	1.28	-1.15	-3.45	-1.74
DF		49	49	49	49	49	49
Significant difference		YES	NO	NO	NO	YES	YES
Element reduction	mean	93.91%	87.21%	81.02%	71.20%	58.61%	49.09%
Rule reduction	mean	82.51%	71.61%	61.38%	49.09%	41.10%	39.48%

***** shows significant difference.

Table 13. Proposed model results with different alpha values applied in the creditcard dataset

model	alpha		f1-score	ER	RR
F-GrM	0.20	mean	0.90	97.63	92.20
		σ	0.02		
	0.40	mean	0.89	94.40	79.70
		σ	0.02		
	0.60	mean	0.90	92.23	75.89
		σ	0.02		
	0.80	mean	0.90	91.86	76.02
		σ	0.02		
	0.95	mean	0.90	91.81	75.77
		σ	0.02		
	1.00	mean	0.90	91.79	76.53
		σ	0.02		
DT		mean	0.88		
		σ	0.00		
RF		mean	0.93		
		σ	0.00		

impacting their interpretability. Fewer antecedents lead to understanding the phenomena with less effort; also, the number of rules affects too. In order to build a knowledge base with the greatest interpretability, it is necessary to create it with fewer antecedents and rules. The antecedents were characterized as type-one granules by ranges, while the implications were represented as type-two granules through relationships between granules of the lower level. The abstraction of decision sets is made by hierarchical structure between the different levels in the granules.

The transition from one level to another is defined through the granulation and degranulation operations.

The methodology for extracting the relevant elements was based on the principle of justifiable granularity, which has a higher value of specificity and coverage. In this approach, the optimization method selects all level granules simultaneously through the hierarchy established. The reconstruction of the selected granules creates a new decision set with fewer elements than the original one due to the optimization restrictions: i) minimize the number of level-one granules; ii) the cumulative Pareto front values must be equal or less than a regularizer hyper-parameter α; iii) a hyperparameter ξ restricts the number of level-one granules to consider in the selection.

The presented work defined a hierarchical measurement that helps to consider the best lower-level information granules better suited to form high-quality higher-level Information Granules. This well-formed Information Granules search is defined as an optimization problem with restrictions. The objective is to find the smallest set of best lower-level granules that maximize the hierarchical quality measurement composed by the Pareto front at different levels of specificity and coverage. As the number of elements decreases in the rule-based system, the bias increases; to increase the model variance, the resulting pruned decision set is converted to a Sugeno-type neuro-fuzzy model for classification tasks that is further optimized.

The proposed Sugeno-type neuro-fuzzy model for classification has the following characteristics:

— The fuzzification layer is not fully connected, which prevents incoherent rule formation. For the sake of interpretability, the rules must be coherent and sound to the data scientist.

— The implications layer is also not fully connected, which reduces ambiguity in the output, maintaining separated rule contribution for each class; this characteristic tends to analyze the conditions to belong to a given class more easily.

— Sigmoid activation functions transform the output of the implication layer to get interpretable outcomes for classification tasks.

— In order to increase the output interpretability, is used a softmax layer to get the outcome of the model in a probability fashion that helps to suit the belongingness to the target classes better.

The results support that using a FIS with the proposed method for rule selection can compress the contained information in a classical decision set without significantly compromising the performance model. The different compression rate values (α) impact the model performance.

As shown in the results, a higher compression rate (lower α value, e.g., 0.2) degrades the model performance significantly; however, this small decision set in a complex domain might be helpful to have a more interpretable model to understand phenomena better. An attractive characteristic is that intermediate compressing rate values (e.g., between 0.4 and 0.8) achieve a considerable reduction in the decision set without significant difference performance. In problems where the performance is crucial, higher α values are recommended, which less compress the decision set but still might be considerable (e.g., at least a mean of 39% in this experimental setup).

The rule compression achieved by the proposed method is relevant in the context of IML due that simplifies the decision model by i) reducing the number of elements (antecedents and rules), decreases as well the complexity of the systems, which improves the interpretability; ii) fuzzy logic brings an interface in natural language to the human and promotes a better understanding of the model; iii) the compression of the model can be controlled by α hyper-parameter and be set to the most convenient value for a specific application domain (see figure 3 compare the behavior of α in different domain applications).

This proposal opens the opportunity to further research in decision-set-based Information Granules for smaller and more interpretable models; this methodology can be used with different models/methods that generate decision sets. An extension of the current work is

considered to design a higher-level information granule, specificity and coverage metric to select the most relevant decision set source elements for ensemble methods.

Another possible extension of this work is to use type-2 fuzzy logic, which better manages decision processes under uncertainty and achieves better performance with a smaller knowledge base in terms of rules. The overlapping of rule partition with different consequent could characterize uncertainty in higher-level fuzzy sets.

The proposed Hierarchical Decision Granules Optimization method can be adapted to any rule-based system by defining specificity and coverage metrics for each granule level. It can be interesting to incorporate different information frameworks such as probabilistic and rough sets to enhance the intrinsic semantic meaning in the Information Granule, therefore generate richer explanations taking advantage of the natural language interface that brings the fuzzy logic.

Acknowledgments

This research was supported by CONACyT (Consejo Nacional de Ciencia y Tecnología) with grant number 691247.

References

1. **Adhyaru, D. M., Patel, J., Gianchandani, R. (2010).** Adaptive neuro-fuzzy inference system based control of robotic manipulators. ICMET 2010 - 2010 International Conference on Mechanical and Electrical Technology, Proceedings, pp. 353–358.
2. **An, S., Hu, Q., Wang, C. (2021).** Probability granular distance-based fuzzy rough set model. Applied Soft Computing, Vol. 102.
3. **Azadeh, A., Gaeini, Z., Motevali Haghighi, S., Nasirian, B. (2016).** A unique adaptive neuro fuzzy inference system for optimum decision making process in a natural gas transmission unit. Journal of Natural Gas Science and Engineering, Vol. 34, pp. 472–485.
4. **Bandyopadhyay, S., Yao, J., Zhang, Y. (2017).** Granular computing with compatibility based intuitionistic fuzzy rough sets. **Chen X. Luo B., L. F. P. V. W. M. A.**, editor, Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, volume 2017-Decem, Institute of Electrical and Electronics Engineers Inc., pp. 378–383.
5. **Bargiela, A., Pedrycz, W. (2003).** Granular Computing.
6. **Bello, M., Nápoles, G., Vanhoof, K., Bello, R. (2021).** Data quality measures based on granular computing for multi-label classification. Information Sciences, Vol. 560, pp. 51–67.
7. **Brasoveanu, A., Moodie, M., Agrawal, R. (2020).** Textual evidence for the perfunctoriness of independent medical reviews. CEUR Workshop Proceedings, Vol. 2657, pp. 1–9.
8. **Castillo, O., Cervantes, L., Soria, J., Sanchez, M., Castro, J. R. (2016).** A generalized type-2 fuzzy granular approach with applications to aerospace. Information Sciences, Vol. 354, pp. 165–177.
9. **Chan, V. K. H., Chan, C. W. (2020).** Towards explicit representation of an artificial neural network model: Comparison of two artificial neural network rule extraction approaches. Petroleum, Vol. 6, No. 4, pp. 329–339.
10. **Chen, Y., Miao, D. (2020).** Granular regression with a gradient descent method. Information Sciences, Vol. 537, pp. 246–260.
11. **Chen, Y., Zhu, S., Li, W., Qin, N. (2021).** Fuzzy granular convolutional classifiers. Fuzzy Sets and Systems.
12. **Cheng, M.-Y., Tsai, H.-C., Ko, C.-H., Chang, W.-T. (2008).** Evolutionary fuzzy neural inference system for decision making in geotechnical engineering. Journal of Computing in Civil Engineering, Vol. 22, No. 4, pp. 272–280.
13. **Cui, H., Yue, G., Zou, L., Liu, X., Deng, A. (2021).** Multiple multidimensional linguistic reasoning algorithm based on property-oriented linguistic concept lattice. International Journal of Approximate Reasoning, Vol. 131, pp. 80–92.
14. **De Medeiros, I. B., Soares Machado, M. A., Damasceno, W. J., Caldeira, A. M., Dos Santos, R. C., Da Silva Filho, J. B. (2017).** A Fuzzy Inference System to Support Medical Diagnosis in Real Time. **Ahuja V., S. Y., Deepak, D. S., Berg, D., Tian, Y., Tien, J. M., Abidi, N.**, editors,

Procedia Computer Science, volume 122, Elsevier B.V., pp. 167–173.

15. **Deshpande, S. U., Bhosale, S. S. (2013).** Adaptive neuro-fuzzy inference system based robotic navigation. 2013 IEEE International Conference on Computational Intelligence and Computing Research, IEEE ICCIC 2013, IEEE Computer Society.
16. **Ding, W., Wang, J., Wang, J. (2020).** Multi-granulation consensus fuzzy-rough based attribute reduction. Knowledge-Based Systems, Vol. 198.
17. **Ding, X., Zeng, Z., Lun, L. (2010).** Granular neural networks computing on fuzzy information table. Proceedings - 2010 7th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2010, volume 1, pp. 412–418.
18. **Gaeta, A., Loia, V., Orciuoli, F. (2021).** A comprehensive model and computational methods to improve Situation Awareness in Intelligence scenarios. Applied Intelligence, Vol. 51, No. 9, pp. 6585–6608.
19. **Gayathri, B. M., Sumathi, C. P. (2016).** Mamdani fuzzy inference system for breast cancer risk detection. **Karthikeyan M., K. N.**, editor, 2015 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2015, Institute of Electrical and Electronics Engineers Inc.
20. **Ghiasi, B., Sheikhian, H., Zeynolabedin, A., Niksokhan, M. H. (2020).** Granular computing-neural network model for prediction of longitudinal dispersion coefficients in rivers. Water Science and Technology, Vol. 80, No. 10, pp. 1880–1892.
21. **Guan, Q., Guan, J. H. (2014).** Knowledge acquisition of interval set-valued based on granular computing. Applied Mechanics and Materials, Vol. 543-547, pp. 2017–2023.
22. **Guo, H., Wang, W. (2019).** Granular support vector machine: a review. Artificial Intelligence Review, Vol. 51, No. 1, pp. 19–32.
23. **Guo, S., Zhao, H. (2021).** Hierarchical classification with multi-path selection based on granular computing. Artificial Intelligence Review, Vol. 54, No. 3, pp. 2067–2089.
24. **Honka, A. M., Van Gils, M. J., Pärkkä, J. (2011).** A personalized approach for predicting the effect of aerobic exercise on blood pressure using a Fuzzy Inference System. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, pp. 8299–8302.
25. **Hu, X., Pedrycz, W., Wu, K., Shen, Y. (2021).** Information granule-based classifier: A development of granular imputation of missing data. Knowledge-Based Systems, Vol. 214.
26. **Jiang, F., Chen, Y.-M. (2015).** Outlier detection based on granular computing and rough set theory. Applied Intelligence, Vol. 42, No. 2, pp. 303–322.
27. **Leite, D., Costa, P., Gomide, F. (2013).** Evolving granular neural networks from fuzzy data streams. Neural Networks, Vol. 38, pp. 1–16.
28. **Li, D., Miao, D., Du, W. (2006).** Application of granular computing to artificial neural network. Tongji Daxue Xuebao/Journal of Tongji University, Vol. 34, No. 7, pp. 960–964.
29. **Li, W., Luo, Y., Tang, C., Zhang, K., Ma, X. (2021).** Boosted Fuzzy Granular Regression Trees. Mathematical Problems in Engineering, Vol. 2021, pp. 9958427.
30. **Li, W., Ma, X., Chen, Y., Dai, B., Chen, R., Tang, C., Luo, Y., Zhang, K. (2021).** Random Fuzzy Granular Decision Tree. Mathematical Problems in Engineering, Vol. 2021, pp. 5578682.
31. **Liu, H., Gegov, A., Cocea, M. (2016).** Rule-based systems: a granular computing perspective. Granular Computing, Vol. 1, No. 4, pp. 259–274.
32. **Mencar, C., Fanelli, A. M. (2008).** Interpretability constraints for fuzzy information granulation. Information Sciences, Vol. 178, No. 24, pp. 4585–4618.
33. **Moreno, J. E., Sanchez, M. A., Mendoza, O., Rodríguez-Díaz, A., Castillo, O., Melin, P., Castro, J. R. (2020).** Design of an interval Type-2 fuzzy model with justifiable uncertainty. Information Sciences, Vol. 513, pp. 206–221.
34. **Panoutsos, G., Mahfouf, M. (2007).** Information fusion using Granular Computing Neural-Fuzzy Networks and expert knowledge. 2007 European Control Conference, ECC 2007, Institute of Electrical and Electronics Engineers Inc., pp. 776–782.
35. **Panoutsos, G., Mahfouf, M., Mills, G. H., Brown, B. H. (2010).** A generic framework for enhancing the interpretability of granular computing-based information. 2010 IEEE International Conference on Intelligent Systems, IS 2010 - Proceedings, pp. 19–24.
36. **Pedrycz, A., Hirota, K., Pedrycz, W., Dong, F. (2012).** Granular representation and granular computing with fuzzy sets. Fuzzy Sets and Systems, Vol. 203, pp. 17–32.

37. **Pedrycz, W. (2010).** Human centricity in computing with fuzzy sets: An interpretability quest for higher order granular constructs. Journal of Ambient Intelligence and Humanized Computing, Vol. 1, No. 1, pp. 65–74.

38. **Pedrycz, W., Homenda, W. (2013).** Building the fundamentals of granular computing: A principle of justifiable granularity. Applied Soft Computing Journal, Vol. 13, No. 10, pp. 4209–4218.

39. **Pedrycz, W., Vukovich, G. (2001).** Granular neural networks. Neurocomputing, Vol. 36, No. 1-4, pp. 205–224.

40. **Peters, G., Lacic, Z. (2012).** Tackling outliers in granular box regression. Information Sciences, Vol. 212, pp. 44–56.

41. **Pota, M., Esposito, M., De Pietro, G. (2017).** Designing rule-based fuzzy systems for classification in medicine. Knowledge-Based Systems, Vol. 124, pp. 105–132.

42. **Qian, Y., Zhang, H., Sang, Y., Liang, J. (2014).** Multigranulation decision-theoretic rough sets. International Journal of Approximate Reasoning, Vol. 55, No. 1 PART 2, pp. 225–237.

43. **Sanchez, M. A., Castillo, O., Castro, J. R., Melin, P. (2014).** Fuzzy granular gravitational clustering algorithm for multivariate data. Information Sciences, Vol. 279, pp. 498–511.

44. **Solis, A. R., Panoutsos, G. (2013).** Granular computing neural-fuzzy modelling: A neutrosophic approach. Applied Soft Computing Journal, Vol. 13, No. 9, pp. 4010–4021.

45. **Sugeno, M., Kang, G. T. (1988).** Structure identification of fuzzy model. Fuzzy Sets and Systems, Vol. 28, No. 1, pp. 15–33.

46. **Truong, H. Q., Ngo, L. T., Pham, L. T. (2019).** Interval type-2 fuzzy possibilistic c-means clustering based on granular gravitational forces and particle swarm optimization. Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 23, No. 3, pp. 592–601.

47. **Tuan, T. M., Lan, L. T. H., Chou, S.-Y., Ngan, T. T., Son, L. H., Giang, N. L., Ali, M. (2020).** M-CFIS-R: Mamdani complex fuzzy inference system with rule reduction using complex fuzzy measures in granular computing. Mathematics, Vol. 8, No. 5.

48. **Vasilev, N., Mincheva, Z., Nikolov, V. (2020).** Decision tree extraction using trained neural network. SMARTGREENS 2020 - Proceedings of the 9th International Conference on Smart Cities and Green ICT Systems, pp. 194–200.

49. **Wang, W., Xiong, S. (2013).** Research of logical reasoning and application based on granular computing rough sets. Advanced Materials Research, Vol. 622, pp. 1877–1881.

50. **Wu, Q., Wang, P., Huang, X., Yan, S. (2005).** Adaptive discretizer for machine learning based on granular computing and rough sets. 2005 IEEE International Conference on Granular Computing, volume 2005, pp. 292–295.

51. **Xie, K., Xie, J., Du, L., Xu, X. (2009).** Granular computing and neural network integrated algorithm applied in fault diagnosis. 6th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2009, volume 1, pp. 188–191.

52. **Xiong, C., Qian, W., Wang, Y., Huang, J. (2021).** Feature selection based on label distribution and fuzzy mutual information. Information Sciences, Vol. 574, pp. 297–319.

53. **Xu, J., Miao, D., Zhang, Y., Zhang, Z. (2017).** A three-way decisions model with probabilistic rough sets for stream computing. International Journal of Approximate Reasoning, Vol. 88, pp. 1–22.

54. **Xu, X., Wang, G., Ding, S., Jiang, X., Zhao, Z. (2015).** A new method for constructing granular neural networks based on rule extraction and extreme learning machine. Pattern Recognition Letters, Vol. 67, pp. 138–144.

55. **Xue, M., Duan, X., Liu, W., Ren, Y. (2020).** A semantic facial expression intensity descriptor based on information granules. Information Sciences, Vol. 528, pp. 113–132.

56. **Yan, E., Song, J., Ren, Y., Zheng, C., Mi, B., Hong, W. (2020).** Construction of three-way attribute partial order structure via cognitive science and granular computing. Knowledge-Based Systems, Vol. 197.

57. **Yang, J.-G., Kim, J.-K., Kang, U.-G., Lee, Y.-H. (2014).** Coronary heart disease optimization system on adaptive-network-based fuzzy inference system and linear discriminant analysis (ANFIS-LDA). Personal and Ubiquitous Computing, Vol. 18, No. 6, pp. 1351–1362.

58. **Zadeh, L. A. (1965).** Fuzzy sets. Information and Control, Vol. 8, No. 3, pp. 338–353.

59. **Zadeh, L. A. (1975).** The concept of a linguistic variable and its application to approximate reasoning–I. Information Sciences, Vol. 8, No. 3, pp. 199–249.

60. **Zein-Sabatto, S., Mikhail, M., Bodruzzaman, M., DeSimio, M., Derriso, M. (2013).** Multistage fuzzy inference system for decision making and fusion in fatigue crack detection of aircraft structures. AIAA Infotech at Aerospace (I at A) Conference.
61. **Zhang, X., Miao, D. (2014).** Quantitative information architecture, granular computing and rough set models in the double-quantitative approximation space of precision and grade. Information Sciences, Vol. 268, pp. 147–168.
62. **Zhang, Y., Zhu, X., Huang, Z. (2009).** Fuzzy sets based granular logics for granular computing. Proceedings - 2009 International Conference on Computational Intelligence and Software Engineering, CiSE 2009.
63. **Zhang, Z.-J., Huang, J., Wei, Y. (2015).** FI-FG: Frequent item sets mining from datasets with high number of transactions by granular computing and fuzzy set theory. Mathematical Problems in Engineering, Vol. 2015.
64. **Zhou, D., Dai, X. (2015).** Combining granular computing and RBF neural network for process planning of part features. International Journal of Advanced Manufacturing Technology, Vol. 81, No. 9-12, pp. 1447–1462.
65. **Zhou, J., Lai, Z., Gao, C., Miao, D., Yue, X. (2018).** Rough possibilistic C-means clustering based on multigranulation approximation regions and shadowed sets. Knowledge-Based Systems, Vol. 160, pp. 144–166.
66. **Zhou, P., Xu, Z., Zhao, J., Song, C., Shao, Z. (2021).** Long-term hybrid prediction method based on multiscale decomposition and granular computing for oxygen supply network. Computers and Chemical Engineering, Vol. 153.
67. **Zhu, X., Pedrycz, W., Li, Z. (2018).** Granular representation of data: A design of families of epsilon-Information granules. IEEE Transactions on Fuzzy Systems, Vol. 26, No. 4, pp. 2107–2119.
68. **Zhu, X., Pedrycz, W., Li, Z. (2021).** A Development of Granular Input Space in System Modeling. IEEE Transactions on Cybernetics, Vol. 51, No. 3, pp. 1639–1650.
69. **Ziqi, Z., Xinting, T., Xiaofeng, Z., Hongjiang, G., Kun, L. (2017).** Research of Rough Set Model under Logical Computing of Granular. Proceedings - 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017, volume 1, Institute of Electrical and Electronics Engineers Inc., pp. 333–336.

Article received on 29/06/2021; accepted on 17/11/2021.
Corresponding author is Raúl Navarro-Almanza.

ISSN 2007-9737

Hybrid Quantum Genetic Algorithm for the 0-1 Knapsack Problem in the IBM Qiskit Simulator

Enrique Ballinas, Oscar Montiel

Instituto Politécnico Nacional,
Centro de Investigación y Desarrollo en Tecnología Digital,
Mexico

lballinas@citedi.mx, oross@ipn.mx

Abstract. In this work, a novel Hybrid Quantum Genetic Algorithm (HQGA) for the 0-1 Knapsack Problem (KP) is presented. It is based on quantum computing principles, such as qubits, superposition, and entanglement of states. The HQGA was simulated in the Qiskit simulator. Qiskit simulator is a platform developed by IBM that allows working with quantum computers at the level of circuits, pulses, and algorithms. The performance of HQGA is evaluated in three strongly correlated KP data sets, and computational results are compared with a Quantum-Inspired Evolutionary Algorithm (QIEA), a modified version of a QIEA (QIEA-Q), and a modified version of the HQGA (HQGA-Q). Experimental results demonstrate that the proposed HQGA can obtain the best solutions in all the KP data sets, and performs well on robustness.

Keywords. Quantum computing, quantum genetic algorithm, knapsack problem.

1 Introduction

Solving combinatorial optimization problems using classical exact algorithms becomes infeasible when the number of instances reaches a tractable limit for classical computation since search space for candidates' solutions tends to grow exponentially [48]. The complication surge when it is inevitable to handle larger values of instances that exceed this limit, which is very common in real-life applications. For example, in energy systems optimization, where there are several challenges for classical computation that can be tackle with quantum computing, such as the problem of Heat exchanger network synthesis (HENS) where a simple HENS subproblem face the programmer with an NP-hard problem [2]; or in the financial market, the transaction settlement problem is also a good example [8]. At present, it is common to handle these problems using approximation algorithms; they have the characteristics of given an approximate solution to a particular kind of problem.

They search for a solution very close to the optimal one in polynomial time according to the sizes of their inputs instead of looking for the optimal solution, which can cause the search time grows exponentially [31]. They are divided into three types, heuristic, meta-heuristic, and hyper-heuristic. Heuristic methods are based on experience that allows finding a satisfactory solution in a reasonable time. Meta-heuristics are those algorithms that are one level higher than heuristics; that is, they are procedures that seek to find a solution to a problem using the least amount of computational resources than heuristic algorithms [11].

Hyper-heuristics can be viewed as search algorithms that explore the space of problem solvers. A hyper-heuristic is a heuristic search method that seeks to automate the process of selecting, combining, generating, or adapting several simple heuristics in order to solve a problem efficiently [11]. The importance of meta-heuristic algorithms was fully understood when the NP-completeness theory was established in 1971 [31]. This theory determined that many of the known problems in state of the art are intractable, which means

that it is not possible to find an optimal solution in a polynomial-time [10]. As a consequence, approximation algorithms were the best option to solve these kinds of problems. The approximation algorithms have experienced significant growth in the last years due to the development of areas such as data mining, bioinformatics, deep learning, and others; this caused a significant number of optimization problems to be born.

Quantum computing offers an exponential speedup for solving some problems that can take advantage of quantum phenomena in their algorithmic formulation. This feature is desirable for solving problems where classical and approximation algorithms cannot offer practical or viable solutions to complex problems that have grown very fast and become unsolvable.

The contribution of this paper is the design of a Hybrid Quantum Genetic Algorithm (HQGA) and its implementation in a quantum simulator (Qiskit IBM simulator). So far, there is no work in state-of-the-art where a Quantum Genetic Algorithm in the IBM Qiskit simulator can solve the Knapsack Problem (KP). There is only one work [29] where Adiabatic Quantum computation was implemented in the Qiskit simulator to solve the Binary Knapsack Problem (it is explained in section 2). Also, a quantum circuit was designed which allows to generate the initial population of qubits, with this, the diversity of the algorithm is expanded since when using qubits, the possible configurations increase exponentially just by adding a qubit to the quantum population. For example, with a population of 10 qubits we would have $2^{10} = 1024$ possible configurations, with 11 qubits $2^{11} = 2048$ possible configurations we would have.

A modified version of the HQGA (HQGA-Q), a QIEA, and a modified version of the QIEA (QIEA-Q) was also designed. HQGA-Q and QIEA-Q were designed to solve an impediment of the Qiskit simulator; in section 4 it will be explained. The HQGA, based on the statistical results of hundreds of tests performed, has proven to outperform the aforementioned quantum evolutionary algorithms. The main advantage of HQGA compared to state-of-the-art algorithms is the implementation of a quantum circuit in a quantum genetic algorithm, which takes advantage of one of the main characteristics of quantum computing, quantum parallelism, with this we can represent n possible configurations with just a quantum register of n qubits.

The organization of this paper is as Follows. Section 1 presents an introduction to approximation algorithms and the importance of meta-heuristics algorithms in solving combinatorial optimization problems. Section 2 presents the main works that use quantum meta-heuristics in solving the knapsack problem. Section 3 describes the main concepts involved in the development of this work. Section 4 shows the experiments carried out and their corresponding results in solving the 0-1 knapsack problem. The conclusions and future work are presented in section 5.

2 Related Work

The knapsack problem (KP) is a widely studied combinatorial optimization problem with NP-hard computational complexity [22, 14, 52, 26]. In the literature, there are several works that have solved the 0-1 Knapsack problem using different methods, such as, Binary Cuckoo Search Algorithm (CSA) [6], Firefly Algorithm (FA) [7], Ant Colony Algorithm (ACO) [42], Comparative study between Tabu Search Algorithm (TS), Sparse Search (SS), and Local Search [41], Comparative study between Simulated annealing algorithm (SA), Iterative local search, Genetic algorithm (GA), and Particle swarm optimization algorithm (PSO) [1], among others.

The methods mentioned above are evolutionary and non-evolutionary algorithms that have been widely used for solving combinatorial optimization problems, showing satisfactory results using conventional computers. However, it is possible to improve these results using quantum algorithms.

Quantum computing is a computational model that uses certain quantum mechanics concepts, such as superposition, entanglement, and qubits. When it is combined with genetic algorithms, quantum genetic algorithms are created [49, 15].

Quantum genetic algorithms have demonstrated the potential to tackle NP-hard problems, even showing better results than classical algorithms.

For example, in [17], a series of experiments using a Parallel Quantum-Inspired Genetic Algorithm, Quantum-Inspired Genetic Algorithm, and a Classical Genetic Algorithm are presented. The results demonstrate the Parallel Quantum-Inspired Genetic Algorithm's superiority over the other two solving the 0-1 Knapsack Problem. The use of parallel computing helps to improve the exploitation and exploration capabilities.

In [46], a Higher-order Quantum Genetic Algorithm to solve the 0-1 Knapsack Problem with 200, 500, and 1000 objects is proposed. The algorithm uses high-order quantum registers, which consists in dividing the register into sub-registers. This reduced the algorithm's execution time compared to the traditional Quantum Genetic Algorithm (QGA) and maintained the same precision, even increasing it in some registers sizes.

In [45], a Quantum Genetic Algorithm (QGA) to solve the Knapsack Problem is implemented. QGA uses an adaptive quantum gate to find the rotation angle's correct value, reducing the qubits states' amplitudes with respect to the previous results. The experimental results showed that the use of the adaptive quantum gate generates a fast local convergence by solving the Knapsack Problem with different quantities of objects.

Another type of meta-heuristic that has proven to be efficient in solving the NP-hard problems, especially the Knapsack Problem, is the Quantum Evolutionary Algorithms. For example, in [43] a Self-organizing Quantum Evolutionary Algorithm for Multi-objective optimization (MSQEA) that solves the Multi-objective Knapsack Problem is designed. The experimental results showed that the MSQEA could obtain solutions very close to the Pareto optimal front in a short time, in addition to generating larger sets of non-dominating points.

In [51] an Improved Quantum Evolutionary Algorithm (IQEA) is proposed; this algorithm's main feature is using the rotating quantum gate that generates a faster convergence and a better global search for solutions. The algorithm was compared against an Evolutionary Quantum Algorithm (QEA), demonstrating its superiority in efficiency and quality when solving the Knapsack Problem.

In [27], a Quantum Evolutionary Algorithm to solve the Quadratic Knapsack Problem (QKP) is designed. The qubits are initialized according to the density values (this value is obtained by dividing the sum of the gains of all objects and the object's weight) and are expressed in angles; this proposal was called Angle-expressed Quantum Evolutionary Algorithm (AQEA). AQEA was compared with a Classic Genetic Algorithm (CGA) and a Quantum Evolutionary Algorithm (QEA), demonstrating its effectiveness and better convergence when solving QKP with 100, 200, and 300 objects.

In [20], a Quantum-Inspired Evolutionary Algorithm to solve the Multidimensional Knapsack Problem is designed. Experiments were carried out with different repair functions: Simple, Random, and Sorted. The results showed that the Sorted repair function has a faster convergence time; however, having to sort the objects beforehand would be a great effort if the data set is huge.

One of the important characteristics of quantum computing is the versatility of the algorithm to be able to adapt to different meta-heuristics and search algorithms. For instance, in [24], a Diversification-based Quantum Particle Swarm Optimization Algorithm (DQPSO) to solve the Multidimensional Knapsack Problem is presented. DQPSO is based on the Quantum Particle Swarm Optimization Algorithm (QPSO) and a population-based diversification criteron, which generates better diversity than QPSO. The experiments were carried out using 30 instances showing the efficiency of DQPSO.

In [25], a Quantum Particle Swarm Optimization Algorithm with a preserving strategy (Diversity-preserving QPSO) to solve the Multidimensional Knapsack Problem is presented. The Diversity-preserving QPSO has a diversity strategy to update and maintain a good diversity in the population, and a method called Variable Neighborhood Descending (VND) which improves the search process to find the optimal solution. The results demonstrated the efficiency of the Diversity-preserving QPSO compared to the state-of-the-art algorithms.

In [34], an algorithm called Binary Quantum Inspired Gravitational Search Algorithm (BQIGSA) is presented; it combines the properties of the Gravitational Search Algorithm (GSA) and quantum algorithms; some experiments were carried out with the Max-one, Royal-road functions, and Knapsack Problem. The results were compared with the Binary Gravitational Search Algorithm (BGSA), Conventional Genetic Algorithm (CGA), Binary Particle Swarm Algorithm (BPSO), a modified version of the BPSO, a new version of the Binary Differential Evolution Algorithm, Quantum-Inspired Particle Swarm Algorithm, and three Quantum-Inspired Evolutionary Algorithms. For the problems mentioned above, the BQIGSA algorithm presented the best results.

In [18], A Binary Multi-scale Quantum Harmonic Oscillator Algorithm (BMQHOA) to solve the Knapsack Problem is designed. BMQHOA is inspired by the probabilistic interpretation of the wave function and using properties such as quantum tunnel effect avoids local optimum. Several experiments were carried out with the following algorithms: Binary Bat Algorithm (BBA), Binary Dragonfly Algorithm (BDA), Binary Particle Swarm Algorithm (BPSO) and Binary Particle Swarm with Gravitational Search Algorithm (BPSOGSA). The results showed the superiority of BMQHOA in precision, convergence, and stability compared to the algorithms mentioned above.

In [13], a Quantum-inspired Wolf pack Algorithm to solve the 0-1 Knapsack Problem is presented. The algorithm is based on the behavior of the wolf pack when hunting; this approach uses quantum gates to update the position of the solutions. Experiments were carried out with 100, 250, 500, and 1000 dimensions. Compared with other algorithms in the literature, the results showed the proposed algorithm's effectiveness, especially for large cases.

In [12], a Quantum Annealing Algorithm (QA) that uses parallel computing properties to solve the Multidimensional Knapsack Problem was introduced. QA is a technique derived from Simulated Annealing Algorithm (SA). The experiments with 500 objects showed that QA outperforms its non-parallel version.

We end this section with a work where Adiabatic Quantum Computing (AQC) was used to solve the Binary Knapsack Problem using the libraries of IBM Qiskit simulator [29]. AQC is considered a particular class of Quantum Annealing (QA), which uses quantum mechanics properties to solve optimization problems without restrictions. The results obtained show the quantum algorithm's effectiveness and a slight superiority compared to its classical counterpart.

3 Theoretical Framework

3.1 The Knapsack Problem

One of the most studied combinatorial optimization problems is the knapsack problem, which is computationally challenging because it is NP-hard. [14, 35, 22]. The problem consists in given a set of objects, each with a weight w_i and value p_i, to determine which objects should be part of a collection with the condition that the weight WX is less than or equal to a certain limit, and the total value PX is what largest possible [52].

The mathematical representation of the model is presented below:

$$\begin{aligned} &\text{maximize } f(x_1, x_2, \ldots, x_n) = PX = \sum_{i=1}^{n} p_i x_i, \\ &\text{subject to } WX = \sum_{i=1}^{n} w_i x_i \leq V, \end{aligned} \tag{1}$$

$x_j \in \{0, 1\}$, $j = 1, 2, \ldots, n$. Here $P = (p_1, p_2, \ldots, p_n)$, $W = (w_1, w_2, \ldots, w_n)$ represent the vector of values and the vector of weights of all objects respectively. V is the maximum capacity of the knapsack, $x_i = 1$ indicates that the object i is inside of the knapsack and $x_i = 0$ that it is not.

3.2 Quantum Computing

A classical computer uses bits to store information, whereas a quantum computer uses quantum bits or qubits. Qubits are a unit of information that describes a two-dimensional quantum system [49]. An important characteristic of quantum computation is that the qubit can be in a state of superposition, that is, the qubit can be in the $|0\rangle$

and $|1\rangle$ state simultaneously. Mathematically this is represented as a matrix of complex numbers:

$$|\psi\rangle = a|0\rangle + b|1\rangle \equiv \begin{pmatrix} a \\ b \end{pmatrix}, \tag{2}$$

where $|a|^2+|b|^2= 1$. Thus $|a|^2$ and $|b|^2$ represent the probability of finding the qubit after being measured in the state $|0\rangle$ and $|1\rangle$ respectively [49, 28].

Dirac's notation allows describing the state of a quantum system formally. For each "ket" $|\psi\rangle$ there is a corresponding "bra" $\langle\psi|$. The ket and the bra contain equivalent information about the quantum state. Mathematically, they are dual with each other, that is:

$$\langle\psi| = a^*\langle 0| + b^*\langle 1| = (a^*b^*). \tag{3}$$

Just as a single qubit can be found in superposition of the possible states $|0\rangle$ and $|1\rangle$, a register of n-qubits can be found in superposition of all 2^n possible states $|00\ldots0\rangle, |00\ldots1\rangle, \ldots, |11\ldots1\rangle$. For example the general form of the state of a 2 qubit quantum memory register is represented as:

$$|\psi\rangle = c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle, \tag{4}$$

where $|c_0|^2+|c_1|^2+|c_2|^2+|c_3|^2= 1$. This implies that we can have a register that contains many different bit strings, each with its corresponding amplitude value.

The general form of a n-qubit quantum memory register is:

$$\begin{aligned} |\psi\rangle = c_0|00\ldots0\rangle + c_1|00\ldots1\rangle + \ldots \\ + c_{2^n-1}|11\ldots1\rangle = \sum_{i=0}^{2^n-1} c_i|i\rangle, \end{aligned} \tag{5}$$

where $\sum_{i=0}^{2^n-1}|c_i|^2= 1$ and $|i\rangle$ represents the eigen state of the computational base whose bit values match those of the decimal number expressed in base 2 notation, padded on the left (if necessary) with bits "0" to make a full complement of n bits.

3.3 Quantum Inspired Algorithms

The concepts and principles of quantum mechanics to develop more efficient evolutionary computing methods were introduced by Narayanan and Moore in 1996 [33]. The main objective was to compare the performance of a classic algorithm and quantum-inspired algorithm in the Traveling Salesman Problem (TSP). They use an interference crossover operator, which consists of taking the first element of chromosome one, the second element of chromosome two, the third element of chromosome three, and so on; if an element already exists on the chromosome, another element that does not exist on the chromosome is chosen. The results showed that the quantum-inspired genetic algorithm outperformed the classical version.

In [32], a basic methodological principle to design a quantum algorithm was presented. The main objective was to identify the novelty and potential of the quantum algorithm in tackling NP-hard problems.

The two first pioneer works on quantum computing are Genetic Quantum Algorithm (GQA) proposed by [15] and Quantum Inspired Evolutionary Algorithm (QEA) introduced in [16]. GQA is based on quantum computing concepts and principles such as qubits and superposition instead of binary, numeric, or symbolic representation. GQA [15] demonstrated its effectiveness and applicability by experimental results on the 0-1 Knapsack Problem. All the experiments were simulated only in classic computers.

Quantum genetic algorithms have an excellent ability to perform global searches due to thier diversity in the population caused by the probabilistic representation; this characteristic allows for finding better solutions in a shorter time than the classical algorithms [17]. For example, with a single quantum register of three qubits, it is possible to represent eight states $(2^3 = 8)$; to represent eight states with a classical algorithm, eight registers would be needed. For large instances of N (large problems), a quantum computer requires only a 32-qubit register to handle all the possible combinations of 32-bit numbers, whereas a classical computer

requires 4,294,967,296 memory registers, which is significantly large.

The QEA [16] is the upgrade of the GQA [15], like the GQA, the QEA was designed to solve the 0-1 Knapsack Problem. The process to find the optimal solution in both algorithms is similar. The main difference between both proposals is the concept of migration introduced by QEA, which is a process that can induce a variation of the probabilities of a quantum chromosome [31].

The interaction between quantum computing and evolutionary computation can be addressed in three different ways [50]: The first is the Evolutionary-Designed Quantum Algorithm (EDQAs); here, the idea is to use genetic programming to generate new quantum algorithms. The second way is to use Quantum Evolutionary Algorithms (QEAs), focusing on developing evolutionary algorithms for quantum computers. The third way is Quantum-Inspired Evolutionary Algorithms (QIEAs), which use quantum mechanics concepts such as qubits, superposition, quantum gates, and quantum measurements to develop evolutionary methods for classic computers; a novel proposal in this category is the Quantum Inspired Acromyrmex Evolutionary Algorithm (QIAEA) [30].

Nowadays, companies such as D-wave [23], Google [3], and IBM [37] have created quantum computers with the ability to solve some problems that today's classical computers cannot. With the arrival of quantum supremacy (a term coined by John Preskill [39]) announced by Google in 2019 [3], it was mentioned that any problem that a classical computer could not solve in polynomial time could be solved by a quantum computer. However, In [40] mentioned that we are currently in the NISQ (Noisy Intermediate-Scale Quantum) era. *Intermediate scale* refers to the size of quantum computers available at present, and *Noisy* emphasizes that we will have imperfect control over those qubits. At present, the hardware for controlling trapped ions [4], or superconducting circuits [5], the error rate per gate for two-qubit gates is above the 0.1% [40], which means that with a sequence of 1000 quantum gate operations, the error rate in the circuit would be 100%, which would not give reliable results at all.

3.4 IBMQ Framework

The IBMQ framework is a software development kit (SDK) for performing quantum computations that utilize quantum mechanical principles such as superposition and entanglement. It allows the development of hybrid quantum computing algorithms that is an essential issue for our proposal. All the experiments were designed and run on the Qiskit platform, an open-source framework computational platform for working with quantum computers at the level of circuits, pulses, and algorithms. Qiskit is made up of four fundamental elements [19]. They are: 1) **Terra** that provides a base for composing quantum programs at the circuit level and pulses; with this module, we can perform optimizations for the constraints of a specific device. 2) **Aer** that allows accelerating the development of applications via simulators and noise models. 3) **Ignis** dedicated to fighting noise and errors; it is meant for those who want to work designing quantum error correction codes. 4) **Aqua** is where algorithms for quantum computing are built; this module focuses on constructing solutions for real-world application problems.

3.5 Hybrid Quantum Genetic Algorithm for solving the Knapsack Problem in the IBM-Q quantum computer

This section will describe our proposal to solve the knapsack problem using a quantum hybrid genetic algorithm. Hybrid computation in the NISQ era is an good way to perform quantum computing to no loose coherency, especially when the length of quantum circuits is high, which is the case of quantum meta-heuristics.

Figure 1 shows a block diagram where a quantum hybrid genetic algorithm is depicted. On the right side are the computation steps that are executed by the classical computer. On the left side are the steps that are performed by the quantum computer. As was mentioned, this model of computation will help us to maintain quantum circuits of small lengths. To understand better this figure is convenient to use Algorithm 1 and Algorithm 2. The computational steps that are achieved by the quantum computer are 3, 5, and 8 of Algorithm 1; the CC executes the remaining.

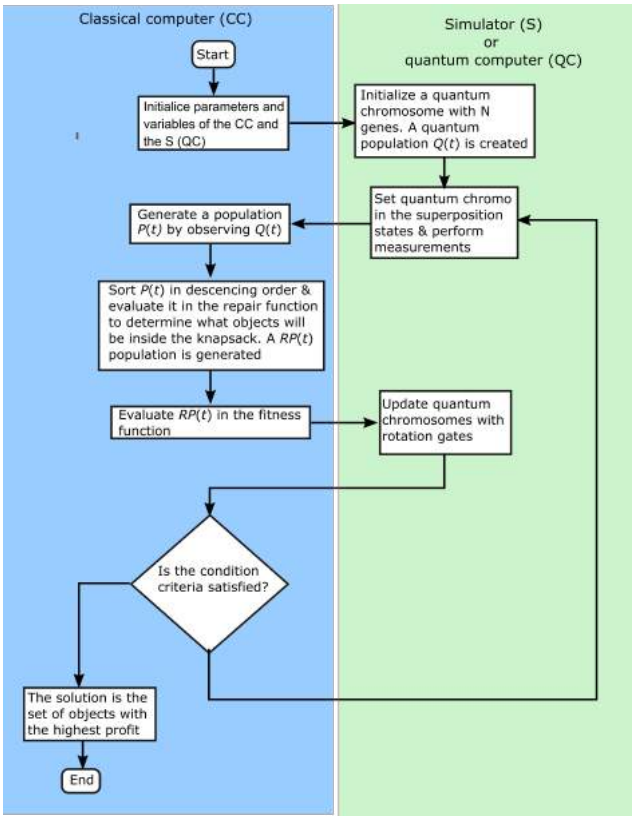


Fig. 1. Hybrid genetic algorithm. On the right side are the classical computer's steps. On the left side, the steps executed by the quantum computer or quantum simulator. In the figure, RP is the classic population after having been evaluated in the repair function

The CC performs the HQGA's parameter initialization for both systems, the classical and the QC. In this first stage, the quantum chromosome length is defined, and this information is sent to the QC. In the QC, the quantum chromosome $|\Psi_i^t\rangle$ is set in the superposition state where a number n of measurements is achieved to create an initial classical population P. In the CC, this population is repaired, evaluated, and sorted in descending order.

Algorithm 1 describes our algorithmic implementation proposal of the Hybrid Quantum Genetic Algorithm (HQGA) to solve the knapsack problem. The input of the HQGA is a set of quantum chromosomes; i.e, the initial quantum population is defined as $Q(t) = [\Psi_1^t, \Psi_2^t, \ldots, \Psi_n^t]$, where n is

Algorithm 1 Hybrid Quantum Genetic Algorithm

Input: The number n of quantum chromosomes, and the maximal number of generations MAX_GEN

Output: The best solution b

1: **Begin**
2: $t \longleftarrow 0$;
3: Initialize $Q(t)$
4: **while** $t < MAX\_GEN$ **do**
5: Measure $Q(t)$ to generate $P(t)$
6: Repair $P(t)$
7: Evaluate $RP(t)$
8: Update $Q(t)$
9: Store the best solution b of $P(t)$ in $B(t)$
10: **End**

the size of the population and t is the generation number. A quantum chromosome is defined as follows:

$$|\Psi_i^t\rangle = \left[\begin{array}{c|c|c|c} \alpha_{i,1}^t & \alpha_{i,2}^t & \cdots & \alpha_{i,m}^t \\ \beta_{i,1}^t & \beta_{i,2}^t & \cdots & \beta_{i,m}^t \end{array} \right], \qquad (6)$$

where m is the number of qubits and $i = 1, 2, \ldots, n$. The length of a qubit string is the same as the number of items.

The algorithm's output will be the best classical solution through generation saved in $B(t)$. In a similar fashion to any evolutionary algorithm, we started by using a generation counter t.

In step 2, the generation counter is initialized. In Step 3, we initialized the quantum chromosomes $Q(t)$ in the zero states, and then, they were put in the superposition state using the Hadamard gate; i.e.; for each quantum chromosome, we performed the next quantum operation $|\Psi_i\rangle|\otimes|H^{\otimes n}\rangle$. For the experiments, a population of one quantum chromosome was used

Step 4 is a while loop that will end when the maximum number of generations MAX_GEN has been reached. In step 5, the quantum population is observed (measured) to generate classical population (0 and 1) according to $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α represents the probability that the qubit collapses to the zero state ($|0\rangle$) and β the probability that the qubit collapses to state one ($|1\rangle$) after being measured.

Algorithm 2 Repair Algorithm

Input: Classic chromosomes $P(t) = 0 \ldots 000 + \ldots + 1 \ldots 111_{2^n-1}$, n is the number of qubits
Output: $RP(t)$

1: **Begin**
2: knapsack-full $\leftarrow$ false
3: **if** $\sum_{j=1}^{m} w_j x_j > V$ **then** knapsack-overfilled $\leftarrow$ true
4: **while** knapsack-full = true **do**
5: Select a j-th item from the knapsack
6: $x_j \leftarrow 0$
7: **if** $\sum_{j=1}^{m} w_j x_j < V$ **then** knapsack-full $\leftarrow$ false
8: **while** knapsack-full = false **do**
9: Select a j-th item from the knapsack
10: $x_j \leftarrow 1$
11: **if** $\sum_{j=1}^{m} w_j x_j > V$ **then** knapsack-full $\leftarrow$ true
12: **End**

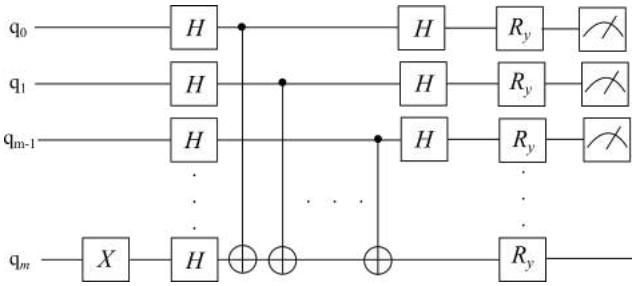


Fig. 2. Quantum circuit of the HQGA implemented in the IBM Qiskit simulator to solve the knapsack problem for 100, 250 and 500 objects

A repair algorithm was used in step 6, the main objective of this algorithm is to add or remove items from the knapsack as shown below.

The inputs of the Algorithm 2 are classic chromosomes $P(t)$ and the output will be classic repair chromosomes $RP(t)$. Algorithm 2 begins with the variable knapsack-full; this indicates if the knapsack is full (true) or not (false). If the sum of the weight w_j of each object x_j is greater than the total capacity V of the knapsack, it means that knapsack-full = true, and the algorithm continues in step 4; otherwise, it continues in step 8.

Continuing with step 7 of Algorithm 1 the profit of a solution x is evaluated by $\sum_{i=1}^{n} p_i x_i$, and it is used to find the best solution b to store in $B(t)$ (step 9) after the update of $|\Psi_i\rangle$, $i = 1, 2, \ldots, n$. A qubit chromosome $|\Psi_i\rangle$ is update (step 8) by using a $R_y(\theta)$ rotation gate. The j-th qubit value (α_j, β_j) is update as:

$$\begin{bmatrix} \alpha'_j \\ \beta'_j \end{bmatrix} = \begin{bmatrix} \cos(\theta_j) & -\sin(\theta_j) \\ \sin(\theta_j) & \cos(\theta_j) \end{bmatrix} \begin{bmatrix} \alpha_j \\ \beta_j \end{bmatrix}. \quad (7)$$

4 Experiments and Results

Figure 2 shows the quantum circuit of the HQGA in the Qiskit simulator (ibm_qasm_simulator) to solve the Knapsack Problem with 100, 250, and 500 objects. The ibm_qasm_simulator is a quantum environment designed by IBM; this simulator allows to generate circuits with a maximum of 32 qubits, which could be considered limited for the problem to be addressed. The simulator allows adding bits to the quantum register so that it is possible to use bits and qubits within the same register.

Quantum algorithms have shown that with a single evaluation of the oracle, it is possible to determine the state of the function [9, 21, 36]. The same principle can be used when solving the Knapsack Problem since with a single evaluation in the fitness function is possible to determine the profit of the 2^n possible configurations.

The circuit (see Figure 2) is composed of a Pauli-X gate (X) which serves as an ancillae gate that will enable the C-NOT (Controlled NOT) gates to be activated; these generate entanglement between each of the qubits. Hadamard gates (H) are also shown, whose main objective is to place the qubits in superposition, as well as rotation gates $(R_y(\theta))$ which are used to update the quantum population.

According to [47] one of the main characteristics to indicate the complexity of a quantum circuit is its length (number of serial gate operations after having parallelized the circuit to the maximum extent possible). The quantum circuit's length is 5 and the width (total number of qubits on which the circuit acts, including any ancillae qubit) is 21 qubits.

In [15, 17, 16] it is mentioned that the number of bits (qubits, in our case) must be equal to the number of objects in the knapsack.

Table 1. Experimental results of the knapsack problem with hybrid quantum genetic algorithm

N	Best	Worst	Average	Std
100	666.23	662.60	663.89	0.945
250	1591.03	1591.010	1591.01	0.003
500	3347.33	3340.91	3347.12	1.173

Table 2. Experimental results of hybrid quantum genetic algorithm with 100 objects

Run	Generations	Rotation angles	Best
1	125	2.0420	663.78
2	93	1.7122	663.78
3	113	2.1049	663.50
4	62	1.4608	663.85
5	93	1.7122	665.85
6	113	1.8378	663.58
7	104	1.9792	663.50
8	72	1.6650	663.85
9	117	2.2619	663.15
10	102	1.9792	663.50
11	144	2.1991	663.19
12	94	1.9792	663.13
13	122	2.2305	663.78
14	956	7.6498	663.85
15	105	2.0892	663.85
16	84	1.7436	663.19
17	234	2.6075	663.17
18	930	8.1524	663.58
19	120	1.9792	663.58
20	106	1.8221	665.85
21	120	2.0420	663.50
22	72	1.5708	665.54
23	203	2.3876	663.42
24	**97**	**1.7907**	**666.23**
25	125	2.1677	663.78
26	119	2.1677	663.17
27	121	2.1677	663.78
28	132	2.1520	665.85
29	123	1.9321	663.42
30	121	1.9949	662.60

However, the Qiskit simulator only allows a maximum of 32 qubits, but to not overload the system's RAM memory it was decided to use only 21 qubits, to complete the remaining qubits, bits were added to the quantum register.

In all the experiments, the following data sets were considered:

$$\begin{aligned} w_i &= uniformly\, random[1, 10) \\ p_i &= w_i + 5, \end{aligned} \tag{8}$$

and the average knapsack capacity was calculated as $V = 1/2 \sum_{i=1}^{m} w_i$. One of the most accepted ways to classify the complexity of an instance is the correlation between its data: the relationship or not between the weights of each object and its profit [29, 38]. This complexity is defined as: Uncorrelated, Weakly correlated, Strongly correlated, Inverse strongly correlated, Almost strongly correlated, and Subset sum. Where strongly correlated instances are hard to solve [38].

We carried out experiments to test the algorithm's performance. We ran the algorithm 30 times during 1000 generations, with strongly correlated data sets of 100, 250, and 500 objects. the IBM Qiskit simulator was used.

Table 1 shows the experimental results of HQGA solving the Knapsack Problem with 100, 250, and 500 objects. The "N" column represents the number of objects in the knapsack, the "Best", "Worst", "Average", and "Std" columns, represent the best solution, the worst solution, the average solution, and the standard deviation, respectively.

The experimental results demonstrated the robustness of the HQGA solving the Knapsack Problem with different quantities of objects because the standard deviation (Std) presented in Table 1 shows small values, even very close to zero.

Tables 2, 3, and 4 show the experiments with 100, 250, and 500 objects, respectively. For each table, the "Run" column represents the run's number, and the "Generations" column shows the generation where the best solution was obtained in each run; the "Rotation angles" column represents the angle of the rotation gate in that generation, the "Best" column shows the best solution obtained in each run.

Table 4 shows the experimental results with 500 objects. The best solution (bold letters) was in the first run and the worst solution (italic letters) in run 12. The rotation angles was 37/50π and 28/50π respectively.

Table 3. Experimental results of hybrid quantum genetic algorithm with 250 objects

Run	Generations	Rotation angles	Best
1	160	2.3562	1,591.01
2	122	2.0420	1,591.01
3	137	1.9635	1,591.01
4	134	2.1834	1,591.01
5	162	2.4347	1,591.01
6	122	1.9792	1,591.01
7	113	1.9321	1,591.01
8	126	2.0892	1,591.01
9	169	2.2934	1,591.01
10	146	2.2305	1,591.01
11	103	1.8692	1,591.01
12	137	2.1206	1,591.01
13	133	2.0263	1,591.01
14	129	2.2462	1,591.01
15	117	2.1206	1,591.01
16	143	2.3248	1,591.01
17	131	1.9792	1,591.01
18	121	1.9949	1,591.01
19	161	2.2619	1,591.01
20	121	1.9792	1,591.01
21	131	2.0577	1,591.01
22	209	2.6232	1,591.01
23	139	2.1834	1,591.01
24	200	2.3091	1,591.01
25	114	1.8064	1,591.01
26	124	2.1206	1,591.01
27	113	1.9321	1,591.01
28	**104**	**1.7593**	**1,591.03**
29	129	1.9792	1,591.01
30	153	1.9949	1,591.01

Table 4. Experimental results of hybrid quantum genetic algorithm with 500 objects

Run	Generations	Rotation angles	Best
1	**247**	**2.3248**	**3,347.33**
2	150	1.9792	3,347.33
3	191	1.9635	3,347.33
4	155	1.7593	3,347.33
5	161	1.9478	3,347.33
6	168	2.1991	3,347.33
7	163	2.0263	3,347.33
8	148	1.7279	3,347.33
9	165	1.9007	3,347.33
10	177	1.9792	3,347.33
11	149	1.8850	3,347.33
12	156	1.7750	3,340.91
13	225	2.1677	3,347.33
14	209	1.9635	3,347.33
15	257	2.1520	3,347.33
16	154	1.8692	3,347.33
17	161	1.9164	3,347.33
18	162	1.9792	3,347.33
19	225	2.0735	3,347.33
20	200	2.1049	3,347.33
21	227	2.2305	3,347.33
22	182	1.9792	3,347.33
23	174	1.9949	3,347.33
24	177	2.1991	3,347.33
25	159	2.0420	3,347.33
26	164	1.8064	3,347.33
27	244	2.3405	3,347.33
28	257	2.3405	3,347.33
29	190	2.3091	3,347.33
30	803	3.5657	3,347.33

In Table 2, the best solution was obtained in run 24 (bold letters) with a rotation angle of 14/25π and the worst in run 30 (italic letters) with a rotation angle of 16/25π.

In some runs a big difference can been seen between the rotation angles (see Table 2). For example, in run 18, the rotation gate has an angle of 8.1524 rad (13/5π), and the run 17 has an angle of 2.6 rad (41/50π); this difference is due to the intrinsic probabilistic condition of quantum algorithms and the number of bits and qubits that the quantum register has. The last characteristic will be explained later.

In Table 3, the experimental results with 250 objects is presented. The worst solution (italic letters) was obtained in run 1 with a rotation angle of 15/20π and the best solution (bold letters) in run 28 with a rotation angle of 11/20π.

The best solutions presented by Table 3 and 4 show values very similar, unlike those presented by Table 2.

This behavior is due to the number of bits (qubits) that the quantum register handles and how the rotation angle is determined; as mentioned above, the number of objects in the knapsack is equal to the number of bits and qubits that the quantum register has. For example, for 250 objects, the quantum register stores 250 bits (the qubits have already been measured). To determined the rotation angle, the $i-$th bit of the

Table 5. θ values for the rotation gates

x_i	b_i	$f(x) \geq f(b)$	$\Delta\theta$
0	0	false	0
0	0	true	0
0	1	false	0
0	1	true	0.05
1	0	false	0.01
1	0	true	0.025
1	1	false	0.005
1	1	true	0.025

Table 6. Best, worst, average solutions, and standard deviation (Std) with 100 objects. Best results in bold

Algorithm	Best	Worst	Average	Std
HQGA	**666.23**	**662.60**	**663.89**	0.945
HQGA-Q	636.09	636.09	636.09	**2.31e-13**
QIEA	663.50	656.01	659.87	2.812
QIEA-Q	660.92	599.60	632.37	19.739

current quantum register and the $i-$th bit of the best solution obtained up to that moment are compared (see Table 5), as there are more bits in the register the probabilities to obtaining large rotation angle values are lower, causing the best solutions to remain constant.

The rotation angles for all experiments were set using Table 5 and obtained from [15, 17]. Where x_i and b_i represent the i-th bits for the binary solution x and the best solution b, respectively. The third column ($f(x) \geq f(b)$) is the comparison between the evaluation of the binary solution and the evaluation of the best solution. The column $\Delta\theta$ represents the rotation value for the rotation gate.

For testing the performance of the HQGA, it was compared against a modified version of HQGA (HQGA-Q), a Quantum Inspired Evolutionary Algorithm (QIEA), and a modified version of QIEA (QIEA-Q). The quantum circuits used for the aforementioned algorithms are the same as shown in Figure 2. They were all implements in the ibm_qasm_simulator. The operation of the HQGA-Q is very similar to the HQGA; the only difference is that the HQGA-Q uses only qubits in its quantum register, while the HQGA uses bits and qubits. For example, with 100 objects, the HQGA has in its quantum register 21 qubits and 79 bits (as we mentioned at the beginning of the section, the number of objects must be equal to the number of bits and qubits), and the HQGA-Q begins with a quantum register of 5 qubits, then the quantum register is measured 20 times, at the end a population of 100 bits is obtained. The same happens between the QIEA and QIEA-Q.

The experimental results of the HQGA, HQGA-Q, QIEA, and QIEA-Q with 100, 250 and 500 objects are presented in Tables 6, 7, and 8 respectively. All the experiments were carried out in the Qiskit IBM simulator with 30 runs and 1000 generations, except for QIEA with 250 and 500 objects; in both cases, 20 runs were carried out.

The results presented in Table 6 demonstrate the superiority of the HQGA over its quantum counterpart. However, the HQGA-Q shows a lower standard deviation (Std); this is due to the number of qubits that the algorithm uses. In all experiments, five qubits were used for the HQGA-Q and the QIEA-Q. With 5 qubits $2^5 = 32$ possible solutions can be obtained, while with 21 qubits (quantity used for the HQGA and QIEA), $2^{21} = 2,097,152$ possible solutions are obtained. This indicates a greater diversity in both algorithms that generate better results.

With 250 objects (see Table 7), the HQGA outperform the rest of the algorithms (HQGA-Q, QIEA, and QIEA-Q) in all cases, presenting even a lower standard deviation, demonstrating its stability and better performance.

Table 8 shows the results with 500 objects, in this case we can see that HQGA and QIEA have the same best values, but QIEA has better standard deviation (Std), worst, and average solutions. In all the experiments HQGA and QIEA have better results than HQGA-Q and QIEA-Q.

By observing the three Tables (6, 7, and 8), we can see that the best solutions are provided by the HQGA and QIEA, as it was mentioned, both algorithms use bits and qubits in their quantum register, instead of HQGA-Q and QIEA-Q, whose algorithms use only qubits.

This shows that the use of bits and qubits in algorithms generates greater diversity allowing them to find better solutions in the search space.

Figure 3 shows the distribution in box plot of the best solutions for each of the quantum-inspired

Table 7. Best, worst, average solutions, and standard deviation (Std) with 250 objects. Best results in bold

Algorithm	Best	Worst	Average	Std
HQGA	**1591.03**	**1591.01**	**1591.01**	**0.003**
HQGA-Q	1576.2	1553.86	1562.37	5.701
QIEA	1591	1590.87	1591	0.031
QIEA-Q	1568.2	1548.08	1559.92	6.051

Table 8. Best, worst, average solutions, and standard deviation (Std) with 500 objects. Best results in bold

Algorithm	Best	Worst	Average	Std
HQGA	**3347.33**	3340.91	3347.12	1.173
HQGA-Q	3260.34	3169.92	3218.42	31.14
QIEA	3347.33	**3347.33**	**3347.33**	**1.86e-12**
QIEA-Q	3257.9	3158.16	3222.64	34.241

methodologies with 100 objects. The HQGA has a higher median demonstrating that it had greater number of better solutions than the HQGA-Q, QIEA, and QIEA-Q. Also, the range is very short comparing with the other methodologies, except for the HQGA-Q, where the range is almost zero. We can see that with a few objects, the HQGA-Q has a robust performance.

Figure 4 shows the best solutions for the knapsack problem with 250 objects in a box plot. Here, we can see a similar behavior between the algorithms that combined bits and qubits (HQGA and QIEA). Also, the algorithms that only use qubits (HQGA-Q and QIEA-Q) have almost the same behavior between each other, since the distribution of their data and the median is very similar for each pair of algorithms.

Another point worth highlighting is that the HQGA-Q presents better distribution in its solutions, unlike those shown in Figure 3, while QIEA-Q shows a lower distribution in its solutions. This demonstrates how the number of objects affects both algorithms' performance as the number of objects increases, while the HQGA and QIEA do not show this behavior.

The behavior of the HQGA and the QIEA shown in Figure 5 is very similar to that presented in Figure 4, even to that shown in Figure 3. This demonstrates the stability of both algorithms when

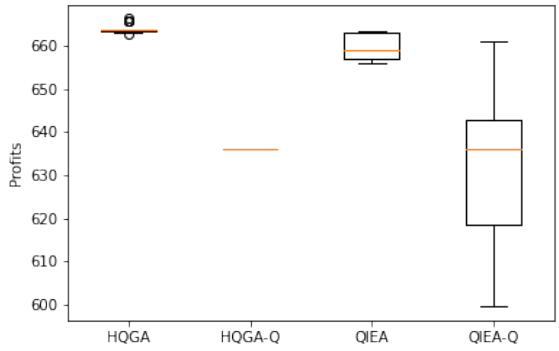


Fig. 3. Box plot of the best profits (solutions) of the HQGA, HQGA-Q, QIEA, QIEA-Q with 100 objects

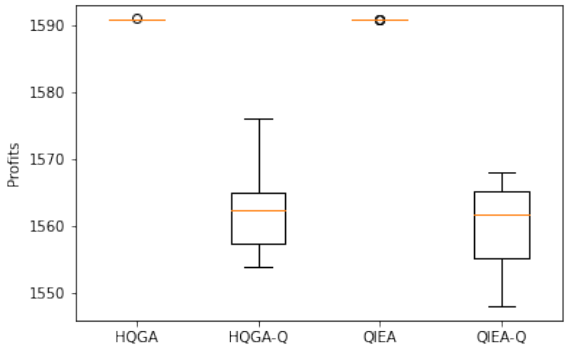


Fig. 4. Box plot of the best profits (solutions) of the HQGA, HQGA-Q, QIEA, QIEA-Q with 250 objects

solving the knapsack problem with a different number of objects.

On the other hand, the HQGA-Q and the QIEA-Q present behaviors very different from those shown in Figures 3 and 4, with a negative asymmetric distribution where most of their data tend towards the third quartile demonstrating the instability of the algorithms that only use qubits (HQGA-Q and QIEA-Q) in solving the knapsack problem.

From the previous results, we can see that the HQGA and QIEA proposals are the ones that have shown the best results when solving the knapsack problem with 100, 250, and 500 objects. For this reason, Figure 6, 7, and 8 present the performance

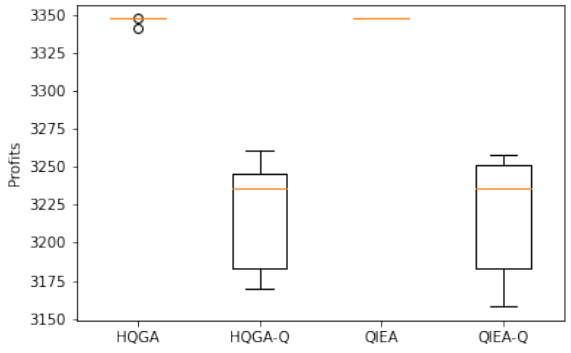


Fig. 5. Box plot of the best profits (solutions) of the HQGA, HQGA-Q, QIEA, QIEA-Q with 500 objects

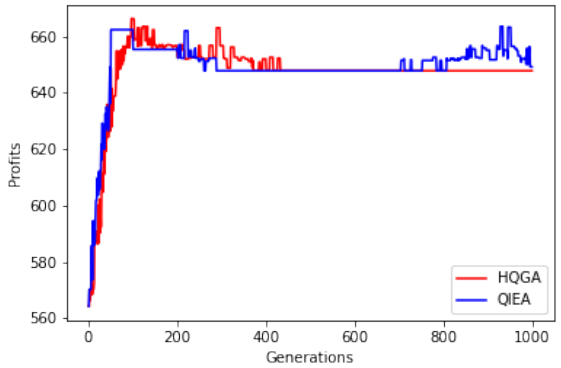


Fig. 6. Performance of the best run of HQGA vs QIEA with 100 objects

of both algorithms in solving the knapsack problem with 100, 250, and 500 objects, respectively.

From Figure 6 a comparison between the performance of HQGA and QIEA solving the knapsack problem with 100 objects is obtained. The solid red line represents the best run of the HQGA, and the solid blue line represents the performance of the best run of the QIEA. The runs number 24 and 6 were the best for the HQGA and QIEA, respectively. For HQGA, a stable solution is reached after generation number 400, while for the QIEA, a stable solution is never reached. The best solution of the HQGA was obtained in generation number 97. For the QIEA, the best solution occurred in generation number 92. It can be seen that the HQGA finds a better solution to the problem in fewer generations than the QIEA, in addition to presenting constant solutions after a certain number of generations, which shows the stability of the HQGA.

With 250 objects, the HQGA in Figure 7 shows its best performance in run 28. The best solutions were obtained in generation 104, and the algorithm reached stable solutions after generation 400 (specifically generation 463). The QIEA had its best performance in the first run, reaching the best solution in generation 101 and the stability after generation 300. For this problem, it can be seen that both algorithms reached stable solutions after a certain number of generations, unlike what is shown in Figure 6, where the QIEA cannot achieve this behavior, we can even see how the migration operator in the QIEA allows obtaining constant solutions through certain generations. However, this does not mean that the QIEA is more stable than the HQGA since with 100 objects (Figure 6) it could be seen that the HQGA was the one that found stable solutions, while QIEA did not present the same behavior.

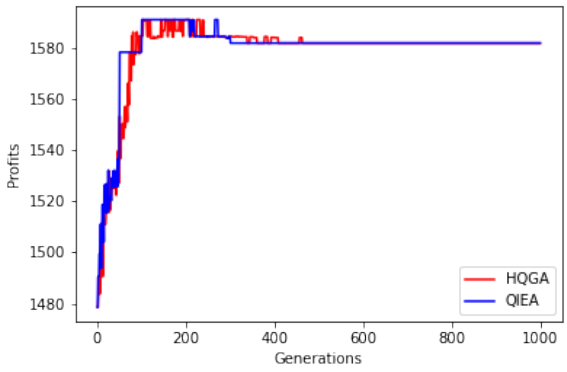


Fig. 7. Performance of the best run of HQGA vs QIEA with 250 objects

The performance of the HQGA and QIEA solving the knapsack problem with 500 objects (see Figure 8) is very similar to that presented in Figure 7. In both cases, the first run was the best. The HQGA obtained its best solution at generation 247

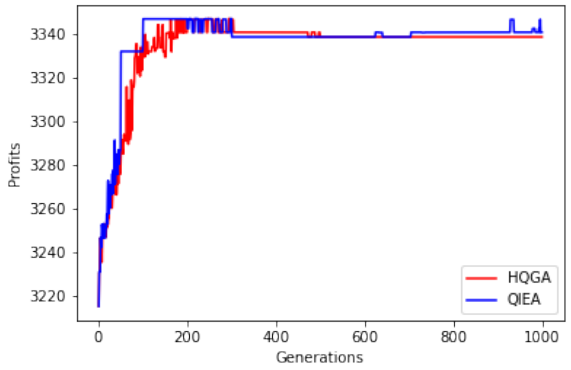


Fig. 8. Performance of the best run of HQGA vs QIEA with 500 objects

and stable solutions after generation 300. QIEA reached its best solution at generation 205 and stable solutions after generation 300, but we can see a small peak at generation 600 and after generation 900. While the HQGA only shows constant solutions.

With 250 and 500 objects (Figure 7 and 5) the QIEA obtained a best solution before the HQGA showing a better convergence time. Nevertheless, the HQGA in all the experiments (100, 250 and 500 objects) has shown a better stability and better optimal solutions.

So far only experiments comparing hybrid quantum algorithms have been shown. However, it is necessary to compare the performance of the HQGA with a classical genetic algorithm (GA) to get a complete picture of the HQGA's behavior. GA was run 30 times, with 1000 generations on three strongly correlated data sets (100, 250, and 500 objects).

Table 9. HQGA and GA experimental results with 100 objects. Best results in bold

Algorithm	Best	Worst	Average	Std
HQGA	666.23	**662.6**	663.89	**0.945**
GA	**712.18**	658.03	**687.35**	12.39

For the selection process, the tournament selection operator was used, in the recombination process the single one-point crossover operator was used, and finally a mutation operator. In all the experiments a population of 100 chromosomes, a recombination rate of 0.8, and a mutation rate of 0.4 were used.

Table 10. HQGA and GA experimental results with 250 objects. Best results in bold

Algorithm	Best	Worst	Average	Std
HQGA	1591.03	1591.01	1591.01	**0.003**
GA	**1688.42**	**1608**	**1651.16**	16.61

Table 11. HQGA and GA experimental results with 500 objects. Best results in bold

Algorithm	Best	Worst	Average	Std
HQGA	3347.33	3340.91	3347.12	**1.173**
GA	**3581**	**3513.86**	**3548.75**	19.6

Tables 9, 10, and 11 show the experimental results of the HQGA and GA with 100, 250, and 500 objects, respectively. In all the experiments, HQGA is outperformed by GA in obtaining the best solutions. Let us remember that both algorithms were executed in a classical computer, this means that the GA has the advantage of being able to generate its complete population of bits, while the HQGA being a hybrid algorithm and due to the limitations of the NISQ era mentioned above, it cannot generate the entire population of qubits, which limits the HQGA's ability to perform better. To correctly approach the comparison of both algorithms, it would be necessary to implement the HQGA in a quantum computer that allows the use of the necessary number of qubits.

Although, the HQGA was clearly surpassed by the GA in the best solutions, the HQGA obtained in all the experiments the smallest standard deviation (Std), demonstrating a better robustness even when implemented in a classical computer.

To see more clearly the performance of the HQGA and GA, in Figures 9, 10, and 11 the performance graphs of both algorithms are presented with 100, 250, and 500 objects, respectively. With 100 objects, Figure 9 clearly shows from the beginning of the generations the superiority of GA compared to HQGA.

However, the search capacity of the HQGA is better, as its first solution was approximately 370 and its best solution was 666.23, therefore there was a 1.8-fold increase over the initial solution. While GA's first solution was approximately 620, and its best solution was 712.18, the increase was 1.14 times, less than presented by the HQGA.

Figure 10 shows the performance of the HQGA and GA with 250 objects. In addition to the evident superiority of the GA over the HQGA, it can be noted that both proposals reached their best solution in the same number of generations (approximately in the 100th generation).

With 500 objects (Figure 11), things remain very similar to what was previously presented.

One of the main disadvantages of the HQGA is the execution time. Because the number of qubits (and bits) in a quantum register equals the number of objects in the knapsack, it is currently impossible to deal with problems with more than 15 elements [29].

For instance, a system with n-qubits can represent 2^n states simultaneously, which becomes a difficult task for a current classic computer. The tests were performed on personal computer equipment with 8 Gigabytes of RAM memory and a Intel(R) Core(TM) i7-3537U CPU @ 2.50 GHz processor.

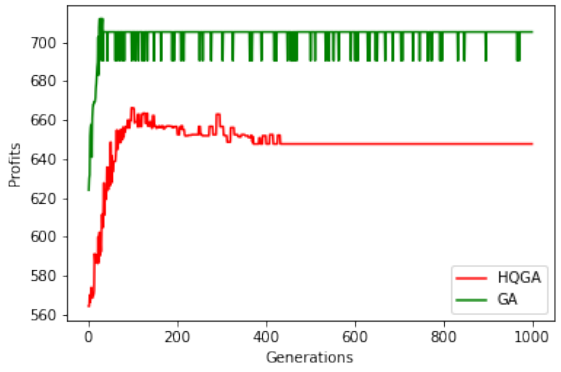


Fig. 9. Performance of the best run of HQGA vs GA with 100 objects

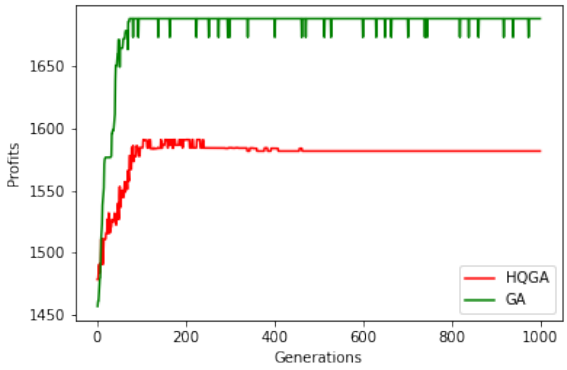


Fig. 10. Performance of the best run of HQGA vs GA with 250 objects

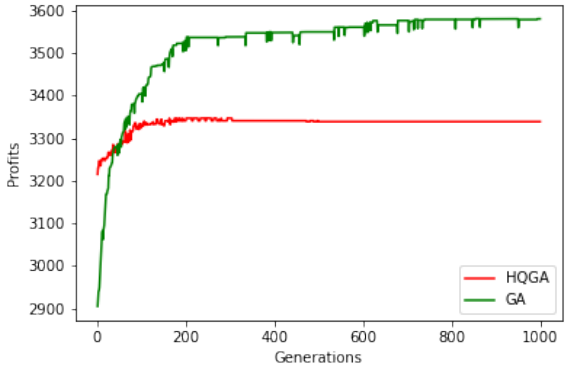


Fig. 11. Performance of the best run of HQGA vs GA with 500 objects

5 Conclusion and Future Work

This paper proposes a Hybrid Quantum Genetic Algorithm (HQGA), a modified version of the HQGA (HQGA-Q), a Quantum Inspired Evolutionary Algorithm (QIEA), and a modified version of QIEA (QIEA-Q) implemented in a quantum simulator to solve the 0-1 knapsack problem.

The novelty of this work is the design of the aforementioned quantum algorithms and their implementation in the Qiskit IBM quantum simulator. There are already some works in the state of the art that have implemented algorithms in the Qiskit IBM simulator [29, 44], solving the binary

knapsack problem and the traveling salesman problem, respectively. However, this proposal implemented and compared the performance of four quantum algorithms solving the knapsack problem with different number of objects using a strongly correlated data set. In addition, a quantum circuit was designed and implemented to generate the initial quantum population. This proposal until today has not been seen in any other work present in the state-of-the-art.

Also, the idea of incorporating bits and qubits to the designed algorithms was proposed and implemented, with the aim of solving the limitation of the quantum simulator.

The results showed that HQGA presents better solutions solving the knapsack problem with 100, 250, and 500 objects than the other hybrid quantum algorithms, except with GA, in this case HQGA was exceeded in all experiments. With hybrid quantum algorithm, HQGA presented a lower standard deviation in 1/3 (33%) of the cases. Furthermore, the HQGA in the performance test demonstrated a better stability in all the experiments than the QIEA. Although, in 2/3 of the experiments the QIEA reached the best solution before the HQGA showing a better convergence time. In none of the experiments, the QIEA-Q and HQGA-Q had better solutions than HQGA and QIEA.

The main disadvantage of the HQGA is the execution time and the ability to find no better solutions than Genetic algorithm (GA), as explained in section 4, the number of qubits (and bits) using by a quantum register in a current classic computer is limited, since for a n-qubit register 2^n states can be represented at the same time, to calculated that amount of data in a classic computer would be a difficult task. It must be remembered that the experiments were carried out on a classical computer using a simulated quantum environment (ibmq_qasm_simulator), and not on a real quantum computer where the natural parallelism of these computers would demonstrate their superiority.

For future work the HQGA, HQGA-Q, QIEA, and QIEA-Q will be implemented in a real IBM quantum computer and the results will be compared against the results presented in this work. Other methods will be tried to update the rotation angle for the quantum gates. The four quantum algorithms will be adapted to solve other kinds of combinatorial optimization problems, such as, the traveling salesman problem or path planning problems. Finally, the quantum-inspired algorithms proposed in this work will be modified to solve multi-objective combinatorial optimization problems.

Acknowledgments

We thank Instituto Politécnico Nacional (IPN), the Comisión de Fomento y Apoyo Académico del IPN (COFAA), and the Mexican National Council of Science and Technology (CONACYT) for supporting our research activities.

References

1. **Adeyemo, H., Ahmed, M. (2017).** Solving 0/1 knapsack problem using metaheuristic techniques. 9th IEEE-GCC Conference and Exhibition (GCCCE), pp. 1–6.
2. **Ajagekar, A., You, F. (2019).** Quantum computing for energy systems optimization: Challenges and opportunities. Energy, Vol. 179, pp. 76–89.
3. **Arute, F., Arya, K., Babbush, R., others (2019).** Quantum supremacy using a programmable superconducting processor. Nature, Vol. 574, pp. 505–510.
4. **Ballance, C., Harty, T., Linke, N., Sepiol, M., Lucas, D. (2016).** High-fidelity quantum logic gates using trapped-ion hyperfine qubits. Physical Review Letters, Vol. 117, No. 6.
5. **Barends, R., Kelly, J., Megrant, A., Veitia, A., Sank, D., Jeffrey, E., White, T. C., Mutus, J., Fowler, A. G., Campbell, B., et al. (2014).** Superconducting quantum circuits at the surface code threshold for fault tolerance. Nature, Vol. 508, No. 7497, pp. 500–503.
6. **Bhattacharjee, K. K., Sarmah, S. P. (2015).** A binary cuckoo search algorithm for knapsack problems. 2015 International Conference on Industrial Engineering and Operations Management (IEOM), pp. 1–5.

7. **Bhattacharjee, K. K., Sarmah, S. P. (2015).** A binary firefly algorithm for knapsack problems. 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pp. 73–77.
8. **Braine, L., Egger, D. J., Glick, J., Woerner, S. (2021).** Quantum algorithms for mixed binary optimization applied to transaction settlement. IEEE Transactions on Quantum Engineering, Vol. 2, pp. 1–8.
9. **Cao, Z., Uhlmann, J., Liu, L. (2018).** Analysis of deutsch-jozsa quantum algorithm. IACR Cryptology ePrint Archive, Vol. 2018, pp. 249.
10. **Cook, S. A. (1971).** The complexity of theorem-proving procedures. IN STOC, ACM, pp. 151–158.
11. **Du, K.-L., Swamy, M. (2016).** Search and Optimization by Metaheuristics.
12. **Forno, E., Acquaviva, A., Kobayashi, Y., Macii, E., Urgese, G. (2018).** A parallel hardware architecture for quantum annealing algorithm acceleration. 2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), pp. 31–36.
13. **Gao, Y., Zhang, F., Zhao, Y., Li, C. (2018).** Quantum-inspired wolf pack algorithm to solve the 0-1 knapsack problem. Mathematical Problems in Engineering, Vol. 2018, pp. 1–10.
14. **García, J., Crawford, B., Soto, R., Castro, C., Paredes, F. (2018).** A k-means binarization framework applied to multidimensional knapsack problem. Applied Intelligence, Vol. 48, No. 2, pp. 357–380.
15. **Han, K.-H., Kim, J.-H. (2000).** Genetic quantum algorithm and its application to combinatorial optimization problem. Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), volume 2, pp. 1354–1360.
16. **Han, K.-H., Kim, J.-H. (2003).** Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. Evolutionary Computation, IEEE Transactions on, Vol. 6, pp. 580–593.
17. **Han, K.-H., Park, K.-H., Lee, C.-H., Kim, J.-H. (2001).** Parallel quantum-inspired genetic algorithm for combinatorial optimization problem. Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), volume 2, pp. 1422–1429.
18. **Huang, Y., Wang, P., Li, J., Chen, X., Li, T. (2019).** A binary multi-scale quantum harmonic oscillator algorithm for 0–1 knapsack problem with genetic operator. IEEE Access, Vol. 7, pp. 137251–137265.
19. **IBM (2020).** The qiskit elements. https://quantum-computing.ibm.com/docs/qiskit/the_elements.
20. **Jindal, A., Bansal, S. (2019).** Effective methods for constraint handling in quantum inspired evolutionary algorithm for multi-dimensional 0–1 knapsack problem. 2019 4th International Conference on Information Systems and Computer Networks (ISCON), pp. 534–537.
21. **Johansson, N., Larsson, J.-A. (2017).** Efficient classical simulation of the Deutsch–Jozsa and Simon's algorithms. Quantum Information Processing, Vol. 16, No. 9.
22. **Khemakhem, M., Chebil, K. (2016).** A tree search based combination heuristic for the knapsack problem with setup. Computers & Industrial Engineering, Vol. 99, pp. 280–286.
23. **King, J., Yarkoni, S., Raymond, J., Ozfidan, I., King, A. D., Nevisi, M. M., Hilton, J. P., McGeoch, C. C. (2017).** Quantum annealing amid local ruggedness and global frustration.
24. **Lai, X., Hao, J., Yue, D., Gao, H. (2018).** A diversification-based quantum particle swarm optimization algorithm for the multidimensional knapsack problem. 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), pp. 315–319.
25. **Lai, X., Hao, J.-K., Fu, Z.-H., Yue, D. (2020).** Diversity-preserving quantum particle swarm optimization for the multidimensional knapsack problem. Expert Systems with Applications, Vol. 149, pp. 113310.
26. **Lai, X., Hao, J.-K., Yue, D. (2019).** Two-stage solution-based tabu search for the multidemand multidimensional knapsack problem. European Journal of Operational Research, Vol. 274, No. 1, pp. 35–48.
27. **Li, H. (2019).** An angle-expressed quantum evolutionary algorithm for quadratic knapsack problem. IOP Conference Series: Materials Science and Engineering, Vol. 631, pp. 052054.
28. **Liu, C.-L., Wan, M.-H., Yang, J.-Y. (2010).** An improved quantum genetic algorithm and its application. 2010 International Conference on Computer Application and System Modeling, pp. 413–418.
29. **López-Sandoval, D., Cobos, C. (2020).** Adiabatic quantum computing applied to the solution of the binary knapsack problem. RISTI - Revista Iberica

de Sistemas e Tecnologias de Informacao, Vol. 38, pp. 214–227.

30. **Montiel Ross, O., Rubio, Y., Olvera, C., Rivera, A. (2019).** Quantum-inspired acromyrmex evolutionary algorithm. Scientific Reports, Vol. 9.
31. **Montiel Ross, O. H. (2020).** A review of quantum-inspired metaheuristics: Going from classical computers to real quantum computers. IEEE Access, Vol. 8, pp. 814–838.
32. **Narayanan, A. (1999).** Quantum computing for beginners. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Vol. 3, pp. 2231–2238.
33. **Narayanan, A., Moore, M. (1996).** Quantum-inspired genetic algorithms.
34. **Nezamabadi-pour, H. (2015).** A quantum-inspired gravitational search algorithm for binary encoded optimization problems. Engineering Applications of Artificial Intelligence, Vol. 40, pp. 62–75.
35. **Ozsoydan, F. B., Baykasoglu, A. (2019).** A swarm intelligence-based algorithm for the set-union knapsack problem. Future Generation Computer Systems, Vol. 93, pp. 560–569.
36. **Paredes López, M., Meneses Viveros, A., Morales-Luna, G. (2018).** Algoritmo cuántico de Deutsch y Jozsa en GAMA. Revista mexicana de física, Vol. 64, pp. 181–189.
37. **Pednault, E., Gunnels, J. A., Nannicini, G., Horesh, L., Wisnieff, R. (2019).** Leveraging secondary storage to simulate deep 54-qubit sycamore circuits.
38. **Pisinger, D. (2005).** Where are the hard knapsack problems? Computers & Operations Research, Vol. 32, No. 9, pp. 2271–2284.
39. **Preskill, J. (2012).** Quantum computing and the entanglement frontier.
40. **Preskill, J. (2018).** Quantum computing in the NISQ era and beyond. Quantum, Vol. 2, pp. 79.
41. **Sapra, D., Sharma, R., Agarwal, A. P. (2017).** Comparative study of metaheuristic algorithms using knapsack problem. 7th International Conference on Cloud Computing, Data Science Engineering - Confluence, pp. 134–137.
42. **Shi, H. (2006).** Solution to 0/1 knapsack problem based on improved ant colony algorithm. IEEE International Conference on Information Acquisition, pp. 1062–1066.
43. **Si, L., Shi, L., Wang, Y. (2010).** A novel self-organizing quantum evolutionary algorithm for multi-objective optimization. 2010 International Conference on Educational and Network Technology, pp. 499–503.
44. **Srinivasan, K., Satyajit, S., Behera, B. K., Panigrahi, P. K. (2018).** Efficient quantum algorithm for solving travelling salesman problem: An IBM quantum experience.
45. **Tkachuk, V. (2018).** An adaptive quantum evolution algorithm for 0–1 knapsack problem. System research and information technologies, pp. 77–88.
46. **Tkachuk, V., Tkachuk, O. (2018).** Higher-order quantum genetic algorithm for 0-1 knapsack problem. System research and information technologies, pp. 52–67.
47. **Williams, C. (2011).** Explorations in Quantum Computing.
48. **Yang, S., Jiang, Y., Nguyen, T. T. (2012).** Metaheuristics for dynamic combinatorial optimization problems. IMA Journal of Management Mathematics, Vol. 24, No. 4, pp. 451–480.
49. **Yanofsky, N., Manucci, M. (2008).** Quantum computing for computer scientists.
50. **Zhang, G. (2011).** Quantum-inspired evolutionary algorithms: A survey and empirical study. J. Heuristics, Vol. 17, pp. 303–351.
51. **Zhang, R., Gao, H. (2007).** Improved quantum evolutionary algorithm for combinatorial optimization problem. 2007 International Conference on Machine Learning and Cybernetics, volume 6, pp. 3501–3505.
52. **Zhou, Y., Chen, X., Zhou, G. (2016).** An improved monkey algorithm for a 0-1 knapsack problem. Applied Soft Computing, Vol. 38, pp. 817–830.

Article received on 20/06/2021; accepted on 17/11/2021.
Corresponding author is Enrique Ballinas.

ISSN 2007-9737

Fuzzy Combination of Moth-Flame Optimization and Lightning Search Algorithm with Fuzzy Dynamic Parameter Adjustment

Yunkio Kawano, Fevrier Valdez, Oscar Castillo

Tijuana Institute of Technology, Computer Science Department, Mexico

monico.kawano@tectijuana.edu.mx, fevrier@tectijuana.mx

Abstract. In general, this paper is focused on creating a fuzzy combination of two optimization algorithms. In this case, the algorithms work with populations and allow us to migrate between them every certain number of iterations. On the other hand, fuzzy logic is responsible for the dynamic adjustment of parameters within each of the algorithms since the variables are different in each algorithm. In previous works, a combination between genetic algorithm and particle swarm optimization was developed, which motivated us to create this combination expecting to obtain better results when compared to the previous works. The moth-flame optimization and lightning search algorithm were combined to obtain a powerful hybrid metaheuristic combining the advantages of both individual algorithms.

Keywords. Swarm intelligence algorithms, fuzzy logic systems, migration.

1 Introduction

In this paper, a combination of two parallel optimization algorithms is made that seeks between the two a better solution to the problem with which we are working. These algorithms are the Moth-Flame Optimization and Lightning Search Algorithm. There is probably no exact relationship between these two algorithms, but as regards the results obtained individually, we can see that the moth-flame optimization is a good algorithm for exploring the search space at the moment that it is trying to fly in a straight line to his destination.

On the other hand, the lightning search algorithm focused on the creation of lightning depending on the richness in terms of the concentration of certain chemical elements that are scattered in the air of the clouds, which the shock that these elements have creates energy of lightnings. Which each small beam of light where the lightning begins and from there different strands come out to form a ray shaped like an inverted tree, which is a set of possible solutions.

The inspiration that led us to make this fuzzy combination in parallel is the previous paper in which we were working focused on the search for a good combination of two optimization algorithms at that time, where we used the optimization algorithms of GA and PSO were we shared between both algorithms a certain portion of the populations.

Now in this paper we will try to improve the results obtained in comparison with the combination of PSO and GA [1].

At the time of creating the paper, we have not found any precedent that the LSA and MFO algorithms are being used to evaluate benchmark functions as well as the use of dynamic parameter adjustment or the combination between both algorithms. We have only found that they are being used for signal optimization emitted by antennas or similar things. In the next section we present some other population-based optimization algorithms and the theory of each of the algorithms that we are going to use.

Which is the Moth-Flame optimization algorithm and the Lightning search algorithm and a bit of how our fuzzy combination is developed. In the experiments section we show the parameters that we will use as well as the fuzzy rules and the benchmark functions that we will evaluate.

There we will show some of the results obtained by the experiments. Finally, we will outline a conclusion about everything we observed within the experiments.

2 Background

There are different categories of optimization algorithms in which they may have been inspired by how an individual within the population uses certain methods to find food. In other cases, it may be that the algorithm is inspired by the movement that the individual has to reach its destination. Algorithms have also been seen that are inspired by how adolescents develop in their lives to reach the next age stage.

The search algorithms can be that of the dragonfly algorithm that is inspired by how it searches food, and we also can find another algorithm, like a grasshopper optimization algorithm.

In the case of movement, the bird swarm optimization algorithm can be seen.

There are different optimization algorithms that work with populations such as the PSO, DA, BSA, AISA, and FA algorithms, among others. But in this case, we will use the MFO and LSA algorithm that although they have nothing to do with each other, the two manage populations, only one is moths in search of food and the other seeks to create lightning. In this case, the two algorithms are dedicated to looking for something in common, which would be a good solution for any problem they are working.

Next, we will show you a brief description of each of the aforementioned algorithms.

In the case of the particle swarm optimization (PSO) is a population-based stochastic optimization technique, are inspired by social behavior of bird flocking or fish schooling in search of food. PSO has many processes similar to those that work with genetic algorithms. This algorithm initiates a swarm of random particles, where each of the contained particles could be a solution to the problem that is being worked on. These possible solutions are evaluated in each iteration that we have [2, 3, 4].

On the social spider optimization (SSO), this algorithm is based on the simulation of the cooperative behavior of social-spiders. The individuals emulate a group of spiders which interact between others spiders with the biological laws in the colony. The algorithm has two search agents are the spider males or females, each gender has different tasks [5, 6, 7].

The Dragonfly algorithm (DA) is inspired by the behavior that can occur in a static and dynamic way for dragonflies, which dynamically while in search of food can communicate with other dragonflies to find food, which for an optimization algorithm would be the part of exploration and when it is in static way it can exploit the area [8].

The Bird swarm algorithm (BSA) is inspired by social behavior and the iterations it does depend on the type of bird it is. There are three types of birds or with different tasks within the swarm, it can be foraging behavior, vigilance behavior and flight behavior [9].

The Adolescent Identity Search Algorithm (AISA) is inspired by the simulation about how the identity a teenager in a couple is formed, where all the experiences that it can lead to live improve their knowledge or behavior [10].

Finally, the firefly algorithm (FA) is a population-based optimization algorithm that mimics a firefly's attraction to flashing light. In particular it used the concept of how the brightness of individual fireflies drew them together and a randomness factor to encourage exploration of the solution space [11, 12, 13, 14].

2.1 Optimization Algorithms Based on Swarms

At the moment there are already algorithms for almost any activity which may be inspired by biology or phenomena that occur in nature, such as talking about algorithms inspired by biology, we can find the PSO algorithm that aims to schools of fish or flocks of birds move to find food, or in another case the algorithm that uses moths that focuses on how the moth makes reference to the moon to be able to fly long distances in a straight line. Or the other case, in algorithms inspired by nature, as in our case would be the lightning search algorithm, which is dedicated to trying to simulate what is known as the lightning that could be described as an inverted tree, where each thread or tip that has the lightning would be a possible solution. That is why we could say that we combine two worlds, where it is a biologically inspired algorithm and also a naturally inspired algorithm to find between the two algorithms a better solution for the problem we are dealing with.

Although there are too many algorithms of different types, we selected these algorithms

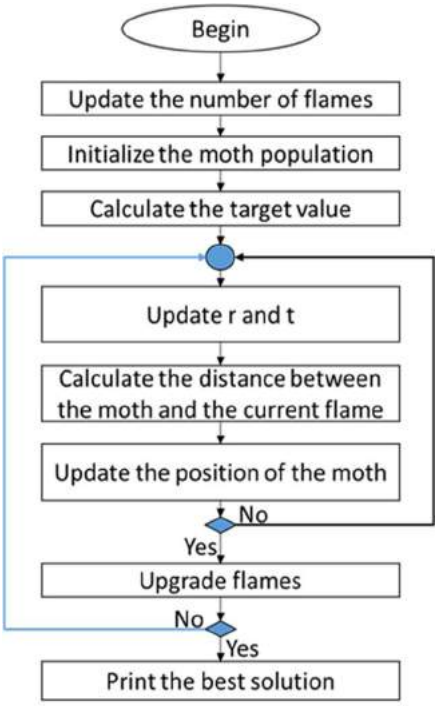


Fig. 1. Flowchart Moth-Flame Optimization

because they were interesting to us to test the combination of both, where we think that speaking of nature, moths and lightning do not get along. Apart from the fact that individually the two algorithms provide good solutions [15].

The following optimization algorithms are used to create the fuzzy combination shown in section 3 of this paper.

2.1.1 Moth Flame Optimization

This optimization algorithm is inspired by the orientation ability of the moths when flying in the middle of the night towards the moon. The moths are oriented by means of a mechanism called transverse orientation, which consists of maintaining a fixed angle in the direction of the moon to fly in a straight line over great distances, since the moon is very far away.

Although currently in nature the moths are confused in the presence of artificial lightning, which causes them to think that they see the moon, but since the distances are much shorter the moths begin to spin around the lamp without control, until unfortunately die.

The algorithm assumes that moths and flames are among the possible solutions, where moths are search agents that move around the space and flames are the best position found so far by the moth. These moths fly close to the flame in case they find a better solution. The flame is updated.

The movement of the moth while in search of a better solution is in logarithmic spiral. Where at the beginning of the spiral is a moth and at the end it must be the position of the flame, and it should be noted that the range of the spiral must not exceed the search space:

$$S(M_i, F_j) = D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_j \,, \tag{1}$$

where $D_i = \left|F_j - M_i\right|$ is the distance between the flame and moth where M_i is the position of the moth in i and F_j is the position of the flame in j. b it's a constant that defines the logarithmic spiral. t is a random number between $[-1,\ 1]$. In MFO, the control between exploitation and exploration is thanks to S that is the spiral movement of the moth near the flame in the search space.

At a certain point of the algorithm, the update of the number of flames is applied since it helps us to improve the exploitation of the MFO algorithm. Because the algorithm searches in various positions within the search space, which reduces the number of possibilities we have to exploit the best possible solutions.

Therefore, reducing the number of flames helps to solve this problem based on the following equation:

$$flames\ no. = \left(N - l * \frac{N-l}{T}\right), \tag{2}$$

where N is the maximum number of flames, *l* is the current number of iterations, and *T* indicates the maximum number of iterations[16, 17, 18, 19].

2.1.2 Lightning Search Algorithm

This is an optimization algorithm that is inspired by the natural phenomenon of how lightning is created in the natural environment.

A propagation mechanism is used in a staggered manner, which takes the form of an

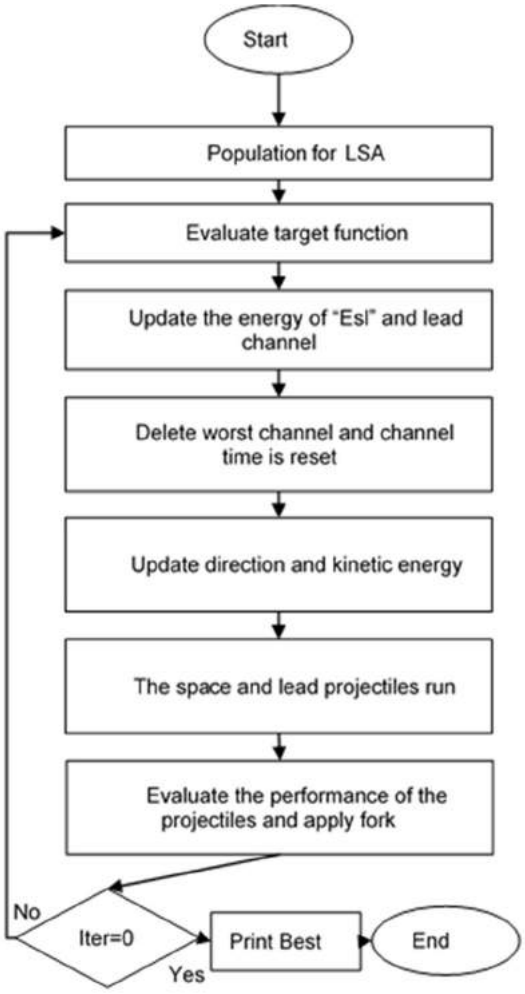


Fig. 2. Flowchart Lightning Search Algorithm

inverted tree, which is why the algorithm has three types of projectiles, where the first is a transition one which generates the first leading population, then continue the space projectiles that try to obtain a better position in the leadership range and finally we have the leading projectiles that have the best position.

The projectiles are composed of hydrogen, nitrogen and oxygen atoms that can be found near the region of the storm clouds, when the molecules of these elements travel at a great speed through the atmosphere and are ionized the produce a path or channel through the collision and transition to the step leader.

The projectiles that travel under normal conditions through the atmosphere lose kinetic energy when they collide with molecules in the air. The velocity of a projectile is obtained with the following equation:

$$v_p = \left[1 - \left(\frac{1}{\sqrt{1-\left(\frac{v_0}{c}\right)^2}} - \frac{sF_i}{mc^2}\right)^{-2}\right]^{-\frac{1}{2}}, \quad (3)$$

where v_p and v_0 are the current velocity and initial velocity, respectively, of the projectile; c is the speed of light; F_i is the constant ionization rate; m is the mass of the projectile; and s is the length of the path traveled.

The equation shows that velocity is a function of leader tip position and projectile mass. When the mass is small or when the path traveled is long, the projectile has little potential to ionize or explore a large space. Other property of a stepped leader is forking, which means that are two symmetrical branches are created because the nuclei collision of the projectile is realized by using the opposite number as in the next equation:

$$\overline{p_i} = a + b - p_i \quad (4)$$

where $\overline{p_i}$ and p_i are the opposite and original projectiles, respectively, in a one-dimensional system; *a* and *b* are the boundary limits. This adaptation may improve some of the bad solutions in the population. If forking does not improve channel propagation in the LSA, one of the channels at the forking point is illuminated to maintain the population size [20, 21, 22, 23].

2.1.3 Fuzzy Combination of MFO and LSA

The fuzzy combination of the Moth-Flame Optimization (MFO) and Lightning Search Algorithm (LSA) is to try to find a better solution, since the MFO algorithm is good to explore within the search space and the LSA algorithm is good to exploit although the algorithm converges prematurely, so that is why we combine them to try to balance the amount to exploration and exploitation necessary so that between the two algorithms better solutions are obtained when comparing the algorithms individually [21, 22, 23].

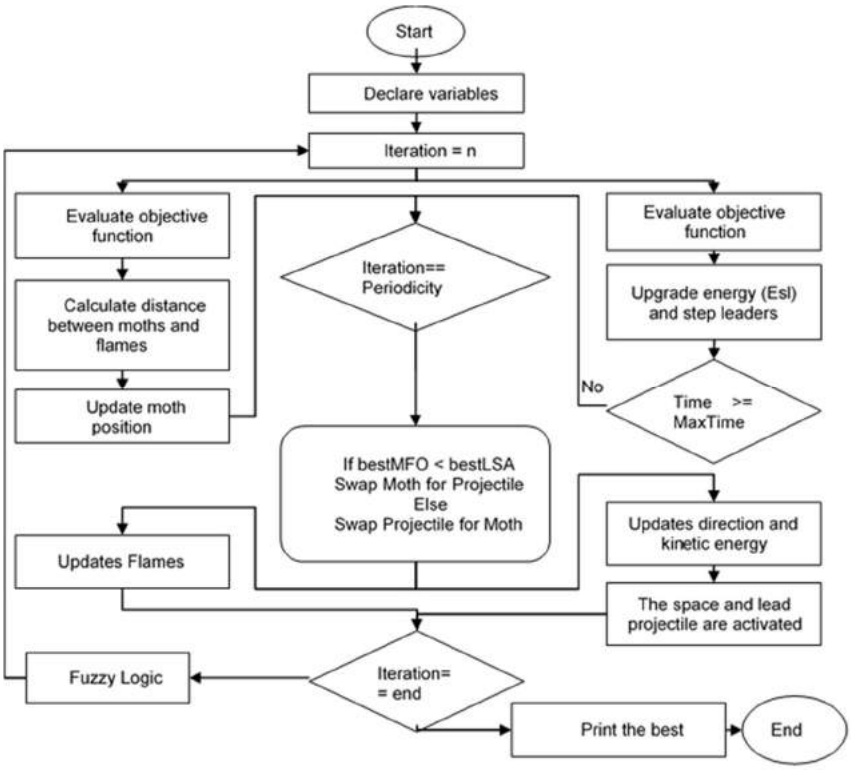


Fig. 3. Flowchart Lightning Search Algorithm

To achieve the combination of the MFO and LSA algorithms, we configure them so that the two algorithms are intertwined to run at the same time and to apply the use of migration blocks, which allow us to exchange a certain number of individuals within of each of the populations, it would be like exchanging a certain number of moths for lightning bolts within the algorithms used [27].

In Figure 3 can be appreciated how the operation of the algorithms combined into one is shown in the diagram. According to the diagram it is seen that the algorithms are executed simultaneously with each of their processes on each side of the drawing and in the central part of the diagram is what would be the migration block, it is used every certain number of individuals to be shared.

The diagram shows where the dynamic parameter adjustment is applied, but does not describe what is the condition to apply it. The condition that it is applied is from iteration 100 onwards, so that the algorithm has time to look for a shortly before starting to share the individuals and this action stops applying 100 iterations before reaching the maximum number of iterations, regardless of the maximum number of iterations that are being used. The variables that are adjusted in the two algorithms that we use, one variable is about the value of the spiral used by the moth-flame algorithm and the other is from the lightning search algorithm, which would be the probability of creating two solutions form one or dividing the lightning channel.

The results obtained are the average of 30 runs of the code, thus being able to perform a valid statistical test.

3 Experiments

The experiments were carried out on a computer with an Intel i5-9400f processor that has 6 cores and 6 threads, it is complemented by 16 gigabytes of RAM memory and a Nvidia GTX 970 video card

Table 1. Parameters for algorithms used

Parameter	Value
Populations	100
Dimensions	5, 10, 20, 40 and 80.
Iterations	500, 1000 and 2000.
Logarithmic spiral amplitude (MFO)	[0.001, 0.99]
Maximum channel time (LSA)	10
Bifurcation percentage (LSA)	[0.001, 0.99]

Table 2. First fuzzy logic system for adapt value of Spiral in Moth-Flame algorithm

Rules	Iteration	Spiral
1	Low	Low
2	Mid-Low	Mid-Low
3	Mid	Mid
4	Mid-High	Mid-High
5	High	High

Table 3. Second fuzzy logic system for adapt value of Spiral in Moth-Flame algorithm

Rules	Iteration	Spiral
1	Low	High
2	Mid-Low	Mid-High
3	Mid	Mid
4	Mid-High	Mid-Low
5	High	Low

Table 4. First fuzzy logic system to adapt the value of Fork in Lightning Search algorithm

Rules	Iteration	Spiral
1	Low	Low
2	Mid-Low	Mid-Low
3	Mid	Mid
4	Mid-High	Mid-High
5	High	High

with 4 gigabytes GDDR5. Although the latter is unnecessary since the code is not focused on having better times with the use of the GPU, in terms of storage it is a high-speed solid hard disk in which we can find MATLAB R2017b installed to run the codes.

Table 5. Second fuzzy logic system to adapt the value of Fork in Lightning Search algorithm

Rules	Iteration	Spiral
1	Low	High
2	Mid-Low	Mid-High
3	Mid	Mid
4	Mid-High	Mid-Low
5	High	Low

3.1 Fuzzy Logic Systems

Fuzzy logic allows a better analysis of the input data that a complex computational system is going to have since it gives us the ability to have several possible best solutions, which is decided by one in terms of the rules that have been specified in the system fuzzy as well as membership functions.

The following table shows the simple configuration that we use where there are only 5 rules for each of the variables with dynamic parameter adjustment, where basically all of them have one input and one output [28].

Currently there are some works that make use of fuzzy logic systems to adjust the parameters of all kinds of algorithms [23, 24, 25, 26].

Table 2 is for dynamic adjustment of the spiral variable that is used by the MFO algorithm, where the output values go increasing.

Table 3 is for dynamic adjustment of the spiral variable that is used by the MFO algorithm, where the output values go decreasing.

Table 4 is for dynamic adjustment of the fork variable that is used by the LSA algorithm, where the output values go increasing.

Table 5 is for dynamic adjustment of the fork variable that is used by the LSA algorithm, where the output values go decreasing.

4 Experiments

Table 6 shows the benchmark functions used to evaluate the performance of each of the algorithms, be they the MFO, LSA algorithms and in the fuzzy combination of both.

The following tables show some of the most relevant results obtained in the experimentation stage of each of the algorithms, already in the original version of MFO and LSA as well as in the fuzzy combined version of both, where they help each other. two for best result.

After each of the tables, an explanation of what is observed in each one is presented.

4.1 Comparison of All Algorithms Used

This section shows all the results obtained from the experiments carried out, it should be noted that each result shown is the average of 30 runs.

In Table 7 the results for 5 dimensions and 500 iterations are compared, we can see that the first four benchmark functions the fuzzy combination can obtain better results but in the other functions the results are very close in terms of the original MFO and LSA algorithms.

In Table 8 the results show that increasing the dimensions increases the complexity of the problem, which is why the results are moving away from zero, but we can see that the column of the fuzzy combination obtains better results than the columns with the original algorithms except for the F8 function.

Table 9, in the same way as in the previous table, it is shown that the more complex the problem is or that it has more dimensions, the fuzzy combination may be more profitable to use since it obtains better results.

Table 6. Benchmark functions

No.	Function	Range
F1	$\sum_{i=1}^{n} x_i^2$	[-5.12, 5.12]
F2	$\sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10, 10]
F3	$\sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_i \right)$	[-100, 100]
F4	$\max\{[x_i], 1 \le i \le n\}$	[-100, 100]
F5	$\sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2 \right]$	[-30, 30]
F6	$\sum_{i=1}^{n} ([x_i + 0.5])^2$	[-100, 100]
F7	$\sum_{i=1}^{n} i x_i^4 + random[0, 1)$	[-1.28, 1.28]
F8	$\sum_{i=1}^{n} -x_i sin\left(\sqrt{\lvert x_i \rvert}\right)$	[-500, 500]
F9	$\sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12, 5.12]
F10	$-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	[-32.768, 32.768]
F11	$\frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600, 600]
F12	$\frac{\pi}{n}\left\{10\, sin^2(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 \, [1 + 10\, sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	[-50, 50]
F13	$0.1\left\{sin^2(3\pi y_i) + \sum_{i=1}^{n-1} (x_i - 1)^2 \, [1 + sin^2(3\pi y_{i+1})] + (x_n - 1)^2[1 + sin^2(2\pi y_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$ $+, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	[-50, 50]

Table 7. Experimental results with MFO, LSA and Fuzzy Combination for 5 dimensions and 500 iterations

Fun.	MFO	LSA	Fuzzy Combination
F1	1.649E-80	7.59E-157	3.533E-157
F2	1.12E-44	6.34E-83	6.272E-83
F3	1.95E-53	6.44E-84	7.583E-86
F4	1.794E-35	2.549E-70	2.253E-69
F5	12.46	1.586	2.275
F6	0	0	0
F7	2.186E-04	3.883E-04	1.787E-02
F8	-1890	-1996	-1989
F9	2.885	86.23	1.327
F10	8.88E-16	3.25E-15	1.243E-15
F11	0.07662	0.04762	0.04122
F12	6.62E-10	2.65E-07	0
F13	3.99E-09	1.36E-06	0

Table 8. Experimental results with MFO, LSA and Fuzzy Combination for 20 dimensions and 500 iterations

Fun.	MFO	LSA	Fuzzy Combination
F1	5.78E-13	2.964E-40	1.848E-39
F2	4.333	3.655E-13	9.012E-15
F3	2501	0.0006403	0.001162
F4	29.28	0.00000947	3.464E-06
F5	12150	20.98	20.18
F6	4.41E-13	0.7	1
F7	0.09774	0.007223	0.04333
F8	-6269	-6451	-6486
F9	67.69	27.26	32
F10	0.4659	0.9644	1.065
F11	0.03244	0.01943	0.02058
F12	0.088112	1.242846	0.15751
F13	0.005045	0.174869	0.00356

Table 9. Experimental results with MFO, LSA and Fuzzy Combination for 80 dimensions and 500 iterations

Fun.	MFO	LSA	Fuzzy Combination
F1	1.55E+04	7.05E-03	4.20E-02
F2	1.13E+02	1.02	1.34
F3	1.05E+05	1.95E+04	2.04E+04
F4	8.69E+01	36.9	3.89E+01
F5	2.85E+07	268.0	3.06E+02
F6	1.54E+04	19.1	1.14E+02
F7	27.3	0.185	2.98E-01
F8	-2.1E+04	-1.99E+04	-2.24E+04
F9	5.76E+02	1.99E+02	1.86E+02
F10	1.91E+01	3.29	5.44
F11	1.27E+02	5.41E-03	1.86E-02
F12	0.523797	21867861	0.47468
F13	3.947527	143748882	3.03139

Table 10. Experimental results with MFO, LSA and Fuzzy Combination for 5 dimensions and 1000 iterations

Fun.	MFO	LSA	Fuzzy Combination
F1	3.71E-81	0	0
F2	3.28E-44	6.48E-165	4.52E-166
F3	4.06E-52	9.14E-174	4.79E-173
F4	7.21E-35	3.45E-140	4.83E-140
F5	8.78E-01	1.24	1.27
F6	0	0	0
F7	2.79E-04	2.57E-04	1.34E-02
F8	-1.95E+03	-2.02E+03	-2.01E+03
F9	2.55	1.13	1.43
F10	8.88E-16	5.49E-02	5.49E-02
F11	7.84E-02	6.21E-02	3.37E-02
F12	0	7.38E-07	0
F13	0	2.152E-06	0

Table 11. Experimental results with MFO, LSA and Fuzzy Combination for 20 dimensions and 1000 iterations

Fun.	MFO	LSA	Fuzzy Combination
F1	6.99E-13	4.25E-81	1.925E-82
F2	2.333	2.65E-16	1.325E-24
F3	2834	2.23E-10	7.922E-11
F4	32.18	2.74E-12	1.654E-12
F5	12240	11.33	10.63
F6	6.06E-13	0.6333	0.8
F7	0.009386	0.00591	0.04692
F8	-6341	-6327	-6748
F9	71.53	31.97	30.68
F10	0.1542	0.7915	1.07
F11	0.02811	0.01739	0.02685
F12	0	0.55150	0
F13	0	0.25621	0

Table 12. Experimental results with MFO, LSA and Fuzzy Combination for 80 dimensions and 1000 iterations

Fun.	MFO	LSA	Fuzzy Combination
F1	1.40E+04	1.81E-09	8.71E-06
F2	1.08E+02	3.83E-01	8.58E-01
F3	9.68E+04	7.27E+03	8.25E+03
F4	8.79E+01	3.14E+01	3.31E+01
F5	2.50E+07	1.99E+02	1.67E+02
F6	1.43E+04	2.03E+01	1.34E+02
F7	5.44E+01	1.24E-01	2.52E-01
F8	-2.06E+04	-1.98E+04	-2.26E+04
F9	5.80E+02	1.98E+02	2.02E+02
F10	1.92E+01	3.57	5.63
F11	1.48E+02	4.11E-03	2.02E-02
F12	0.20298	3.519E+07	0.15478
F13	0.54241	7.085E+07	0.59517

The next three tables are using 1000 iterations, which allows the algorithm to have more time to find a better solution. In Table 10, when working for 5 dimensions and 1000 iterations, we can see that the fuzzy combination finds zeros in function 1 and in function 6 all algorithms reach zero.

Table 13. Experimental results with MFO, LSA and Fuzzy Combination for 5 dimensions and 2000 iterations

Fun.	MFO	LSA	Fuzzy Combination
F1	6.42E-165	0	0
F2	2.54E-89	2.27E-186	0
F3	3.93E-110	8.78E-101	0
F4	6.06E-71	3.31E-114	3.02E-282
F5	2.04	1.22	7.93E-01
F6	0	0	0
F7	1.24E-04	1.03E-03	1.12E-02
F8	-1.90E+03	-3.73E+03	-2.03E+03
F9	2.82	6.40	1.43
F10	8.88E-16	3.41E-01	1.01E-15
F11	1.17E-01	1.04E-01	4.19E-02
F12	0	9.433E-32	0.02526
F13	0	0.00037	0.03138

Table 14. Experimental results with MFO, LSA and Fuzzy Combination for 20 dimensions and 2000 iterations

Fun.	MFO	LSA	Fuzzy Combination
F1	1.339E-29	2.44E-169	8.912E-168
F2	1.667	7.032E-17	6.864E-18
F3	2667	1.845E-24	8.608E-24
F4	27.38	9.111E-27	2.211E-24
F5	6289	4.35	1.793
F6	2.275E-29	0.5667	0.3333
F7	0.004921	0.004996	0.0474
F8	-6457	-6433	-6690
F9	81.41	29.85	30.45
F10	0.1319	0.9823	1.038
F11	0.02352	0.01632	0.02384
F12	0	0.34955	0
F13	0	4.71827	0

In Table 11, 20 dimensions and 1000 iterations are used here, what we expect is that our fuzzy combination will be better when evaluating each of the different benchmark functions, but in this case, the results obtained are very similar to those obtained by the original LSA algorithm where by very little only in some of the functions wins our combination.

In Table 12, with 80 dimensions and 1000 iterations, the results show that in the same way the column of the LSA algorithm is the winner in most of the benchmark functions.

The next three tables show the results with 5 and 20 dimensions, but with 2000 iterations.

In Table 13, adding more time to the experiments, it is observed that at least it is the first three functions as well as in the sixth they show zeros as results and in the some others they are very close to zero.

In Table 14, we can observe the pattern of the results where 1000 iterations and 20 or 80 dimensions were used that the LSA column beasts the fuzzy combination column, that is why we do not show the table of 80 dimensions and 2000 iterations.

5 Conclusions

Based on the obtained results, it can be observed that the fuzzy combination can be a good idea to use when we are working with a complex problem in question in a more efficient way. The moth-flame optimization and lightning search algorithm were combined to obtain a powerful hybrid metaheuristic combining the advantages of both individual algorithms. In our case, evaluating the thirteen benchmark functions that are in a certain way arranged in an ascending level of complexity. That is why the results on some occasions are shown that the values reach zero with any of the algorithms, that is if is recommended that we work at least with one thousand iterations or more to allow the algorithm time to find the best solution.

In the future, we would like to perform tests with much more complicated benchmark functions to see if the algorithm with migration is really viable.

As future work, to improve the proposed method, we envision adding dynamic parameter adjustment using type-2 fuzzy systems to obtain better results.

On other hand, although this article is not about that, we would like to add improvements in terms of execution times, with the use of CUDA functions that we have used in other algorithms and they could help in saving time.

Acknowledgments

The authors would like to thanks CONACYT and Tijuana Institute of Technology for the support during this research work.

References

1 **Kawano, Y., Valdez, F., Castillo, O. (2018).** Performance Evaluation of Optimization Algorithms based on GPU using CUDA Architecture. IEEE Latin American Conference on Computational Intelligence (LA-CCI), pp. 1–6. DOI: 10.1109/LA-CCI.2018.8625236.

2 **Wang, D., Tan, D., Liu, L. (2018).** Particle swarm optimization algorithm: An overview. Soft Computing, Vol. 22, No. 2, pp. 387–408. DOI: 10.1007/s00500-016-2474-6.

3 **Sho, H. (2021).** Comparison of Centralized and Distributed Intelligent Particle Multi-Swarm Optimization on Search Performance. Artificial Intelligence Research, Vol. 10, No. 1, pp. 1–11. DOI: 10.5430/air.v10n1p1.

4 **Sánchez, D., Melin, P., Castillo, O. (2020).** Comparison of particle swarm optimization variants with fuzzy dynamic parameter adaptation for modular granular neural networks for human recognition. Journal of Intelligent & Fuzzy Systems, Vol. 38, No. 3, pp. 3229–3252. DOI: 10.3233/JIFS-191198.

5 **Yu, J.J.Q., Li, V.O.K. (2015).** A social spider algorithm for global optimization. Applied Soft Computing, Vol. 30, pp. 614–627. DOI: 10.1016/j.asoc.2015.02.014.

6 **Lai, Z., Feng, X., Yu, H., Luo, F. (2021).** A Parallel Social Spider Optimization Algorithm Based on Emotional Learning. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 51, No. 2, pp. 797–808. DOI: 10.1109/TSMC.2018.2883329.

7 **Priyadharshini, V., Divya, P., Preethi, D., Pazhaniraja, N., Paul, P.V. (2015).** A novel Web service publishing model based on social spider optimization technique. International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC),

pp. 0373–0387. DOI: 10.1109/ICCPEIC. 2015.7259488.

8 **Mirjalili, S. (2016).** Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Computing and Applications, Vol. 27, No. 4, pp. 1053–1073. DOI: 10.1007/s00521-015-1920-1.

9 **Meng, X.B., Gao, X.Z., Lu, L., Liu, Y., Zhang, H. (2016).** A new bio-inspired optimisation algorithm: Bird Swarm Algorithm. Journal of Experimental & Theoretical Artificial Intelligence, Vol. 28, No. 4, pp. 673–687. DOI: 10.1080/0952813X.2015.1042530.

10 **Bogar, E., Beyhan, S. (2020).** Adolescent Identity Search Algorithm (AISA): A novel metaheuristic approach for solving optimization problems. Applied Soft Computing, Vol. 95, pp. 1–43. DOI: 10.1016/j.asoc.2020.106503.

11 **Lagunes, M.L., Castillo, O., Valdez, F., Soria, J. (2019).** Multi-Metaheuristic Competitive Model for Optimization of Fuzzy Controllers. Algorithms, Vol. 12, No. 5, pp. 1–21. DOI: 10.3390/a12050090.

12 **Ezugwu, A.E.S., Agbaje, M.B., Aljojo, N., Els, R., Chiroma, H., Elaziz, M.A. (2020).** A Comparative Performance Study of Hybrid Firefly Algorithms for Automatic Data Clustering. IEEE Access, Vol. 8, pp. 121089–121118. DOI: 10.1109/ACCESS.2020. 3006173.

13 **Mashhour, E.M., El-Houby, Wassif, K.T., Salah, A.I. (2020).** A Novel Classifier based on Firefly Algorithm. Journal of King Saud University – Computer and Information Sciences, Vol. 32, No. 10, pp. 1173–1181, DOI: 10.1016/j.jksuci.2018.11.009.

14 **Xu, C., Meng, H., Wang, Y., (2020).** A Novel Hybrid Firefly Algorithm Based on the Vector Angle Learning Mechanism. IEEE Access, Vol. 8, pp. 205741–205754. DOI: 10.1109/ACCESS.2020.3037802.

15 **Molina, D., Poyatos, J., Del Ser, J., García, S., Hussain, A., Herrera, F. (2020).** Comprehensive Taxonomies of Nature- and Bio-inspired Optimization: Inspiration versus Algorithmic Behavior, Critical Analysis and Recommendations. Cognitive Computation, Vol. 12, No. 5, pp. 897–939. DOI: 10.1007/s12559-020-09730-8.

16 **Mirjalili, S. (2015).** Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-Based Systems, Vol. 89, pp. 228–249. DOI: 10.1016/j.knosys.2015. 07.006.

17 **Jangir, N., Pandya, M.H., Trivedi, I.N., Bhesdadiya, R.H., Jangir, P., Kumar, A. (2016).** Moth-Flame optimization Algorithm for solving real challenging constrained engineering optimization problems. IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS), pp. 1–5. DOI: 10.1109/SCEECS.2016. 7509293.

18 **Li, C., Niu, Z., Song, Z., Li, B., Fan, J., Liu, P.X. (2018).** A Double Evolutionary Learning Moth-Flame Optimization for Real-Parameter Global Optimization Problems. IEEE Access, Vol. 6, pp. 76700–76727. DOI: 10.1109/ACCESS.2018.2884130.

19 **Tumar, I., Hassouneh, Y., Turabieh, H., Thaher, T. (2020).** Enhanced Binary Moth Flame Optimization as a Feature Selection Algorithm to Predict Software Fault Prediction. IEEE Access, Vol. 8, pp. 8041–8055. DOI: 10.1109/ACCESS.2020.2964321.

20 **Shareef, H., Islam, M.M., Ibrahim, A.A., Mutlag, A.H. (2015).** A Nature Inspired Heuristic Optimization Algorithm Based on Lightning. 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS), pp. 9–14. DOI: 10.1109/AIMS.2015.12.

21 **Liu, W., Huang, Y, Zong, X., Shi, H., Ye, Z., Wei, S. (2018).** Application of lightning search algorithm in localization of Wireless Sensor Networks. 4th IEEE International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), pp. 57–61. DOI: 10.1109/IDAACS-SWS.2018.8525518.

22 **Asvany, T., Amudhavel, J., Sujatha, P. (2017).** Lightning search algorithm for solving coverage problem in wireless sensor network.

Advances and Applications in Mathematical Sciences, Vol. 17, No. 1, pp. 113–127.

23 **Abualigah, L., Elaziz, M.A., Hussien, A.G., Alsalibi, B., Jalali, S.M.J., Gandomi, A.H. (2021).** Lightning search algorithm: a comprehensive survey. Applied Intelligence, Vol. 51, pp. 2353–2376. DOI: 10.1007/s10489-020-01947-2.

24 **Valdez, F., Castillo, O., Melin, P. (2021).** Bio-Inspired Algorithms and Its Applications for Optimization in Fuzzy Clustering. Algorithms, Vol. 14, No. 4, pp. 1–21. DOI: 10.3390/a14040122.

25 **Lagunes, M.L., Castillo, O., Soria, J., Valdez, F. (2021).** Optimization of a fuzzy controller for autonomous robot navigation using a new competitive multi-metaheuristic model. Soft Computing, Vol. 25, No. 17, pp. 11653–11672. DOI: 10.1007/s00500-021-06036-1.

26 **Bernal, E., Castillo, O., Soria, J., Valdez, F., Melin, P. (2018).** A variant to the dynamic adaptation of parameters in galactic swarm optimization using a fuzzy logic augmentation. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–7. DOI: 10.1109/FUZZ-IEEE.2018.8491623.

27 **Ma, G., Zhou, W., Chang, X. (2012).** A novel particle swarm optimization algorithm based on particle migration. Applied Mathematics and Computation, Vol. 218, No. 11, pp. 6620–6626. DOI: 10.1016/j.amc.2011.12.032.

28 **Valdez, F., Melin, P., Castillo, O. (2010).** Fuzzy control of parameters to dynamically adapt the PSO and GA Algorithms. International Conference on Fuzzy Systems, pp. 1–8. DOI: 10.1109/FUZZY.2010.5583934.

29 **Alalaween, W.H., Alalawin, A.H., Mahfouf, M., Abdallah, O.H. (2021).** A Dynamic Type-1 Fuzzy Logic System for the Development of a New Warehouse Assessment Scheme. IEEE Access, Vol. 9, pp. 43611–43619. DOI: 10.1109/ACCESS.2021.3060293.

30 **Avelar, E., Castillo, O., Soria, J. (2020).** Fuzzy Logic Controller with Fuzzylab Python Library and the Robot Operating System for Autonomous Mobile Robot Navigation. Journal of Automation, Mobile Robotics and Intelligent Systems, Vol. 14, No. 1, pp. 48–54. DOI: 10.14313/JAMRIS/1-2020/6.

31 **Cuevas, F., Castillo, O., Cortés-Antonio, P. (2021).** Design of a Control Strategy Based on Type-2 Fuzzy Logic for Omnidirectional Mobile Robots. Journal of Multiple-Valued Logic & Soft Computing, Vol. 37, No. 1–2, pp. 107–136.

32 **Valdez, F., Vázquez, J.C., Melin, P., Castillo, O. (2017).** Comparative study of the use of fuzzy logic in improving particle swarm optimization variants for mathematical functions using co-evolution. Applied Soft Computing, Vol. 52, pp. 1070–1083. DOI: 10.1016/j.asoc.2016.09.024.

Article received on 30/06/2021; accepted on 20/11/2021.
Corresponding author is Oscar Castillo.

ISSN 2007-9737

Thematic Section:

Logic/Languages, Algorithms, Novel Methods of Reasoning

Guest editors:

Claudia Zepeda Cortés
José Luis Carballido Carranza
José Raymundo Marcial-Romero
Everardo Bárcenas

Computación y Sistemas, Vol. 26, No. 2, 2022, pp. 759–759
doi: 10.13053/CyS-26-2-4299

ISSN 2007-9737

Toward Relevance Term Logic

J.-Martín Castro-Manzano

UPAEP, Facultad de Filosofía,
Mexico

josemartin.castro@upaep.mx

Abstract. Term Functor Logic is a term logic that recovers some important features of the traditional, Aristotelian logic; however, it turns out that it does not preserve all of the Aristotelian properties a valid inference should have insofar as its class of theorems includes some inferences that may be considered irrelevant. Given this situation, in this contribution we tweak a tableaux method in order to avoid said irrelevance.

Keywords. Semantic trees, term logic, relevance logic.

1 Introduction

Term Functor Logic is a logic that recovers some core features of the traditional, Aristotelian logic, mainly, its term syntax; however, as we will see, it turns out that it does not preserve all of the Aristotelian properties a full-blooded inference should have insofar as its class of theorems includes some inferences that may be considered irrelevant by the lights of the Aristotelian paradigm.

Given this situation, in this contribution we advance some tentative steps towards the creation of a relevance term logic. Hence, for a more detailed exposition of the family of term logics we are considering here [24, 11, 10, 20, 26, 14] and their tableaux, we refer the reader to our previous works [7, 3, 6, 5]; meanwhile, in order to achieve our present goal, we first provide a summary of some preliminary concepts and results (i.e. syllogistic, and Term Functor Logic and its tableaux), then we briefly explain the problem (how irrelevance is parasitic of Term Functor Logic) and, finally, we suggest a possible solution by tweaking a tableaux method.

2 Preliminaries

Syllogistic is a term logic that has its origins in Aristotle's *Prior Analytics* [1] and deals with inference using categorical statements. A *categorical statement* is a statement composed by two terms, a quantity, and a quality. The subject and the predicate of a statement are called *terms*: the term-schema S denotes the subject term of the statement and the term-schema P denotes the predicate. The *quantity* may be either universal (*All*) or particular (*Some*) and the *quality* may be either affirmative (*is*) or negative (*is not*). These categorical statements have a *type* denoted by a label (either a (universal affirmative, SaP), e (universal negative, SeP), i (particular affirmative, SiP), or o (particular negative, SoP)) that allows us to determine a *mood*, that is, a sequence of three categorical statements ordered in such a way that two statements are premises (major and minor) and the last one is a conclusion. A *categorical syllogism*, then, is a mood with three terms one of which appears in both premises but not in the conclusion. This particular term, usually denoted with the term-schema M, works as a link between the remaining terms and is known as the middle term. According to the position of this middle term, four *figures* can be set up in order to encode the valid syllogistic moods (Table1).[1]

This quick overview of syllogistic, though formally correct, is a little bit out of context. Syllogistic is an integral part of what we could call a basic *corpus aristotelicum* that, in turn, could be defined by the tuple $\mathfrak{A} = \langle Th_E, Th_C, Th_O, Th_P, Th_L \rangle$

[1] For sake of brevity, but without loss of generality, here we omit the syllogisms that require existential import.

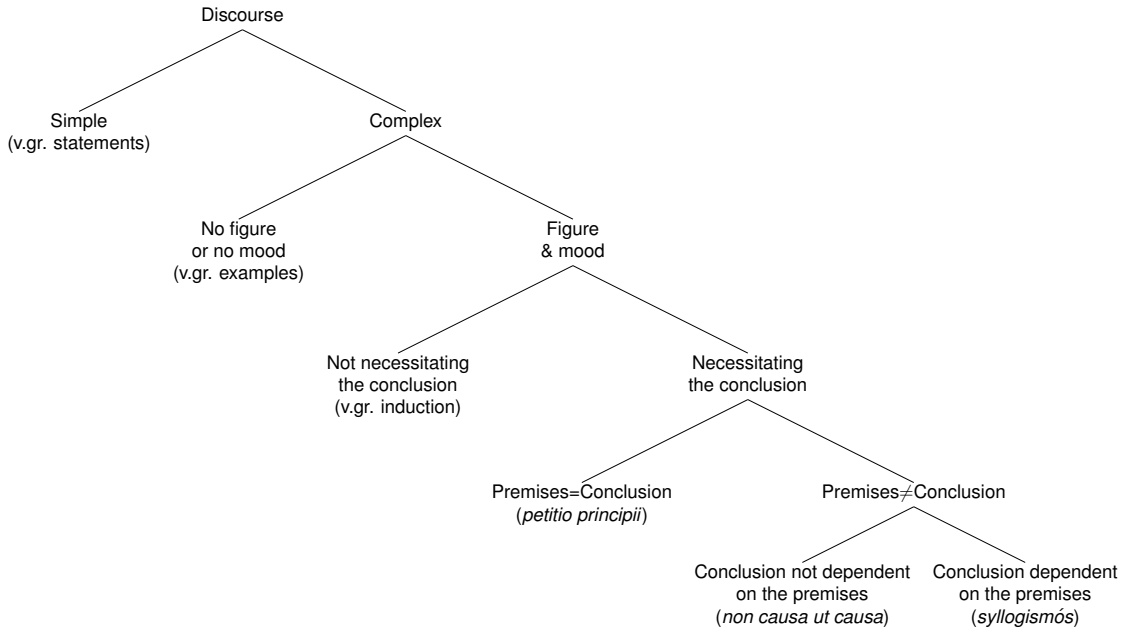


Fig. 1. The Boethian exposition of syllogistic (adapted from [27, p.44])

Table 1. Valid syllogistic moods by figure

First	Second	Third	Fourth
aaa	eae	iai	aee
eae	aee	aii	iai
aii	eio	oao	eio
eio	aoo	eio	

(cf. [16, p.4ff]) where Th_E is an epistemological theory that includes the production of hypothesis and inferences under the Aristotelian concepts of *epagogé* and *syllogismós*, respectively[2]; Th_C is a theory of causality that distinguishes material, formal, efficient, and final causes[3]; Th_O is an ontological theory that assumes a systemic view of the world given the double claim that there are no unhad properties (*contra* universals *ante rem*) nor objects without properties (*contra* bare particulars); Th_P is a psychological theory that makes good use of the concept "habit" in order to explain behavior (both *epagogé* and *syllogismós*, for example, would be habits when performed by agents); and Th_L is a logical theory designed for understanding categories, statements, inferences, explanations, and cognitive biases.

This last theory includes syllogistic as a theory of deductive inference but, as we have tried to imply, it has some specific semantic requirements related to the other components of the *corpus*. Hence, the formal description of syllogistic that we have given above lacks a quality that may be better understood given the previous context: syllogistic is a deductive theory designed to avoid causal irrelevance. In order to illustrate this last point consider Thom's explanation of Kilwardby's first exposition of syllogistic—also called the Boethian exposition (Figure 1).

[2]The concept *epagogé* refers to some sort of essential induction, so to speak, that is different from a numerical induction. This difference helps explain why some general statements are admissible (v.gr., *All human beings are living beings*) while others are not (v.gr., *All human beings are mexican*). The concept *syllogismós*, on the other hand, will be treated with more detail below.

[3]Notice this concept of cause is different, for example, from our current idea of a factor: two material factors may explain a state of affairs, and hence we may have a multi-factorial explanation of said state, but such an explanation needs not be a multicausal explanation, since both factors are instances of material causes.

This Boethian exposition clarifies that, within the Aristotelian way of thinking or paradigm, a syllogistic inference or syllogism—*syllogismós*—is a piece of complex discourse (insofar as it includes at least two premises and one conclusion) with mood and figure (because the order of statements and terms matters) in which a conclusion that is different from the premises (thus avoiding *petitio principii*) necessarily (and hence deductively) follows from and dependes on said premises (thus avoiding irrelevance, *non causa ut causa*).

This Aristotelian view of inference should not be understated because it differs from the contemporary, Fregean-Tarskian approach, at least in three respects: *i)* the contemporary approach takes it that content and form are independent (as when the usual logic handbooks claim, almost dogmatically, that logic does not deal with truth, but with validity), yet that independence is not crystal clear (cf. [2]); whereas in the Aristotelian approach content and form are systemic and codependent (as when Aristotle distinguishes between natural and unnatural predication (cf. [12, 13])). *ii)* The contemporary approach usually follows the Fregean paradigm that results from droping the ternary syntax of traditional logic (subject-copula-predicate) in order to promote a binary syntax (function-argument) imported from mathematics, which turns out to be not that natural (cf. [12, 28, 13]). And *iii)* the contemporary approach admits reflexivity (i.e. $p \vdash p$) both as a valid pattern of inference and, sometimes, as the principle of identity (i.e. $\vdash p \Rightarrow p$); whereas the Aristotelian approach rejects the former (i.e. $p \nvdash p$) while admits a version of the latter (i.e. $\vdash p \Rightarrow p$). We will return to some of these issues later.

2.1 Term Functor Logic and its Tableaux

Term Functor Logic (TFL, for short) [24, 26, 9, 11, 14] is a plus-minus algebra that employs terms and functors rather than first order language elements such as individual variables or quantifiers (cf. [23, 21, 15, 24, 25, 19]). According to this algebra, the four categorical statements can be represented by the following syntax [11]:

a. $\mathsf{SaP} := -\mathsf{S} + \mathsf{P}$,

b. $\mathsf{SeP} := -\mathsf{S} - \mathsf{P}$,

c. $\mathsf{SiP} := +\mathsf{S} + \mathsf{P}$,

d. $\mathsf{SoP} := +\mathsf{S} - \mathsf{P}$.

Table 2. A valid syllogism

	Statement	TFL
1.	All computer scientists are animals.	$-\mathsf{C} + \mathsf{A}$
2.	All logicians are computer scientists.	$-\mathsf{L} + \mathsf{C}$
$\vdash$	All logicians are animals.	$-\mathsf{L} + \mathsf{A}$

Given this representation, TFL provides a simple rule for syllogistic inference: a conclusion follows validly from a set of premises if and only if *i)* the sum of the premises is algebraically equal to the conclusion and *ii)* the number of conclusions with particular quantity (viz., zero or one) is the same as the number of premises with particular quantity [11, p.167]. Thus, for instance, if we consider a valid syllogism (say, a syllogism aaa of the first figure, aaa-1), we can see how the application of this rule produces the right conclusion (Table 2).

In this example we can clearly see how the rule works: *i)* if we add up the premises we obtain the algebraic expression $(-\mathsf{C} + \mathsf{A}) + (-\mathsf{L} + \mathsf{C}) = -\mathsf{C} + \mathsf{A} - \mathsf{L} + \mathsf{C} = -\mathsf{L} + \mathsf{A}$, so that the sum of the premises is algebraically equal to the conclusion and the conclusion is $-\mathsf{L} + \mathsf{A}$, rather than $+\mathsf{A} - \mathsf{L}$, because *ii)* the number of conclusions with particular quantity (zero in this case) is the same as the number of premises with particular quantity (zero in this case)[4]. In contrast, for sake of comparison, consider an invalid syllogism (aaa-3) that does not add up (Table 3).

Now, as exposed in [7, 4] and following [8, 22], we can develop a tableaux proof method for TFL. Hence, we say a *tableau* for TFL is an acyclic connected graph determined by nodes and vertices. The node at the top is called *root*. The nodes at the bottom are called *tips*. Any path from

[4] Although we are exemplifying this logic with syllogistic inferences, this system is capable of representing relational, singular, and compound inferences with ease and clarity. Furthermore, TFL is arguably more expressive than classical first order logic [9, p.172].

Table 3. An invalid syllogism

	Statement	TFL
1.	All computer scientists are animals.	$-\mathsf{C}+\mathsf{A}$
2.	All computer scientists are logicians.	$-\mathsf{C}+\mathsf{L}$
$\nvdash$	All logicians are animals.	$-\mathsf{L}+\mathsf{A}$

Fig. 2. TFL expansion rules

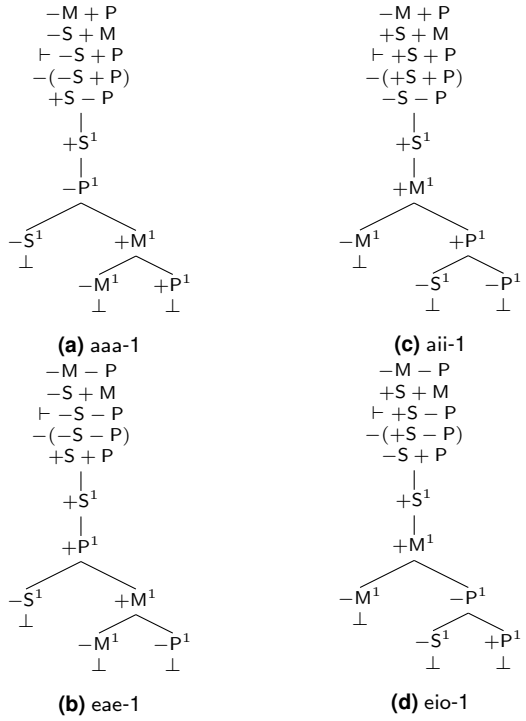


Fig. 3. Valid syllogistic moods of the first figure

the root down a series of vertices is a *branch*. To test an inference for validity we construct a tableau which begins with a single branch at whose nodes occur the premises and the rejection of the conclusion: this is the *initial list*. We then apply the rules that allow us to extend the initial list (Figure 2).

Figure 2a depicts the rule for a (e) statements, while Figure 2b shows the rule for i (o) statements. After applying a rule we introduce some index $i \in \{1, 2, 3, \ldots\}$. For statements a and e, the index may be any natural; for statements i and o, the index has to be a new natural if they do not already have an index. Also, following TFL tenets, we assume the following rules of rejection: $-(\pm\mathsf{T}) = \mp\mathsf{T}$, $-(\pm\mathsf{T} \pm \mathsf{T}) = \mp\mathsf{T} \mp \mathsf{T}$, and $-(--\mathsf{T}--\mathsf{T}) = +(-\mathsf{T}) + (-\mathsf{T})$.

A tableau is *complete* if and only if every rule that can be applied has been applied. A branch is *closed* if and only if there are terms of the form $\pm\mathsf{T}^i$ and $\mp\mathsf{T}^i$ on two of its nodes; otherwise it is *open*. A closed branch is indicated by writing a $\perp$ at the end of it; an open branch is indicated by writing ∞. A tableau is *closed* if and only if every branch is closed; otherwise it is *open*. So, as usual, $\pm\mathsf{T}$ is a logical consequence of the set of terms Γ (i.e. $\Gamma \vdash \pm\mathsf{T}$) if and only if there is a closed complete tableau whose initial list includes the terms of Γ and the rejection of $\pm\mathsf{T}$ (i.e. $\Gamma \cup \{\mp\mathsf{T}\} \vdash \perp$). Accordingly, we provide some examples (Figure 3).

To describe the process we follow to unfold each tableaux consider Figure 3a (cf. [4]). The first three lines are the premises and the conclusion, and the fourth line is the rejection of the conclusion: all these lines but the conclusion define the initial list. Then the fifth line is the result of applying a rule of rejection to the conclusion. Then the next couple of lines is the result of applying the rule for an i proposition to the fifth line, picking index 1. Then the first split results from applying the rule for an a proposition to the second line (i.e. the minor premise), also picking index 1, since we want the indexes to unify. This split produces two branches, one of which (the leftmost) includes terms $+\mathsf{S}^1$ and $-\mathsf{S}^1$ on two of its nodes, and hence is closed; the remaining branch is not closed yet, so we continue with the same process: we split the last available premise (i.e. the major premise) to obtain, again, a couple of branches, one of which (the leftmost) includes terms $-\mathsf{M}^1$ and $+\mathsf{M}^1$ on two of its nodes, and hence is closed; and the other (the rightmost) that contains terms $+\mathsf{P}^1$ and $-\mathsf{P}^1$ on two of its nodes, and hence is closed as well.

Table 4. Some problematic inferences

I	II	III	IV
1.$-\mathsf{A}+\mathsf{B}$	1.$\pm\mathsf{B}$	1.$-\mathsf{A}+\mathsf{B}$	1. $\pm\mathsf{A}$
2.$+\mathsf{A}-\mathsf{B}$	$\vdash -\mathsf{A}+\mathsf{A}$	2.$-\mathsf{C}+\mathsf{A}$	$\vdash \pm\mathsf{A}$
$\vdash -\mathsf{A}+\mathsf{A}$		$\vdash -\mathsf{C}+\mathsf{A}$	

Fig. 4. RTL expansion rules

3 Toward Relevance Term Logic

At this point it should be clear that TFL recovers some syntactical features of the traditional, Aristotelian logic, particularly, a term syntax; however, it turns out that it does not preserve all of the Aristotelian properties a proper inference should have because its class of theorems includes some inferences that can be considered irrelevant by the lights of the Boethian exposition and the Aristotelian paradigm. In order to exemplify this issue consider the problematic inferences shown in Table 4.

Such inferences are problematic because all of them are valid in TFL (cf. [26]) (as well as in classical First Order Logic, we should add), and yet, they cannot be valid within an Aristotelian framework: inference I is a case of *ex contradictione sequitur quodlibet* (ECSQ)—i.e. a contradiction implies anything—; inference II is an instance of the (positive) paradox of implication—i.e. a tautology is implied by anything—; inferences III and IV are instances of *petitio principii*. But then there is an impasse: while TFL is close to an Aristotelian notion of inference (given its syntatical features), it is still far from being a relevance logic in an Aristotelian sense (since irrelevance is parasitic of TFL). To solve this deadlock, consider the proposal given in Figure 4.

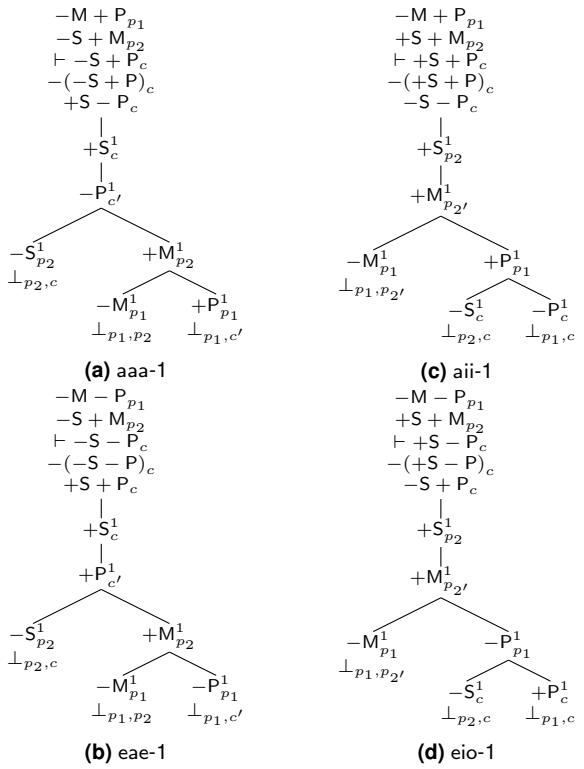
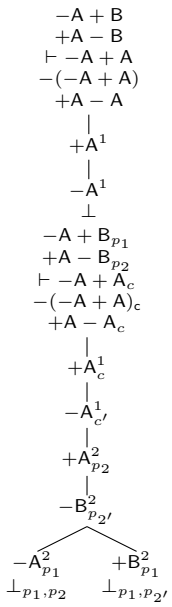
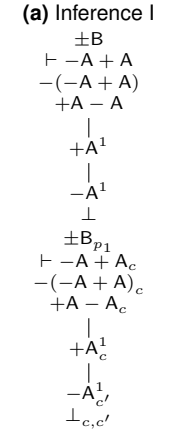


Fig. 5. Valid syllogistic moods within the first figure (again)

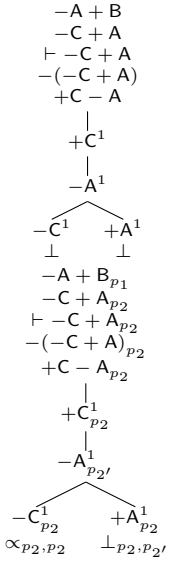
These Relevance Term Logic (RTL) tableaux rules behave as the tableaux rules for TFL, but notice that besides the indexes, we introduce and keep a flag $f, f' \in \{p_i, c\}$ for $i \in \{1, 2, 3, \ldots\}$ (p for premise, c for conclusion). Now we say a branch is *open* if and only if there are no terms of the form $\pm\mathsf{T}^i$ and $\mp\mathsf{T}^i$ on it; a branch is *semi-open* (or *semi-closed*) if and only if there are terms of the form $\pm\mathsf{T}^i_f$ and $\mp\mathsf{T}^i_f$; otherwise it is *closed*. An open branch is indicated by writing ∞ at the end of it; a semi-open (semi-closed) branch is indicated by writing $\propto_{f,f}$ ($\infty_{f,f}$); and a closed branch, as usual, is denoted by $\perp_{f,f'}$. We say a tableau is *Aristotelian* if and only if every branch is closed and all the flags are carried at the end of every tip; a tableau is *open* if and only if it has an open branch; otherwise, it is *classical*. The rest of definitions is as usual.



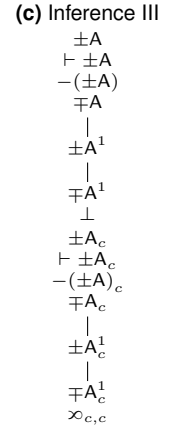
(a) Inference I



(b) Inference II



(c) Inference III



(d) Inference IV

Fig. 6. Some problematic inferences: TFL (above) vs RTL (below)

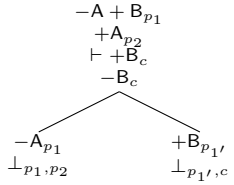


Fig. 7. *Modus Ponens*

Accordingly, reconsider and compare the basic syllogistic moods—they are correct both in TFL and in RTL (Figure 5)—and the problematic inferences shown in Table 4 above—even though they are classically valid, they are not Aristotelian (Figure 6).

We think the examples shown in Figure 5 are self-explanatory, but perhaps a brief description of Figure 6 may help explain further the use of these rules. So, Figure 6a shows an instance of ECSQ. We can see that the TFL tree is just closed, whereas the RTL tree is also closed but is not Aristotelian because the closure does not use any conclusion (i.e. the premises are not relevant to the conclusion). Figure 6b shows an instance of a paradox of implication and, while the TFL tree is just closed, the RTL tree is closed but not all the flags are carried to the tips, and hence the conclusion is not relevant to the premise. Figure 6c and 6d show instances of *petitio principii*: observe that while the corresponding TFL trees are closed, the RTL trees are semi-open (semi-closed) because the closure does not use the conclusion or the premises (i.e. the conclusion is not relevant to the premises or vice versa). This means that these inferences, although truth preserving, are not relevant; and hence, while they are not to be regarded as full-blooded inferences, they should not be discarded altogether as totally wrong inferences.

Additionally consider, just out of curiosity, some inferences in order to suggest that this proposal seems to be suitable for non-syllogistic logic. Take an instance of a *Modus Ponens* for propositional logic (Figure 7), and take an instance of a relational inference (say, "since every B loves some G and every G is W, it follows that every B loves something W") for relational logic (Figure 8). This particular examples would suggest said inferences are not only classically valid or truth preserving, but also relevant in an Aristotelian sense.

Our claim, thus, is that this proposal moves TFL into the direction of a relevance logic that is skeptical of both *petitio principii* and *non causa ut causa* inferences. So, in a sense, we are saying that:

Theorem 1 *RTL is Aristotelian.*

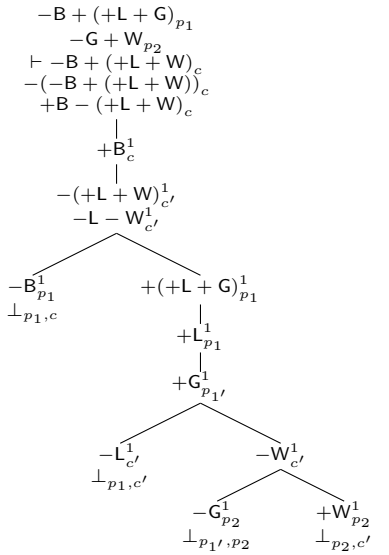


Fig. 8. A relational inference

Indeed, *i)* RTL is wary of *petitio principii* (i.e. instances of inferences such as III and IV). Aristotle suggested that a *petitio principii* is a fallacy because it fails to account for a causal explanation since it depends upon assuming what has to be explained (*De Sophisticis Elenchis* 168b23-27). It is a requirement of a legitimate inference that the conclusion (i.e. what has to be explained) has to be different from the premises (*Topics* 100a25-26, *De Sophisticis Elenchis* 165a1-2, *Prior Analytics* 24b19-20).

ii) RTL is wary of *non causa ut causa* (i.e. instances of inferences such as I and II). Contemporary, classical First Order Logic admits both the rule ECSQ and the paradoxes of implication as patterns of valid inference, but this view allows some sort of irrelevance that Aristotle did not quite accept (*Prior Analytics* 2 4-57b3): this sort of irrelevance, as we have seen, is parasitic of TFL as well.

iii) Finally, RTL avoids transforming the First Principle (i.e. the identity principle) into the First Fallacy (i.e. *petitio principii*), as [18] would put it (inference IV)—of course, our proposal is far from being as sophisticated as theirs, but we believe it could be useful.

4 Final remarks

As we have tried to show, Term Functor Logic is an alternative logic that recovers some important features of the traditional, Aristotelian logic but, as we have seen, it does not preserve all of the Aristotelian properties a proper inference should have insofar as its class of theorems includes some inferences that may be considered irrelevant. Since this situation is problematic, in this contribution we have offered a preliminary, provisional tableaux method for a relevance logic version of Term Functor Logic; nevertheless, given the current scope and the space limitations of this tentative research, we believe our immediate challenges include, at least: *i)* checking the (in)validity of more (non-)problematic inferences and looking for soundness and completeness; *ii)* offering a cogent, philosophical interpretation of the proposed method (say, in terms of *propter quid* and *quia* inferences); *iii)* reverse engineering the rules of RTL into TFL; and *iv)* further discussing the place of this proposal, if any, within the current literature about relevance logic (cf. [17]). We are currently working on these issues.

Acknowledgments

We would like to thank anonymous reviewers for corrections and suggestions. This work was funded by an UPAEP Research Grant.

References

1. **Aristotle (1989).** Prior Analytics. Hackett Classics Series. Hackett.
2. **Cabrera, J. (2003).** Es realmente la lógica tópicamente neutra y completamente general. Ergo, Nueva Época, Revista de Filosofía, Vol. 12, pp. 7–33.
3. **Castro-Manzano, J. M. (2019).** Silogística intermedia, términos y árboles. Tópicos, Revista De Filosofía, Vol. 58, pp. 209–237.
4. **Castro-Manzano, J.-M. (2020).** Distribution tableaux, distribution models. Axioms, Vol. 9, No. 2.

5. **Castro-Manzano, J. M. (2020).** Murphree's numerical term logic tableaux. Electronic Notes in Theoretical Computer Science, Vol. 354, pp. 17–28. Proceedings of the Eleventh and Twelfth Latin American Workshop on Logic/Languages, Algorithms and New Methods of Reasoning (LANMR).
6. **Castro-Manzano, J. M. (2020).** Un método de árboles para la silogística modal. Open Insight, Revista De Filosofía, Vol. 58, pp. 209–237.
7. **Castro-Manzano, J. M., Reyes-Cardenas, P.-O. (2018).** Term functor logic tableaux. South American Journal of Logic, Vol. 4, No. 1, pp. 9–50.
8. **D'Agostino, M., Gabbay, D. M., Hähnle, R., Posegga, J. (1999).** Handbook of Tableau Methods. Springer.
9. **Englebretsen, G. (1987).** The New Syllogistic. P. Lang.
10. **Englebretsen, G. (1988).** Preliminary notes on a new modal syllogistic. Notre Dame J. Formal Logic, Vol. 29, No. 3, pp. 381–395.
11. **Englebretsen, G. (1996).** Something to Reckon with: The Logic of Terms. Canadian electronic library: Books collection. University of Ottawa Press.
12. **Englebretsen, G. (2013).** Robust Reality: An Essay in Formal Ontology. Philosophische Analyse / Philosophical Analysis. De Gruyter.
13. **Englebretsen, G. (2017).** Bare Facts and Naked Truths: A New Correspondence Theory of Truth. Taylor & Francis.
14. **Englebretsen, G., Sayward, C. (2011).** Philosophical Logic: An Introduction to Advanced Topics. Bloomsbury Academic.
15. **Kuhn, S. T. (1983).** An axiomatization of predicate functor logic. Notre Dame J. Formal Logic, Vol. 24, No. 2, pp. 233–241.
16. **Losee, J. (2001).** A Historical Introduction to the Philosophy of Science. OUP Oxford.
17. **Mares, E. (2004).** Relevant Logic: A Philosophical Interpretation. Cambridge University Press.
18. **Meyer, R., Martin, E. (2019).** S (for syllogism) revisited. The Australasian Journal of Logic, Vol. 16, No. 3, pp. 49–67.
19. **Moss, L. (2015).** Natural logic. In **Lappin, S., Fox, C.**, editors, The Handbook of Contemporary Semantic Theory. John Wiley & Sons.
20. **Murphree, W. A. (1998).** Numerical term logic. Notre Dame J. Formal Logic, Vol. 39, No. 3, pp. 346–362.
21. **Noah, A. (1980).** Predicate-functors and the limits of decidability in logic. Notre Dame J. Formal Logic, Vol. 21, No. 4, pp. 701–707.
22. **Priest, G. (2008).** An Introduction to Non-Classical Logic: From If to Is. Cambridge Introductions to Philosophy. Cambridge University Press.
23. **Quine, W. V. O. (1971).** Predicate functor logic. **Fenstad, J. E.**, editor, Proceedings of the Second Scandinavian Logic Symposium, North-Holland.
24. **Sommers, F. (1982).** The Logic of Natural Language. Clarendon Library of Logic and Philosophy. Oxford University Press.
25. **Sommers, F. (2005).** Intelectual autobiography. In **Oderberg, D. S.**, editor, The Old New Logic: Essays on the Philosophy of Fred Sommers. Bradford book, pp. 1–24.
26. **Sommers, F., Englebretsen, G. (2000).** An Invitation to Formal Reasoning: The Logic of Terms. Ashgate.
27. **Thom, P. (2007).** Logic and Ontology in the Syllogistic of Robert Kilwardby. Studien Und Texte Zur Geistesgeschichte Des Mittelalters. Brill.
28. **Woods, J. (2016).** Logic Naturalized. Springer International Publishing, Cham, pp. 403–432.

Article received on 09/10/2020; accepted on 18/02/2021.
Corresponding author is J-Martín Castro-Manzano.

ISSN 2007-9737

Two-agent Approximate Agreement from an Epistemic Logic Perspective

Jorge Armenta-Segura[1], Jeremy Ledent[2], Sergio Rajsbaum[1]

[1] Universidad Nacional Autónoma de México,
Insituto de Matemáticas,
Mexico

[2] University of Strathclyde,
GMSP group,
Scotland

jesusarmenta@ciencias.unam.mx, jeremy.ledent@strath.ac.uk, rajsbaum@im.unam.mx

Abstract. We investigate the two agents approximate agreement problem in a dynamic network in which topology may change unpredictably, and where consensus is not solvable. It is known that the number of rounds necessary and sufficient to guarantee that the two agents output values $1/k^3$ away from each other is k. We distil ideas from previous papers to provide a self-contained, elementary introduction, that explains this result from the epistemic logic perspective.

Keywords. Distributed algorithm, approximate consensus, fault-tolerance, epistemic logic.

1 Introduction

Problems of reaching agreement are central to distributed computing. Often *consensus* is needed to agree on the same value, for example, when agents need to agree on whether to commit or abort the results of a distributed database transaction. However, in many situations consensus is impossible to achieve, such as networks where agents may crash and delays are unpredictable [15], or read/write shared memory models [22], or certain dynamic networks [11]. All these impossibility results are due to the same reason: the indistinguishability graph [2] of the global states of a protocol trying to solve consensus is connected, while that of the consensus task is not [5, 18, 23].

In this paper we are interested in many other situations, where *approximate agreement* is sufficient; for example, when sensors estimate a certain measurement or clock synchronization where agents maintain similar time estimates. Many variants of approximate agreement, and in various message passing and shared memory models have been considered since early on, e.g., [14].

Approximate agreement is an interesting weakening of consensus that can be solved in many more situations. The task is parametrized by a real number $\varepsilon > 0$, and the agents must decide values that are at most ε away from each other. The time to reach a decision however, will be a function of how small ε is. Many algorithms have been proposed to try to minimize the time until decision is reached, e.g. in shared memory models [20], in networks [10], and others.

Consensus has been thoroughly studied from the epistemic logic perspective, and the close connection with common knowledge is well-known [24]. Approximate agreement has been less studied from the epistemic logic perspective. It was recently shown that it is closely related to k-iterated knowledge in a certain shared memory model [25], roughly, meaning that the agents must know, that they know, that they know that they know, and so forth (k times), each other's inputs,

in order to reach certain degree of approximation of their decisions. This was shown within a general simplicial complex model for multi-agent epistemic logic for task solvability. In this paper we are interested in studying in more detail this result, as well as providing a self-contained, more elementary introduction to the topic.

We focus on one specific distributed computing model, that has received much attention recently, *dynamic networks* (see surveys [6, 8, 21, 27]). There is a fixed set of agents that operate in rounds communicating by sending messages to each other. Rounds are synchronous in the sense that the messages received at some round have been sent at that round. At each round, the communication graph is chosen arbitrarily among a set of directed graphs, $\mathcal{G}$. This set $\mathcal{G}$ determines the network model. Hence the communication graph can change unpredictably from one round to the next.

Dynamic networks are interesting for various reasons, in particular because they include as special cases other classic models, such as shared memory models [19], so results in dynamic networks can often be extrapolated to other models.

Approximate agreement is solvable in a dynamic network model $\mathcal{G}$ if and only if each communication graph in $\mathcal{G}$ has a rooted spanning tree [9] (e.g. [5, 7]). In this work we consider such a $\mathcal{G}$, where approximate agreement is solvable, but not consensus. Two agents, starting with binary input values, are to decide values in the interval of their inputs, which are at most $1/3^k$ apart from each other[1]. We rephrase this problem epistemically, as requiring that the agents reach k-iterated knowledge about their respective inputs. This knowledge can be achieved, if and only if, the protocol runs for k rounds. We show these results in the dynamic epistemic logic (DEL) [4] framework of [25], instantiated to a dynamic network. We follow recent work, in using the dual of a Kripke graph, a simplicial complex, as a model for multi-agent epistemic logic, that exposes the topological features that determine if a task is solvable [17, 26]. While the results we present here have been shown in these papers, we instantiate the results using graphs instead of simplicial complexes, making the development more accesible.

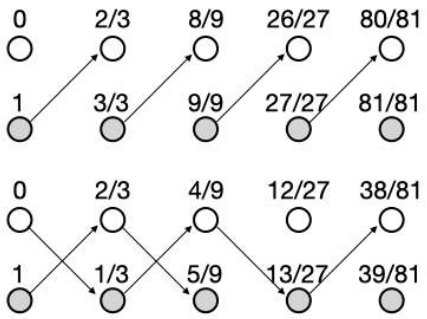
Organization. Before getting into epistemic logic, in Section 2 we present two algorithms that solve approximate agreement. In Section 3 we overview the DEL framework. In Section 4 we give the formal epistemic logic semantics to our dynamic network model, and we do the same for tasks, in Section 5. The solvability results are in Section 6. The conclusions are in Section 12. Additional details and proofs are in the Appendix.

2 Approximate Agreement Algorithms

We start by presenting algorithms, to guide the reader's intuition before diving into the formal semantics considerations of the subsequent sections.

Two agents, g, w (for gray, white), communicate with each other exchanging messages in a sequence of k rounds. In a *round,* each agent sends a message to the other agent, receives the message sent by the other agent, and then updates its local state. In each round, at most one of the two messages sent may be lost.

Let $N = 3^k$, $k \geq 0$. In the N-*approximate agreement* task, after k rounds, agents decide values d_g, d_w, resp., of the form i/N, for an integer i, $0 \leq i \leq N$. Each agent starts with an input value from the set $\{0, 1\}$:



The decisions must be such that if the two input values are equal, then both output values

[1] Fixing the constant in ε to $1/3^k$ facilitates the presentation, avoiding collateral details, and can be done without loss of generality.

AVERAGING N-APPROXIMATE AGREEMENT (ℓ)

round r **from** 1 **to** k **do**
  send(ℓ)
  $m = receive()$ /* receive $m \in \{0, 1, \bot\}$ */
  if $m \neq \bot$ **then**
    $\ell = \ell/3 + 2m/3$ /* else ℓ does not change */
ouput ℓ

Fig. 1. Each agent $p \in \{g, w\}$ runs this code. The input is $\ell \in \{0, 1\}$. A message $m = \bot$ indicates that no message was received from the other process

ONE-BIT N-APPROXIMATE AGREEMENT (ℓ) /* input $\ell \in \{0, 1\}$ */

view = $()$ /* start with empty view */
$\mathrm{nbMsg} = 0$ /* total messages received until now */
round r **from** 1 **to** k **do**
  send($(\mathrm{nbMsg} + \ell) \bmod 2$) /* send $\mathrm{nbMsg} + \ell$'s parity */
  $m = receive()$ /* receive $m \in \{0, 1, \bot\}$ */
  if $m \neq \bot$ **then** $\mathrm{nbMsg} = \mathrm{nbMsg} + 1$ /* update nbMsg */
  view = view $\cdot\, m$ /* append m to the view */
output $\delta(\ell, \mathrm{view})$

Fig. 2. Approximate agreement with 1-bit messages. Each agent runs this code with input $\ell \in \{0, 1\}$; the output function $\delta(\ell, \mathrm{view})$ is detailed in Figure 3

are equal to the input value. Otherwise, the output values d_g, d_w satisfy $|d_g - d_w| \leq 1/N$. The algorithm in Figure 1 solves N-approximate agreement in k rounds (e.g [10, 16] and [18] Chapter 2). A simple induction argument on the number of rounds proves its correctness. The examples on the right show two 4-round executions where agents agree on values $1/3^4 = 1/81$ away from each other. The first example (represented in the top two rows) represent an execution where all messages from w are lost. The first column, with labels 0 and 1, corresponds to the initial input values. Then each subsequent column shows the values of the processes after each round. An edge indicates that a message was received. In the second execution (bottom two rows of the picture), both messages arrive in the first two rounds, but in the 3rd round the message from g to w is lost, and in the 4th round the message from w to g.

The algorithm of Figure 1 can be used by the agents to decide values arbitrarily close to each other, by running enough rounds, k. However, the size of messages sent grows with k. Remarkably, there is an algorithm that solves the problem sending 1-bit messages, see Figure 2, which is a reformulation and small adaptation of the algorithm given in [12]. In a 1-bit protocol, each message received from the other process can be either 0, 1 or $\bot$. Thus, in a k-round computation, we call the *view* of a process the sequence $v \in \{0, 1, \bot\}^k$ of messages that were received. For instance with four rounds, $v = (0, 0, \bot, 1)$ indicates that the process received message "0" in the first two rounds, no message in the third round, and message "1" in the fourth round. To reduce notation, we often write such a view as a word, e.g. $v = 00\bot1$. Given a view v and a message $m \in \{0, 1, \bot\}$, we write $v \cdot m$ for the new view where we append m at the end of v.

The algorithm in Figure 2 below is generic, in the sense that both processes simply collect their view during the computation, and at the very end a decision function δ computes the output depending on the view. To choose which bit to send, at each round, the process counts how many messages were received until now, $\mathrm{nbMsg} = \#\{m \in \mathrm{view} \mid m \neq \bot\}$, adds its input $\ell \in \{0, 1\}$, and sends the parity of the result. The computation of δ is done locally at the end of the protocol, and does not involve communication between the processes. It is adapted from [12], except that we deal with all four combinations of inputs in $\{0, 1\}$ instead of fixing in advance one input edge. We use the following facts:

— The first time a process p receives a message, it can immediately guess the input of the other process p'. Indeed, since all previous messages were lost by p, we know for sure that all messages were received by p'.

— Process p can then use this information to "decode" the subsequent messages, and behave like in the algorithm of [12].

The rest of the paper is devoted to answering the question: what do the agents learn about their inputs after running protocols like these ones? For these two specific cases, do they learn something different? We will give a formal semantics based on dynamic epistemic logic to answer such questions. And as an application, we will show that these two algorithms are optimal in the number of rounds: in

ISSN 2007-9737

LOCAL COMPUTATION OF $\delta(\ell, \text{view})$

other = $\perp$ /* other's input value */
d = 0 /* distance to move towards the other */
for $i = 1$ **to** k **do**
  nbMsg = $\#\{x \in \text{view}[1...(i-1)] \mid x \neq \perp\}$
  m = view$[i]$
  if $m \neq \perp$ **then** /* if $m = \perp$, keep the same d */
    if other = $\perp$ **then**
      other = $(m + i + 1) \bmod 2$ /* find other's input*/
      if other = ℓ **then return** ℓ
    $m = (m + other) \bmod 2$ /* decode the message */
    if ($m =$ nbMsg and $i \bmod 2 = 1$) or
    ($m \neq$ nbMsg and $i \bmod 2 = 0$)
      then $d = d + 2/3^i$
      else $d = d - 2/3^i$
if $\ell = 0$ **then return** d
else return $1 - d$

Fig. 3. Decision function $\delta(\ell, \text{view})$ used at the end of the one-bit algorithm of Figure 2

less than k rounds, it is impossible for the agents to produce outputs closer than $1/3^k$.

This result has been proved using combinatorial arguments e.g. [16, 18], here we will see a different explanation: in less than k rounds, they cannot acquire sufficient knowledge about their inputs.

Also, while the two algorithms seem rather different, the agents learn exactly the same information about their inputs in both.

3 Fundamentals about Dynamic Epistemic Logic (DEL)

3.1 A Simplicial Model for Epistemic Logic

We describe here the model for epistemic logic, based on chromatic simplicial complexes [25, 26]. This reformulates the usual semantics of formulas in Kripke models, in terms of simplicial models. See Appendix 8 and Theorem 8.2.

Syntax Let AP be a countable set of propositional variables and A a finite set of agents. The language of epistemic logic formulas $\mathcal{L}_{\mathcal{K}}(A, AP)$, or just $\mathcal{L}_{\mathcal{K}}$ if A and AP are implicit, is generated by the following BNF grammar:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi \qquad p \in AP,\ a \in A.$$

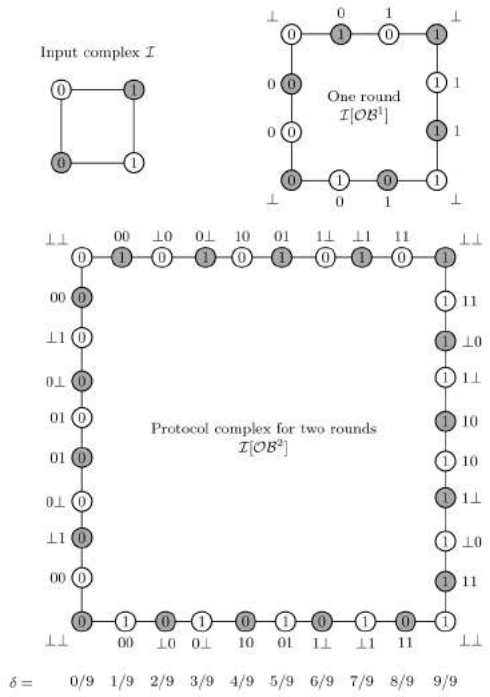


Fig. 4. Protocol complex of the 1-bit protocol from Figure 2, for 1 and 2 rounds of computation. Process names are represented by colors; inputs $\ell \in \{0, 1\}$ are written inside the vertices; and views are written next to the vertices. For the two-round protocol, the value of $\delta(\ell, \text{view})$ at each node is given by projecting vertically downwards. These pictures represent protocol complexes in the usual sense of distributed computing, but also in the sense of the *product-update model* construction defined later

The proof theory of epistemic logics can be found in [13]; in the rest of the paper, we focus on studying models. In the following, the number of agents A is two and we let $A = \{g, w\}$. The agents will be represented as colors, gray and white. We can then use terminology of graphs, a specialization of the case of larger number of agents that requires using simplicial complexes, see Appendix 8.

Given a set V, an (undirected) *graph*[2] C over a

[2]Our non-traditional notations, C for a graph and $\mathcal{F}(C)$ for the set of edges of C, come from the general setting for n agents where graphs are replaced by simplicial *complexes*, and edges are replaced by *facets*. This is consistent with the notations of previous papers [25, 26].

set of *vertices* V is defined by a non-empty finite set of edges. Each *edge* is a set of two vertices. The set of vertices of C is noted $\mathcal{V}(C)$, and the set of edges $\mathcal{F}(C)$. A *chromatic graph* $C, \chi >$ consists of a graph C and a coloring map $\chi : \mathcal{V}(C) \to A$, such that for all $X \in \mathcal{F}(C)$, the two vertices of X have distinct colors.

Simplicial Models. Let be some countable set of values, and $AP = \{p_{a,x} \mid a \in A, x \in\}$ be the set of *atomic propositions*. Intuitively, $p_{a,x}$ is true if agent a holds the value x. We write AP_a for the atomic propositions of agent a.

A *simplicial model* $M =< C, \chi, \ell >$ consists of a chromatic graph $< C, \chi >$, and a labeling $\ell : \mathcal{V}(C) \to \mathcal{P}(AP)$ that associates with each vertex $v \in \mathcal{V}(C)$ a set of atomic propositions concerning agent $\chi(v)$, i.e., such that $\ell(v) \subseteq AP_{\chi(v)}$. Given an edge $X = \{v_0, v_1\} \in C$, we write $\ell(X) = \ell(v_0) \cup \ell(v_1)$.

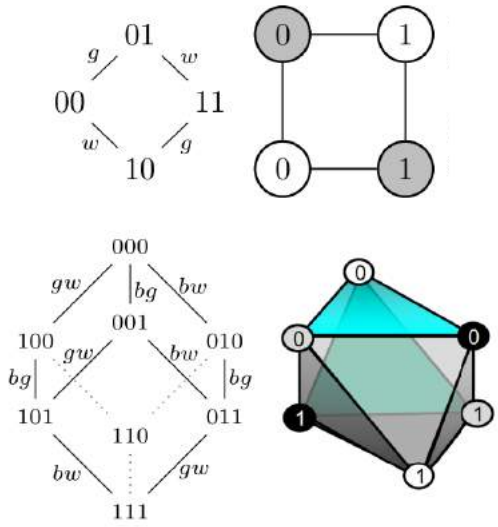
Binary Input Model. Each agent gets an input value from the set $\{0, 1\}$. Each agent knows its own input value, but doesn't know which value has been received by the other agents. The figure below from [25] shows the *binary input simplicial model* and its associated Kripke model for two agents, and for comparison also for three agents (although we will not use it in this paper). Notice that every possible combination of 0's and 1's is a possible world.

In the Kripke model, the agents are called b, g, w, and the labeling L of the possible worlds is represented as a sequence of values, e.g., 101, representing the values chosen by the agents b, g, w (in that order). In the 3-agents case, the labels of the dotted edges have been omitted to avoid overloading the picture, as well as other edges that can be deduced by transitivity.

In the simplicial model, agents are represented as colors (black, grey, and white).

The labeling ℓ is represented as a single value in a vertex, e.g., "1" in a grey vertex means that agent g has chosen value 1.

The possible worlds correspond to edges in the 2-agents case, and triangles in the 3-agents case. Note that the simplicial model depicted on the right, with three agents, is out of the scope of this paper.



Indeed, here we only work with two agents, so that the simplicial complex is nothing more than a graph.

Semantics. The definition below mimics the usual semantics of formulas in Kripke models, reformulated here in terms of simplicial models:

Definition 3.1 *We define the satisfaction relation* $M, X \models \phi$ *determining when a formula* φ *is* true *in some epistemic state* (M, X). *Let* $M = C, \chi, \ell >$ *be a simplicial model,* $X \in \mathcal{F}(C)$ *an edge of* C *and* $\varphi \in_K$*:*

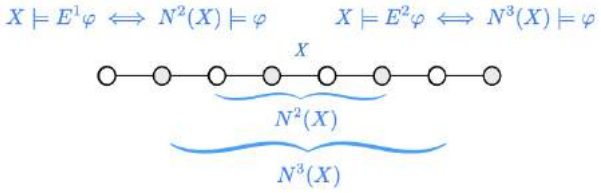
$$
\begin{array}{lll}
M, X \models p & \textit{iff} & p \in \ell(X), \\
M, X \models \neg\varphi & \textit{iff} & M, X \not\models \varphi, \\
M, X \models \varphi \wedge \psi & \textit{iff} & M, X \models \varphi \wedge M, X \models \psi, \\
M, X \models K_a\varphi & \textit{iff} & \forall Y \in \mathcal{F}(C), a \in \chi(X \cap Y), \\
 & & \Rightarrow M, Y \models \varphi.
\end{array}
$$

Group Knowledge. For a formula φ, we write $E\varphi = K_g\varphi \wedge K_w\varphi$ for the *group knowledge* of φ among the agents $\{g, w\}$. Let E^k denote k nested E operators. An *edge path* is a sequence of (not necessarily distinct) edges, such that each two consecutive edges in the sequence intersect in a vertex. For an edge X let $N^k(X)$ be the set of edges reachable from X by an edge path of at most k edges. Thus, $N^1(X) = \{X\}$, denoted by $N(X)$. Also, $N^2(X)$ is equal to X together

with the edges that intersect X, and in general $N^k(X) = N(N^{k-1}(X))$.

Lemma 3.1 *For a simplicial model M, edge X, and any $\varphi \in \mathcal{L}_{\mathcal{K}}$, we have that $M, X \models E^k\varphi$, iff $M, Y \models E\varphi$ for every $Y \in N^k(X)$.*

The figure below is an illustration of the cases $k = 1, 2$. It shows that $M, X \models E\varphi$ states that φ should hold in X and in its two neighboring edges, namely, in the edges that belong to $N^2(X)$, and analogously for $k = 2$.



Morphisms Between Models Let C and D be two graphs. A *simplicial map*[3] $f : C \to D$ maps the vertices of C to vertices of D, such that if X is an edge of C, $f(X)$ is an edge of D. A *chromatic simplicial map* between two chromatic graphs is a simplicial map that preserves colors, i.e. $\chi(f(v)) = \chi(v)$ for all v.

Lemma 3.2 *Let $f : C \to D$ be a simplicial map. If C' is a connected subgraph of C then $f(C')$ is a connected subgraph of D.*

A *morphism* of simplicial models $f : M \to M'$ is a chromatic simplicial map that preserves the labeling: $\ell'(f(v)) = \ell(v)$. The next theorem from [25] states that morphisms of simplicial models cannot "gain knowledge".

Lemma 3.3 (Knowledge Gain) *Consider simplicial models $M = C, \chi, \ell >$ and $M' = C', \chi', \ell' >$, and a morphism $f : M \to M'$. Let $X \in \mathcal{F}(C)$ be an edge of M, a an agent, and $\varphi \in \mathcal{L}_{CK}$ a positive formula, i.e. which does not contain negations except, possibly, in front of atomic propositions. Then, $M', f(X) \models \varphi$ implies $M, X \models \varphi$.*

[3] A simplicial map is usually called a homomorphism in graph theory.

3.2 Dynamic Epistemic Logic Basic Notions

Dynamic epistemic logic (DEL) is the study of modal logics of model change [4, 13]. A modal logic studied in DEL is obtained by using action models [3]. An action can be thought of as an announcement made by the environment (not necessarily public). An action model describes all the possible actions that might happen, as well as how they affect the different agents. The product-update operation takes an epistemic model M and an action model $\mathcal{A}$, and creates a new model $M[\mathcal{A}]$ that describes all the new possible worlds after an action from $\mathcal{A}$ has occurred in M. This classic version is described in Appendix 10, here we present the simplicial model version [25, 26].

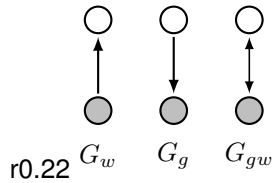
Action Models A *simplicial action model* $T, \chi, pre >$ consists of a chromatic graph $T, \chi >$, where the edges $\mathcal{F}(T)$ represent communicative *actions*, and pre assigns to each edge $X \in \mathcal{F}(T)$ a precondition formula $pre(X)$ in $_{\mathcal{K}}$.

Given two chromatic graphs C and T of dimension n, the Cartesian product $C \times T$ is the following chromatic graph. Its vertices are of the form (u, v) with $u \in \mathcal{V}(C)$ and $v \in \mathcal{V}(T)$ such that $\chi(u) = \chi(v)$; the color of (u, v) is $\chi((u, v)) = \chi(u) = \chi(v)$. Its edges are of the form $X \times Y = \{(u_0, v_0), \ldots, (u_k, v_k)\}$ where $X = \{u_0, \ldots, u_k\} \in C$, $Y = \{v_0, \ldots, v_k\} \in T$ and $\chi(u_i) = \chi(v_i)$.

Let $M = C, \chi, \ell >$ be a simplicial model, and $\mathcal{A} = T, \chi, pre >$ be a simplicial action model. The *product update simplicial model* $M[\mathcal{A}] = C[\mathcal{A}], \chi[\mathcal{A}], \ell[\mathcal{A}] >$ is a simplicial model whose underlying graph is a sub-graph of the Cartesian product $C \times T$, induced by all the edges of the form $X \times Y$ such that $pre(Y)$ holds in X, i.e., $M, X \models pre(Y)$. The valuation $\ell : \mathcal{V}(C[\mathcal{A}]) \to \mathcal{P}(AP)$ at a pair (u, v) is just as it was at u: $\ell[\mathcal{A}]((u, v)) = \ell(u)$.

4 An Action Model for Dynamic Networks

Distributed Computing Model For two processes, the model $\mathcal{G}$ where approximate agreement is solvable, but not consensus, is unique: It consists of the three digraphs G_g, G_w, G_{gw} on two



vertices, g, w, one with the antiparallel arrows, and the other two with an arrow from one vertex to the other, meaning that either both messages arrive, or only one. It is easy to see that in any submodel of $\mathcal{G}$ consensus can be solved (if it has at least one arrow). The model $\mathcal{G}$ is equivalent to several message passing and shared memory models that have been previously considered, see e.g. [1, 18].

Each agent has some input value, and they communicate r rounds. In each round an agent sends a message to the other agent, and the message arrives by the end of the round, or not at all. Which messages arrive depends on which graph from $\mathcal{G}$ is selected (it is convenient to assume that each vertex of a graph of $\mathcal{G}$ has a loop; an agent learns its own messages. But we do not draw the loops in the figures). In every round, any of the three graphs in $\mathcal{G}$ can be selected.

Input Model An *input model*, $\mathcal{I}$, represents all the possible input values. To illustrate the ideas, we use the input simplicial model of Section 3.1 where two agents $A = \{g, w\}$ each has an input value from the set $\{0, 1\}$.

Actions An *action*[4] $t \in \hat{T}$ is given by b_0, b_1, c, where c is a sequence of r digraphs from $\mathcal{G}$ and $b_0, b_1 \in \{0, 1\}$ are binary values, representing the inputs of both processes. Such an action is written $c^{b_0 b_1}$. The precondition $pre(c^{b_0 b_1})$ of this action is a formula expressing the fact that the inputs of the agents g, w are respectively b_0, b_1. Formally, if $input_a^x$ denotes the atomic proposition expressing that agent a has input value x, then $pre(c^{b_0 b_1}) = input_g^{b_0} \wedge input_w^{b_1}$.

[4]For the moment, an action is simply an element of the set $\hat{T}$; we will later define a simplicial action model $< T, \chi, pre >$ whose set of edges is in bijection with $\hat{T}$.

Distributed Algorithm An action $t = c^{b_0 b_1}$ includes all the information about the inputs and message deliveries of an execution, but not about the actual algorithm being executed by the agents. Namely, an algorithm specifies the contents of messages sent in each round, and the state transition of each agent. Each agent starts in an initial state, determined by its input value. At the end of a round, each agent changes to a new state following a deterministic transition function, based on its current state, and on the message received. The new state of the agent determines the message sent in the new round. The local state of agent a at the end of execution t is denoted $view_a(t)$.

For example, consider the execution $c = G_w, G_{gw}, G_g$, meaning that in the first round only w receives a message; in the second round both processes receive a message; and in the last round only g receives a message. Picking $b_0 = 0$ and $b_1 = 1$ gives us the action c^{01}, where g starts with value 0 and w starts with value 1. This is independent of the algorithm. In the algorithm of Figure 1, local states are simply the value of variable ℓ, so this action leads to the local states $view_g(c^{01}) = 4/27$ and $view_w(c^{01}) = 3/27$. In the one-bit algorithm of Figure 2, the local states are the views, so this action leads to the local states $view_g(c^{01}) = \bot 01$ and $view_w(c^{01}) = 00\bot$.

Indistinguishability The indistinguishability relation $t \sim_a t'$ is a function of both the actions t, t' and the algorithm. It is defined as $t \sim_a t'$ iff $view_a(t) = view_a(t')$.

Action Model For a given algorithm, we can reformulate the indistinguishability relation as a simplicial action model $< T, \chi, pre >$ where $< T, \chi >$ is a chromatic graph whose vertices are $\mathcal{V}(T) = \{a, view_a(c^{b_0 b_1}) > |\ a \in A, c^{b_0 b_1} \in \hat{T}\}$ and whose edges are of the form, for each action $c^{b_0 b_1} \in \hat{T}$, $X = \{< b, view_w(c^{b_0 b_1}) >, < g, view_g(c^{b_0 b_1}) >\}$. The precondition of such a facet is $pre(X) = input_w^{b_0} \wedge input_g^{b_1}$.

We write $\mathcal{A}^r$ for the simplicial action model of the r-round averaging algorithm of Figure 1, and $\mathcal{OB}^r$ for the action model of the r-round one-bit algorithm of Figure 2. We also write $\mathcal{DN}^r$ for

a generic r-round dynamic network action model over an unspecified algorithm. The action models $\mathcal{OB}^1$ and $\mathcal{OB}^2$ are depicted in Figure 4 (as we mention below, $\mathcal{DN}^r$ and $\mathcal{I}[\mathcal{DN}^r]$ are always isomorphic). For $r = 1$, each of the four input edges has been subdivided into 3 "smaller" edges: one for each possible graph of $\mathcal{G}$. The colors white, grey, of the vertices correspond respectively to agents w, g. The view of each vertex is written next to it. The precondition of the three top edges is true exactly in the top edge of the input model $\mathcal{I}$ on the left, and similarly for the other sides of the square. For $r = 2$, the situation is similar except that each edge of $\mathcal{I}$ is subdivided into $3^2 = 9$ edges.

Perhaps surprisingly, the product update models $\mathcal{I}[\mathcal{A}^r]$ and $\mathcal{I}[\mathcal{OB}^r]$ are isomorphic, meaning that both algorithms from Figures 1 and 2 acquire the same knowledge. They both subdivide each edge of the input complex into exactly 3^r edges, thus they are both also isomorphic to the well-known *full-information* protocol complex.

Theorem 4.1 *Let $\mathcal{I}$ be the binary input model for two processes. For any number of rounds r, the product update models $\mathcal{I}[\mathcal{A}^r]$ and $\mathcal{I}[\mathcal{OB}^r]$ are isomorphic.*

Properties about the Action Model One last thing to notice about this construction is that, when we compute the product update model $\mathcal{I}[\mathcal{DN}^r]$, we obtain a simplicial model whose underlying graph is the same as the one of $\mathcal{DN}^r$. So, starting from the input model $\mathcal{I}$, the effect of applying $\mathcal{DN}^r$ is to subdivide each edge of the input (the same thing happens for any other input model). Remarkably, the topology of the input graph is preserved. And in fact, there is a rate of subdivision speed, determined by the constant $1/3$. These are the two basic properties that we will need in the analysis of approximate agreement (known in several specific contexts e.g. [12, 18]). Remarkably, they hold for any algorithm, not only full information algorithms.

Theorem 4.2 *For any algorithm for two agents, if the input model $\mathcal{I}$ is connected, then the product update model $\mathcal{I}[\mathcal{DN}^r]$ is connected. Furthermore, each edge is subdivided into at most 3^r edges.*

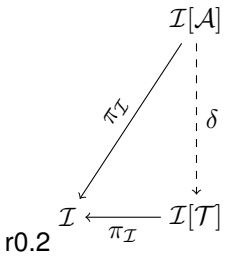
We have seen that the two properties mentioned in this theorem hold for a full information algorithm. It is not hard to see that they hold for any algorithm, since the indistinguishability relation of a non-full information algorithm is a coarsening of the full information indistinguishability algorithm.

5 Tasks

The action models of the previous section, $\mathcal{A}^r$ and $\mathcal{OB}^r$, describe how the knowledge of the processes evolve when we execute a specific algorithm. Both of those algorithms are solving the same distributed *task*, namely, approximate agreement. In this section, we introduce the notion of *task*, which is an abstract specification of the goal that we are trying to solve, independently of the particular algorithm used to solve it. Informally, a task specifies for each possible input configuration, what are the possible output values that the agents may decide. This is once again formalized using the notion of action model; however, in the case of tasks, the actions do not correspond to communicative events, but to decisions taken by the processes. Tasks have been studied since early on in distributed computability [5]. The DEL semantics that we use here was first introduced in [25].

Consider a simplicial model $\mathcal{I} = I, \chi, \ell >$ called the *initial simplicial model*. Each edge of I, with its labeling ℓ, represents a possible initial configuration. We fix $\mathcal{I}$, the binary inputs model of Section 3.1.

A *task* for $\mathcal{I}$ is a simplicial action model $\mathcal{T} =< T, \chi, pre >$ for agents A, where each edge is of the form $X = \{< w, d_w >, < g, d_g >\}$. The values d_w, d_g are taken from an arbitrary domain of *output values*. Each such X has a precondition that is true in one or more facets of $\mathcal{I}$, interpreted as "if the input configuration is a facet in which $pre(X)$ holds, and every agent $a \in A$ decides on the value d_a, then this is a valid execution".



r0.2

5.1 Semantics of Task Solvability

Given the simplicial input model $\mathcal{I}$ and a communication model $\mathcal{A}$ such as $\mathcal{DN}^r$, we get the *simplicial protocol model* $\mathcal{I}[\mathcal{A}]$, that represents the knowledge gained by the agents after executing $\mathcal{A}$. To solve a task $\mathcal{T}$, each agent, based on its own knowledge, should produce an output value, such that the edge with the output values corresponds to an edge of $\mathcal{T}$, respecting the preconditions of the task.

The following gives a formal epistemic logic semantics to task solvability. Recall that a morphism δ of simplicial models is a chromatic simplicial map that preserves the labeling: $\ell'(\delta(v)) = \ell(v)$. Also recall that the product update model $\mathcal{I}[\mathcal{A}]$ is a sub-graph of the Cartesian product $\mathcal{I} \times \mathcal{A}$, whose vertices are of the form (i, ac) with i a vertex of $\mathcal{I}$ and ac a vertex of $\mathcal{A}$. We write $\pi_{\mathcal{I}}$ for the first projection on $\mathcal{I}$, which is a morphism of simplicial models.

Definition 5.1 *A task $\mathcal{T}$ is* solvable *using the algorithm $\mathcal{A}$ if there exists a morphism $\delta : \mathcal{I}[\mathcal{A}] \to \mathcal{I}[\mathcal{T}]$ such that $\pi_I \circ \delta = \pi_I$, i.e., the diagram of simplicial complexes below commutes:*

The justification for this definition is the following. An edge X in $\mathcal{I}[\mathcal{A}]$ corresponds to a pair (i, ac), where i is an edge of $\mathcal{I}$ representing input value assignments to the agents, and ac is an edge of $\mathcal{A}$ codifying the communication exchanges that took place. The morphism δ takes X to an edge $\delta(X) = (i, dec)$ of $\mathcal{I}[\mathcal{T}]$, where dec is the edge of $\mathcal{T}$ defining the set of decision values that the agents will choose in X. Moreover, $pre(dec)$ holds in i, meaning that dec corresponds to valid decision values for input i. The commutativity of the diagram expresses the fact that both X and $\delta(X)$ correspond to the same input assignment i. Now consider a single vertex $v \in X$ with $\chi(v) = a \in A$. Then, agent a decides its value solely according to its knowledge in $\mathcal{I}[\mathcal{A}]$: if another edge X' contains v, then $\delta(v) \in \delta(X) \cap \delta(X')$, meaning that a has to decide on the same value in both edges.

Two observations about the diagram. First, by Lemma 3.3, we know that the knowledge of each agent can only decrease (or stay constant) along the δ arrow. So, any (positive) formula which is known in $\mathcal{I}[\mathcal{T}]$ should already be known in $\mathcal{I}[\mathcal{A}]$. In other words, the goal of the agents is to improve knowledge through communication, by going from $\mathcal{I}$ to $\mathcal{I}[\mathcal{A}]$, in order to match the knowledge required by $\mathcal{I}[\mathcal{T}]$. Second, the possibility of solving a task depends on the existence of a certain simplicial map from the graph of $\mathcal{I}[\mathcal{A}]$ to the graph of $\mathcal{I}[\mathcal{T}]$. We have already seen the appearance of a topological property in Theorem 4.2. Here again topology appears: a simplicial map is the discrete equivalent of a continuous map, thus connectivity is preserved by simplicial maps (Lemma 3.2); hence the topological nature of task solvability.

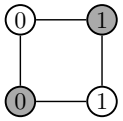
5.2 Approximate Agreement

Recall that in the approximate agreement problem agents are required to decide on values which are close to each other. We have seen in Section 2 that no matter how close to each other one requires the agents to decide, this task is solvable in the $\mathcal{DN}^r$ model, taking a sufficiently large r. Many versions of this task have been considered e.g. [10]. We present here the version of [25], for two agents g and w, which is at the core of previously considered situations.

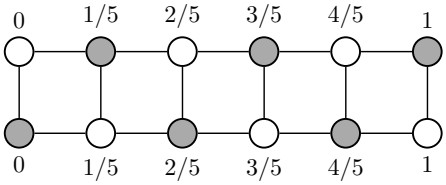
The input complex is the binary input complex for two agents of Section 3.1: so, every possible combination of 0 and 1 can be assigned to the two agents. Their goal will be to output real values in the interval $[0, 1]$, such that: (i) if their input is the same, they both decide the same output, and (ii) if their input is different, they both decide on values d_g and d_w such that $|d_g - d_w| \leq \varepsilon$, for some fixed parameter $\varepsilon \in [0, 1]$. A discrete version of this task is N-*approximate agreement*, where the output values are only allowed to be of the form

k/N for $0 \leq k \leq N$. The two decision values should be within $1/N$ of each other: $|d_g - d_w| \leq 1/N$. And to simplify the presentation, let us assume N is odd, without loss of generality, in the sense that to solve ε-agreement, one has to select N large enough, so that $1/N \leq \varepsilon$.

Let $\mathcal{I} =< I, \chi, \ell >$ be the input simplicial model for two agents with binary inputs, and $\mathcal{T} =< T, \chi, pre >$ be the following action model. The set of vertices of T is $\mathcal{V}(T) = \{(a, k/N) \mid a \in A \text{ and } 0 \leq k \leq N\}$. The facets of T are edges $X_{k,k'} = \{(g, k/N), (w, k'/N)\}$ with $|k - k'| \leq 1$. The color of a vertex is $\chi(a, k/N) = a$. The precondition $pre(X_{0,0})$ is true in the worlds 00, 01 and 10 of $\mathcal{I}$; the precondition $pre(X_{N,N})$ is true in the worlds 11, 01 and 10; and all the other preconditions $pre(X_{k,k'})$ are true in the worlds 01 and 10. In the figure below are the input model $\mathcal{I}$ (left) and the action model $\mathcal{T}$ (right), for $N = 5$:

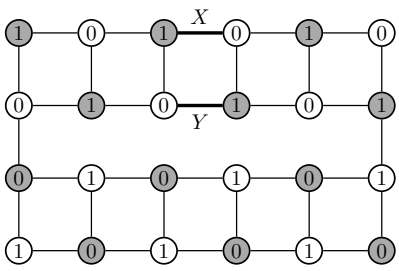


$\mathcal{I}$



$\mathcal{T}$

The product update simplicial model $\mathcal{I}[\mathcal{T}]$ is depicted in the next figure, for $N = 5$.



$\mathcal{I}[\mathcal{T}]$ for $N = 5$

The numbers depicted in the nodes are the atomic propositions describing the input values from $\mathcal{I}$. The decision values (of the form $k/5$) are implicit, the first column of nodes corresponds to the decision value 0, the second column is decision value $1/5$, and so on.

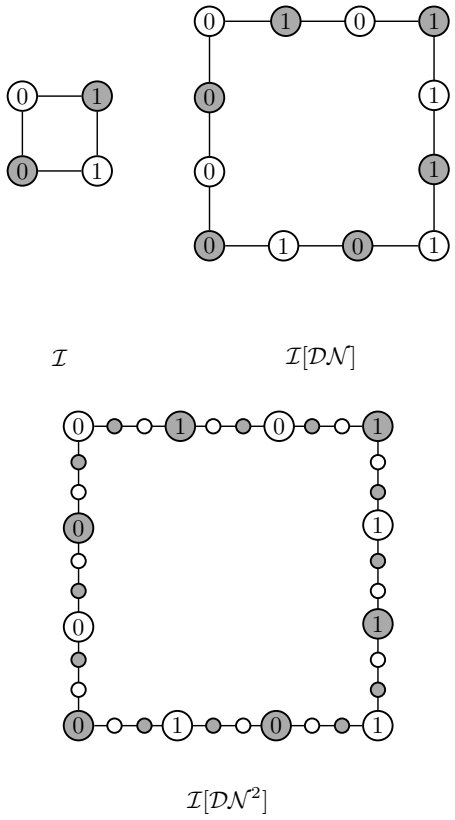
6 Approximate Agreement Solvability

The world X on the figure is the one with the most knowledge in the following sense. We write ϕ_{01} for the formula expressing that the two inputs are different. Recall (Section 3.1) that $E\phi = K_g\phi \wedge K_w\phi$ for the group knowledge of ϕ among the agents $\{g, w\}$. Then, we have $\mathcal{I}[\mathcal{T}], X \models E^3\phi_{01}$. On the other hand, the world Y has less knowledge: we have $\mathcal{I}[\mathcal{T}], Y \models E^2\phi_{01}$, but $\mathcal{I}[\mathcal{T}], Y \not\models E^3\phi_{01}$.

Lemma 6.1 *In the simplicial model $\mathcal{I}[\mathcal{T}]$ for the N-approximate agreement task, there are two worlds X, Y which satisfy the formula $E^k\phi_{01}$, for $k = \lceil N/2 \rceil$.*

Proof. We choose the worlds in the "middle" of the model $\mathcal{I}[\mathcal{T}]$, as shown in the picture above. More formally, recall that the vertices of $\mathcal{I}[\mathcal{T}]$ are defined as tuples (a, i, d) where a is an agent, i its input value and d its decision value. The world X is defined as the edge $\{(g, 1, \lfloor N/2 \rfloor/N), (w, 0, (\lfloor N/2 \rfloor + 1)/N)\}$, and $Y = \{(w, 1, \lfloor N/2 \rfloor/N), (g, 0, (\lfloor N/2 \rfloor + 1)/N)\}$. Checking that the formula is satisfied in these worlds simply consists in computing the length of the shortest path to one of the 00 or 11 edges on the sides (Lemma 3.1). ■

Solvability of Approximate Agreement We now study the solvability of approximate agreement in the r-round dynamic graph model $\mathcal{DN}^r$. Recall that each round subdivides each edge into three edges (Theorem 4.2). The picture below shows the input model $\mathcal{I}$, the model $\mathcal{I}[\mathcal{DN}]$ after one round, and the model $\mathcal{I}[\mathcal{DN}^2]$ after two rounds:



Lemma 6.2 *In the r-round model $\mathcal{I}[\mathcal{DN}^r]$, there is no world X such that $\mathcal{I}[\mathcal{DN}^r], X \models E^k\phi_{01}$, for $k = \lceil 3^r/2 \rceil$.*

Proof. After r rounds each of the four edges of the input model $\mathcal{I}$ has been subdivided into 3^r edges. Thus, every world is at a distance at most $k-1$ from the nearest world with inputs 00 or 11. ■

Putting the two lemmas together, we get the following result:

Theorem 6.1 *The N-approximate agreement task is not solvable in the r-round model $\mathcal{I}[\mathcal{DN}^r]$, when $N \geq 3^r + 1$.*

Proof. Assume for contradiction that the task is solvable. Then, we would have a map $\delta : \mathcal{I}[\mathcal{DN}^r] \to \mathcal{I}[\mathcal{T}]$. Our goal is to find a contradiction using Lemma 3.3. To achieve this, we should find a formula ϕ and a world Z of $\mathcal{I}[\mathcal{DN}^r]$, such that ϕ is false in Z but true in $\delta(Z)$. We choose the formula $\phi := E^k\phi_{01}$ for $k = \lceil N/2 \rceil$. Since $N \geq 3^r+1$ implies $\lceil N/2 \rceil \geq \lceil 3^r/2 \rceil$, we know by Lemma 6.2 that this formula is false in every world Z of $\mathcal{I}[\mathcal{DN}^r]$. All that remains to do is prove that there exists a world of $\mathcal{I}[\mathcal{T}]$, *which is in the image of* δ, and where the formula ϕ is true.

Since $\mathcal{I}[\mathcal{DN}^r]$ is connected, its image $\delta(\mathcal{I}[\mathcal{DN}^r])$ is connected (by Lemma 3.2). Moreover, the world 00 and the world 11 of $\mathcal{I}[\mathcal{T}]$ must both be in the image of δ, because of the commutative diagram of Definition 5.1. By connectedness, at least one of the middle worlds X, Y of Lemma 6.1 must belong to the image of δ. By Lemma 6.1, this world satisfies ϕ, which concludes the proof. ■

Conversely, we have seen in Section 2 that N-approximate agreement is solvable in r rounds whenever $N = 3^r$. The proof of the above theorem sheds light on the required knowledge to solve approximate agreement: while consensus is about reaching common knowledge, approximate agreement is about reaching some finite level of nested knowledge.

7 Approximate Agreement Algorithms

Here we provide additional details about the correctness of the algorithms described in Section 2. Recall that $N = 3^k$, agents start with values in $0, 1$, and they have to decide values at most $1/N$ apart, unless their inputs are equal, in which case their outputs should be equal to their inputs. First we show that the algorithm in Figure 1 solves N-approximate agreement in k rounds.

Theorem 7.1 *The algorithm Averaging Approximate Agreement is correct.*

Proof. Let me_p be the initial value of agent p. For $k = 1$, suppose w.l.g. that the message from g arrives, then the final value of agent w is $d_w = (me_w + 2me_g)/3$. If both messages arrives then $d_w = (me_g+2me_w)/3$ so $|d_w - d_g| = |(2me_g - me_g + me_w - 2me_w)/3| = |(me_g - me_w)/3| \leq 1/3$. If just one message arrives then $d_g = me_g$, so $|d_w - d_g| = |(me_w + 2me_g - 3me_g)/3| = |(me_w - me_g)/3| \leq 1/3$. Let d_p^k be the final value of agent p after k rounds, and suppose that $|d_w^k - d_g^k| \leq 1/3^k$ for some k, then, if message from g arrives in round $k+1$ (w.l.g.), $d_w^{k+1} = (d_w^k + 2d_g^k)/3$. The rest of the proof is

analogous to induction base, analysing both cases whether message from w gets lost or not. ■

To prove the correctness of the One-Bit Messages algorithm of Figure 2, we rely on Theorem 4.1. So let us prove it first:

Theorem 4.1. *Let $\mathcal{I}$ be the binary input model for two processes. For any number of rounds r, the product update models $\mathcal{I}[\mathcal{A}^r]$ and $\mathcal{I}[\mathcal{OB}^r]$ are isomorphic.* **Proof.** Recall that $\mathcal{I}[\mathcal{A}^r]$ is the product update model for r-rounds of the Averaging algorithm in Figure 1, and $\mathcal{I}[\mathcal{OB}^r]$ is the product update model for r-rounds of the One-Bit algorithm of Figure 2.

It is well known that $\mathcal{I}[\mathcal{A}^r]$ is isomorphic to the full-information protocol complex, which subdivides each edge of the input into 3^k edges, preserving the topology. Thus, we show that $\mathcal{I}[\mathcal{OB}^r]$ does the same thing.

We proceed by induction on r. For the one-round algorithm, $\mathcal{I}[\mathcal{OB}^1]$ is depicted in Figure 4, and indeed each edge of the input complex has been subdivided into three.

Let us now assume that $\mathcal{I}[\mathcal{OB}^r]$ is indeed isomorphic to $\mathcal{I}[\mathcal{A}^r]$, i.e., that each edge of $\mathcal{I}$ has been subdivided in 3^k small edges.

Consider an arbitrary vertex of $\mathcal{I}[\mathcal{OB}^r]$ (a grey vertex w.l.o.g.), of the form $< g, view_g(t) >$ for some action $t = c^{b_0,b_1}$. Let us write $v = view_g(t)$ for the view of this vertex.

Moreover, let us assume that the two neighbors of this vertex, with views u and w, are about to send two distinct messages, 0 and 1 respectively. (One can check that this property is true everywhere in $\mathcal{I}[\mathcal{OB}^1]$: every vertex has neighbors that will send different messages in the next round.)

After one more round of the one-bit algorithm, we obtain six edges in $\mathcal{I}[\mathcal{OB}^{r+1}]$ as depicted below (we assume w.l.o.g. that the grey vertex sends the message "0"; the same picture can be drawn with a 1 instead). Moreover, we can check that the five vertices in the middle at round $r+1$ still have the inductive property that both neighbors will send a different message at the next round.

This local reasoning can be done around each node of $\mathcal{I}[\mathcal{OB}^r]$. Thus, every edge is subdivided into 3,which concludes our proof. ■

Now we can show that the algorithm in Figure 2 solves N-approximate agreement in k rounds.

Theorem 7.2 *The algorithm One-Bit Messages Approximate Agreement is correct.*

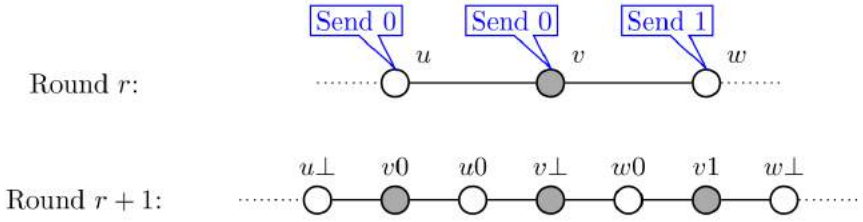
Proof. Since $\mathcal{I}[\mathcal{OB}^r]$ is isomorphic to $\mathcal{I}[\mathcal{A}^r]$, and we have shown in Theorem 7.1 that $\mathcal{I}[\mathcal{A}^r]$ solves 3^r-approximate agreement, we know that the One-Bit algorithm can solve approximate agreement as long as we choose the right decision function δ.

This is precisely what the algorithm from Figure 3 does: $\delta(\ell, \text{view})$ simulates the Average Approximate Agreement algorithm via the isomorphism exhibited in Theorem 4.1. A detailed proof of correctness can be found in [12]. ■

8 Simplicial Models and Epistemic Logic

Here we include generalized notions for any number of agents from section 3.

Generalizing graphs to complexes Given a set V, a *simplicial complex* C is a family of non-empty finite subsets of V such that for all $X \in C$, $Y \subseteq X$ implies $Y \in C$. We say Y is a *face* of X. Elements of V (identified with singletons) are called *vertices*. Elements of C are *simplexes*, and those which are maximal w.r.t. inclusion are *facets*. The set of vertices of C is noted $\mathcal{V}(C)$, and the set of facets $\mathcal{F}(C)$. The *dimension* of a simplex $X \in C$ is $|X| - 1$, and a simplex of dimension n is called an *n-simplex*. A simplicial complex C is *pure* if all its facets are of the same dimension, n. In this case, we say C is of dimension n. A graph without isolated vertices is a pure sumplicial complex of dimension 1. Given the set A of agents (that we will represent as colors), a *chromatic simplicial complex* $< C, \chi >$ consists of a simplicial complex C and a coloring map $\chi : \mathcal{V}(C) \to A$, such that for all $X \in C$, all the vertices of X have distinct colors.



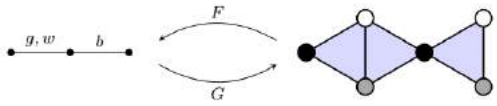
Simplicial Maps Let C and D be two simplicial complexes. A *simplicial map* $f : C \to D$ maps the vertices of C to vertices of D, such that if X is a simplex of C, $f(X)$ is a simplex of D. A *chromatic simplicial map* between two chromatic simplicial complexes is a simplicial map that preserves colors. Let $\mathcal{S}_A$ be the category of pure chromatic simplicial complexes on A, with chromatic simplicial maps for morphisms.

Equivalence with Epistemic Logic A *Kripke frame* $M =< S, \sim >$ over a set A of agents consists of a set of states S and a family of equivalence relations on S, written $\sim_a$ for every $a \in A$. Two states $u, v \in S$ such that $u \sim_a v$ are said to be *indistinguishable* by a. A Kripke frame is *proper* if any two states can be distinguished by at least one agent. Notice that being proper means that the intersection of all equivalence relations $\sim_a$ is the identity; this may reveal interesting parallels with distributed knowledge (a formula that is true in all states in the intersection relation), see e.g. [24]. Let $M =< S, \sim >$ and $N =< T, \sim' >$ be two Kripke frames. A *morphism* from M to N is a function f from S to T such that for all $u, v \in S$, for all $a \in A$, $u \sim_a v$ implies $f(u) \sim'_a f(v)$. We write $\mathcal{K}_A$ for the category of proper Kripke frames, with morphisms of Kripke frames as arrows.

The following theorem states that we can canonically associate a proper Kripke frame with a pure chromatic simplicial complex, and vice versa. In fact, this correspondence extends to morphisms, and thus we have an equivalence of categories, meaning that the two structures contain the same information.

Theorem 8.1 ([25]) *$\mathcal{S}_A$ and $\mathcal{K}_A$ are equivalent categories.*

Example 8.1 ([25]) *The picture below shows a Kripke frame (left) and its associated chromatic simplicial complex (right). The three agents, named b, g, w, are represented as colors black, grey and white on the vertices of the simplicial complex. The three worlds of the Kripke frame correspond to the three triangles (i.e., 2-dimensional simplexes) of the simplicial complex. The two worlds indistinguishable by agent b, are glued along their black vertex; the two worlds indistinguishable by agents g and w are glued along the grey-and-white edge.*



Simplicial Models Let be some countable set of values, A be a finite set of agents and $AP = \{p_{a,x} | a \in A, x \in \mathcal{V}\}$ be the set of *atomic propositions*. Intuitively, $p_{a,x}$ is true if agent a holds the value x. We write AP_a for the atomic propositions concerning agent a. A *simplicial model* $M =< C, \chi, \ell >$ consists of a chromatic simplicial complex $< C, \chi >$, and a labeling $\ell : \mathcal{V}(C) \to \mathcal{P}(AP)$ that associates with each vertex $v \in \mathcal{V}(C)$ a set of atomic propositions concerning agent $\chi(v)$, i.e., such that $\ell(v) \subseteq AP_{\chi(v)}$. Given a simplex $X \in C$, we write $\ell(X) = \bigcup_{i=0}^{n} \ell(v_i)$. A *morphism* of simplicial models $f : M \to M'$ is a chromatic simplicial map that preserves the labeling: $\ell'(f(v)) = \ell(v)$. We denote by $\mathcal{SM}_{A,AP}$ the category of simplicial models over the set of agents A and atomic propositions AP.

Equivalence with Epistemic Logic A *Kripke model* $M = < S, \sim, L >$ consists of a Kripke frame $< S, \sim >$ and a function $L : S \to \mathcal{P}(AP)$. Intuitively,

$L(s)$ is the set of atomic propositions that are true in the state s. A Kripke model is *proper* if the underlying Kripke frame is proper. A Kripke model is *local* if for every agent $a \in A$, $s \sim_a s'$ implies $L(s) \cap AP_a = L(s') \cap AP_a$, i.e., an agent always knows its own values. Let $M =< S, \sim, L >$ and $M' =< S', \sim', L' >$ be two Kripke models on the same set AP. A *morphism of Kripke models* $f : M \to M'$ is a morphism of the underlying Kripke frames such that $L'(f(s)) = L(s)$ for every state s in S. We write $\mathcal{KM}_{A,AP}$ for the category of local proper Kripke models.

[[25]] Given a simplicial model M and a facet X, $M, X \models \varphi$ iff $F(M), X \models_{\mathcal{K}} \varphi$. Conversely, given a local proper Kripke model N and state s, $N, s \models_{\mathcal{K}} \varphi$ iff $G(N), G(s) \models \varphi$, where $G(s)$ is the facet $\{v_0^s, \ldots, v_n^s\}$ of $G(N)$.

We can now extend Theorem 8.1 to an equivalence between simplicial models and Kripke models.

Theorem 8.2 ([25]) *$\mathcal{SM}_{A,AP}$ and $\mathcal{KM}_{A,AP}$ are equivalent categories.*

9 Knowledge in Simplicial Models

Here we include additional details about group knowledge and knowledge gain from subsection 3.1.

Lemma 3.1 For a simplicial model M and edge X, we have that $M, X \models E^k\phi$, iff $M, Y \models E\varphi$ for every $Y \in N^k(X)$.

Proof. For $k = 1$ the proof is immediate.
Let k such that $M, X \models E^k\phi$ iff $M, Y \models E\phi$ for every $Y \in N^k(X)$. Then $M, X \models E^k(E\phi)$ iff $M, Z \models E^2\phi$ for every $Z \in N^k(X)$. Finally $M, Y \models E\phi$ for every $Y \in N^2(Z)$ such that $Z \in N^k(X)$, i.e. $Y \in N^{k+1}(X)$ (since $N^{k+1}(X) = N(N^k(X))$). ■

Lemma 3.3 Consider simplicial models $M =< C, \chi, \ell >$ and $M' =< C', \chi', \ell' >$, and a morphism $f : M \to M'$. Let $X \in \mathcal{F}(C)$ be an edge of M, a an agent, and $\varphi \in \mathcal{L}_{CK}$ a positive formula, i.e. which does not contain negations except, possibly, in front of atomic propositions. Then, $M', f(X) \models \varphi$ implies $M, X \models \varphi$.

Proof. Suppose that ϕ is atomic, then $M', f(X) \models \phi$ iff $\phi \in l(f(X)) = l(X)$, so $M, X \models \phi$.
If $\phi = \neg p$ for some atomic p then $M', f(X) \not\models p$, so $p \notin l(f(X)) = l(X)$, therefore $M, X \not\models p$.
If $\phi = \psi \wedge \theta$ for some formulas ψ and θ as depicted above, then $M', f(X) \models \psi$ and $M', f(X) \models \theta$. Therefore $M, X \models \psi \wedge \theta$.
If $\phi = K_a(\psi)$ with ψ being a formula like depicted above, let $Y \in \mathcal{F}(M)$ such that $a \in \chi(Y \cap X)$, then $a \in \chi(f(Y) \cap f(X))$ so $M', f(Y) \models \psi$ and therefore $M, Y \models \psi$, implying that $M, X \models \phi$.
Finally, every possitive formula can be seen as combinations of formulas like depicted above but linked with $\wedge$. Hence, if $\phi = \bigwedge_{i\in} \psi_i$ then for all $i \in$, $M', f(X) \models \psi_i$ so that $M, X \models \psi_i$. Finally, $M, X \models \bigwedge_{i\in} \psi_i$. ■

10 Dynamic Epistemic Logic

Here we presents the classic version of DEL.

An *action model* is a structure $< T, \sim, pre >$,where T is a domain of *action points*, such that for each $a \in A$, $\sim_a$ is an equivalence relation on T, and $pre : T \to_{\mathcal{K}}$ is a function that assigns a *precondition* formula $pre(t)$ to each $t \in T$. Let $M =< S, \sim, L >$ be a Kripke model and $\mathcal{A} =< T, \sim, pre >$ be an action model. The *product update model* is $M[\mathcal{A}] =< S[\mathcal{A}], \sim^{[\mathcal{A}]}, L[\mathcal{A}] >$, where each world of $S[\mathcal{A}]$ is a pair (s, t) with $s \in S$, $t \in T$ such that $pre(t)$ holds in s. Then, $(s, t) \sim_a^{[\mathcal{A}]} (s', t')$ whenever it holds that $s \sim_a s'$ and $t \sim_a t'$. The valuation $L[\mathcal{A}]$ at a pair (s, t) is just as it was at s, i.e., $L[\mathcal{A}]((s, t)) = L(s)$. For an initial Kripke model M, the effect of action model $\mathcal{A}$ is a Kripke model $M[\mathcal{A}]$. Notice that if M is a local proper Kripke model and $\mathcal{A} =< T, \sim, pre >$ is a proper action model, then $M[\mathcal{A}]$ is proper and local.

Equivalence with Simplicial Action Models Recall from Theorem 8.2 the two functors F and G that define an equivalence of categories between simplicial models and Kripke models. We have a similar correspondence between action models and simplicial action models. On the underlying Kripke frame and simplicial complex they are the same as before; and the precondition of an action

point is just copied to the corresponding facet. The following proposition says that the "classic" product update agrees with the "fully simplicial" one.

Consider a simplicial model M and simplicial action model $\mathcal{A}$, and their corresponding Kripke model $F(M)$ and action model $F(\mathcal{A})$. Then, the Kripke models $F(M[\mathcal{A}])$ and $F(M)[F(\mathcal{A})]$ are isomorphic. The same is true for G, starting with a Kripke model M and action model $\mathcal{A}$.

Proof. Particular case of theorem 8.2, since $F(M[\mathcal{A}])$ is the image of $F(M)[F(\mathcal{A})]$ under the functor who states equivalence. ■

11 Dynamic Networks and Tasks

Here we include additional details about Section 4 and 5.

Theorem 4.2 For any algorithm for two agents, the product update model $M[\mathcal{DN}^r]$ is a graph which is connected (assuming the input model is connected). Furthermore, each edge is subdivided into at most 3^r edges.

Proof.

Since underlying simplicial complex of $M[\mathcal{DN}^r]$ is the same as the one of $G(\mathcal{DN}^r)$, we just need to prove that $G(\mathcal{DN}^r)$ is connected.

For $r = 1$, let $\mathcal{I}$ be the input model, then the function who carries any edge X of $\mathcal{I}$ into the set $\{X \in \mathcal{F}(G(\mathcal{DN}^r))|M, \{(b, b_0), (g, b_1)\} \models pre(X)\}$ is bijective. Moreover $f[\mathcal{I}] = \mathcal{F}(G(\mathcal{DN}))$ and since any edge $X \in \mathcal{F}(\mathcal{I})$ is such that $|f(X)| = 3$, X splits into 3 edges in $G(\mathcal{DN})$.

Now consider arbitrary edges $X, Y \in \mathcal{F}(\mathcal{I})$ and let $X_i, Y_j \in \mathcal{F}(G(\mathcal{DN}))$ subdivisions of X, Y respectively. Since $\mathcal{I}$ is connected, exists $(X, Z_1, \ldots, Z_n, Y)$ a sequence of connected edges in $\mathcal{I}$. Hence, in $G(\mathcal{DN})$ we have another edge path who links X_i with any subdivision of Z_1, which is linked with any subdivision of Z_2 and so on, until reach any subdivision of Z_n, and finally Y_j.

For r such that $G(\mathcal{DN}^r)$ is connected, we can consider $G(\mathcal{DN}^r)$ as an input model, and since $G(\mathcal{DN}^{r+1})$ is also the underlying graph of $(M[\mathcal{DN}^r])[\mathcal{DN}]$, the rest of the proof is analogous to induction base. Finally, since each edge is already subdivided into 3^r edges in $G(\mathcal{DN}^r)$, they are subdivided into $3(3^r) = 3^{r+1}$ edges in $G(\mathcal{DN}^{r+1})$. ■

12 Conclusions

We have considered a basic notion of approximate agreement, and a computational model for two agents, where the task is solvable for any desired level of precision. We first presented two algorithms, which solve the task for a given precision level, using the same number of communication rounds. With this concrete setting in mind, we have proceeded to give a formal semantics based on dynamic epistemic logic, both to our computational model, and to the approximate agreement task. We derived a lower bound result showing that these algorithms are optimal in the number of rounds. The lower bound is due to the impossibility of the agents gaining global knowledge about their inputs faster, and a consequence of the connectivity of the computational model's epistemic states. Although much of these results were previously known, we have made an effort to distil the essential ingredients of several previous papers, and combined them into a unified, self-contained way, providing thus a more elementary introduction to the area, based only of graph theory, instead of higher dimensional simplicial complexes.

Acknowledgments

This work is partially supported by UNAM-PAPIIT grant IN106520 (Sergio Rajsbaum).

References

1. **Attiya, H., Rajsbaum, S. (2002).** The combinatorial structure of wait-free solvable tasks. SIAM J. Comput., Vol. 31, No. 4, pp. 1286–1313.
2. **Attiya, H., Rajsbaum, S. (2020).** Indistinguishability. Commun. ACM, Vol. 63, No. 5, pp. 90–99.
3. **Baltag, A., Moss, L., Solecki, S. (1998).** The logic of common knowledge, public announcements, and private suspicions. TARK VII, pp. 43–56.

4. **Baltag, A., Renne, B. (2016).** Dynamic epistemic logic. In The Stanford Encyclopedia of Philosophy, see `https://plato.stanford.edu/archives/win2016/entries/dynamic-epistemic/`. Metaphysics Research Lab, Stanford University.

5. **Biran, O., Moran, S., Zaks, S. (1990).** A Combinatorial Characterization of the Distributed 1-Solvable Tasks. J. Algorithms, Vol. 11, No. 3, pp. 420–440.

6. **Braud-Santoni, N., Dubois, S., Kaaouachi, M.-H., Petit, F. (2016).** The next 700 impossibility results in time-varying graphs. Int. J. Netw. Comput., Vol. 6, pp. 27–41.

7. **Castañeda, A., Fraigniaud, P., Paz, A., Rajsbaum, S., Roy, M., Travers, C. (2019).** A topological perspective on distributed network algorithms. **Censor-Hillel, K., Flammini, M.**, editors, Structural Information and Communication Complexity - 26th International Colloquium, SIROCCO 2019, L'Aquila, Italy, July 1-4, 2019, Proceedings, volume 11639 of Lecture Notes in Computer Science, Springer, pp. 3–18.

8. **Casteigts, A., Flocchini, P., Quattrociocchi, W., Santoro, N. (2011).** Time-varying graphs and dynamic networks. **Frey, H., Li, X., Ruehrup, S.**, editors, Ad-hoc, Mobile, and Wireless Networks, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 346–359.

9. **Charron-Bost, B., Függer, M., Nowak, T. (2015).** Approximate consensus in highly dynamic networks: The role of averaging algorithms. **Halldórsson, M. M., Iwama, K., Kobayashi, N., Speckmann, B.**, editors, Int. Colloquium on Automata, Languages, and Programming (ICALP), number 9135 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 528–539.

10. **Charron-Bost, B., Függer, M., Nowak, T. (2016).** Fast, Robust, Quantizable Approximate Consensus. **Chatzigiannakis, I., Mitzenmacher, M., Rabani, Y., Sangiorgi, D.**, editors, 43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016), volume 55 of Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 137:1–137:14.

11. **Coulouma, É., Godard, E., Peters, J. G. (2015).** A characterization of oblivious message adversaries for which consensus is solvable. Theor. Comput. Sci., Vol. 584, pp. 80–90.

12. **Delporte-Gallet, C., Fauconnier, H., Rajsbaum, S. (2020).** Communication complexity of wait-free computability in dynamic networks. **Richa, A. W., Scheideler, C.**, editors, Structural Information and Communication Complexity (SIROCCO), number 12156 in LNCS, Springer International Publishing, Cham, pp. 291–309.

13. **Ditmarsch, H. v., van der Hoek, W., Kooi, B. (2007).** Dynamic Epistemic Logic. Springer.

14. **Dolev, D., Lynch, N. A., Pinter, S. S., Stark, E. W., Weihl, W. E. (1986).** Reaching approximate agreement in the presence of faults. J. ACM, Vol. 33, No. 3, pp. 499–516.

15. **Fischer, M., Lynch, N. A., Paterson, M. S. (1985).** Impossibility Of Distributed Commit With One Faulty Process. Journal of the ACM, Vol. 32, No. 2, pp. 374–382.

16. **Függer, M., Nowak, T., Schwarz, M. (2018).** Tight bounds for asymptotic and approximate consensus. **Newport, C., Keidar, I.**, editors, Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom, July 23-27, 2018, ACM, pp. 325–334.

17. **Goubault, É., Lazić, M., Ledent, J., Rajsbaum, S. (2019).** A dynamic epistemic logic analysis of the equality negation task. Dynamic Logic. New Trends and Applications - Second International Workshop, DaLí 2019, Proceedings, pp. 53–70.

18. **Herlihy, M., Kozlov, D., Rajsbaum, S. (2013).** Distributed Computing Through Combinatorial Topology. Elsevier-Morgan Kaufmann.

19. **Herlihy, M., Rajsbaum, S., Raynal, M., Stainer, J. (2017).** From wait-free to arbitrary concurrent solo executions in colorless distributed computing. Theor. Comput. Sci., Vol. 683, pp. 1–21.

20. **Hoest, G., Shavit, N. (2006).** Towards a topological characterization of asynchronous complexity. SIAM J. Comput., Vol. 36, No. 2, pp. 457–497.

21. **Kuhn, F., Oshman, R. (2011).** Dynamic networks: models and algorithms. SIGACT News, Vol. 42, No. 1, pp. 82–96.

22. **Loui, M. C., Abu-Amara, H. H. (1987).** Memory requirements for agreement among unreliable asynchronous processes. volume 4 of Advances in Computing Res., JAI press, pp. 163–183.

23. **Moses, Y., Rajsbaum, S. (2002).** A layered analysis of consensus. SIAM J. Comput., Vol. 31, No. 4, pp. 989–1021.

24. **R. Fagin, Y. M., J. Halpern, Vardi, M. (1995).** Reasoning About Knowledge. MIT Press.

25. **Éric Goubault, Ledent, J., Rajsbaum, S. (2020).** A simplicial complex model for dynamic epistemic logic to study distributed task computability. Information and Computation, pp. 104597. In print, a preliminary version appeared in Proc. of GandALF 2018.

26. **van Ditmarsch, H., Goubault, E., Ledent, J., Rajsbaum, S. (2020).** Knowledge and simplicial complexes. arXiv e-prints . To appear in the journal Information and Computation, Elsevier.

27. **Winkler, K., Schmid, U. (2019).** An overview of recent results for consensus in directed dynamic networks. Bull. EATCS, Vol. 128.

Article received on 14/10/2020; accepted on 20/02/2021.
Corresponding author is Jorge Armenta-Segura.

ISSN 2007-9737

A Dual-Context Sequent Calculus for $\mathrm{S4}$ Modal Lambda-Term Synthesis

Favio E. Miranda-Perea[1], Sammantha Omaña Silva[2], Lourdes del Carmen González Huesca[1]

[1] Universidad Nacional Autónoma de México,
Facultad de Ciencias,
Departamento de Matemáticas,
Mexico

[2] Universidad Nacional Autónoma de México,
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas,
Posgrado en Ciencia e Ingeniería de la Computación,
Mexico

{favio, luglzhuesca}@ciencias.unam.mx, samm.omana@gmail.com

Abstract. In type-based program synthesis, the search of inhabitants in typed calculi can be seen as a process where a specification, given by a type A, is considered to be fulfilled if we can construct a λ-term M such that $M : A$, or more precisely if $\Gamma \vdash M : A$ holds, that is, if under some suitable assumptions Γ the term M inhabits the type A. In this paper, we tackle this inhabitation/synthesis problem for the case of modal types in the necessity fragment of the constructive logic $\mathrm{S4}$. Our approach is human-driven in the sense of the usual reasoning procedures of modern theorem provers. To this purpose we employ a so-called dual-context sequent calculus, where the sequents have two contexts, originally proposed to capture the notions of global and local truths without resorting to any formal semantics. The use of dual-contexts allows us to define a sequent calculus which, in comparison to other related systems for the same modal logic, exhibits simpler typing inference rules for the $\Box$ operator. In several cases, the task of searching for a term having subterms with modal types is reduced to the quest for a term containing only subterms typed by non modal propositions.

Keywords. Dual-context sequent calculus, constructive necessity, type inhabitation, modal lambda calculus, program synthesis.

1 Introduction

Modal logic, originated in Mathematics and Philosophy, plays nowadays an important role in Computer Science. The use of modalities is relevant in the theory of programming languages where modal formulas of the form $\Box A$ designate a type of encapsulated values, to be considered *enhanced*, related to *ordinary* values of type A.

For instance in staged computation [8] where $\Box A$ is the type of run-time generated code that computes values of type A. Another important reading of modal types comes out in mobile computation [23, 22] where $\Box A$ is the type of mobile code of type A. Other relevant interpretations of the necessity modality appear in the analysis of information flow either in computer networks [6] or in software security [21].

Coming from practical applications in the mentioned areas of computation, attention is focused on abstracting this kind of behaviors from real scenarios through specifications, which generates an outstanding task: the problem of constructing a program from a given specification.

In this paper we tackle a version of this synthesis problem at the foundational level given

ISSN 2007-9737

by lambda-term type-based synthesis. This is an instance of constructive/deductive synthesis [4] where a type A, coming from the constructive modal logic $\mathrm{S4}$ for necessity in our case, plays the role of the specification that the sought after lambda-term has to meet together with a context Γ to represent a collection of subcomponents, specified again by their types, considered as already synthetized.

This approach corresponds to the type inhabitation problem: *given a context Γ of type declarations for variables and a type A, is it possible to find a term M such that the typing $\Gamma \vdash M : A$ holds?* which in turn corresponds, under the Curry-Howard correspondence, to proof-search: *given a context of assumptions Γ and a formula A, find a derivation of the sequent $\Gamma \vdash A$.*

This problem has been addressed before in the case of modal logic [1, 13, 26] but not from the point of view we take here, which is an interactive human-driven process reminiscent of the ways of modern proof-assistants (like Coq [7]).

Let us show an example of the kind of programs (lambda-terms) we want to synthetize.

> The specification $\Box(A \to B) \to \Box A \to \Box B$ corresponds to a program K witnessing the fact that encapsulated values are well-behaved under function application. In staged computation this means that taking as inputs run-time generated codes of types $A \to B$ and A, program K produces a code of type B:
>
> $$\mathsf{K}\ f\ x =_{def} \mathsf{let\ box}\ f' = f\ \mathsf{in},\ \mathsf{let\ box}\ x' = x\ \mathsf{in},\ \mathsf{box}\ (f' \star x').$$
>
> Here the box constructor signals the encapsulation process: if e denotes an encapsulated value then, e' denotes its ordinary associated value so that the equality $\mathsf{box}\ e' = e$ holds. The $\star$ operator corresponds to function application of ordinary values retrieved from their encapsulated versions.

The above program K corresponds to the characteristic scheme of the modal logic $\mathrm{S4}$, namely $\mathbb{K}$. This example emphasizes the distinction between the two kinds of values[1]: ordinary and enhanced. The idea behind an enhanced value is that it does not depend on any ordinary value. In summary we will consider two kinds of values and two processes which can be applied to any value, namely encapsulation and retrieval. All these behaviors will be enforced by the syntax and the type system.

We start in Section 2 by discussing the dual-context sequent calculus $\mathcal{GS}4$, as a lambda-term type system. Our interactive program synthesis procedure is presented in Section 3, including an example. The soundness of the synthesis process is developed in Section 4, followed by some final remarks in Section 5.

2 A Dual-Context Sequent Calculus for Interactive Program-Synthesis

There are several discussions and applications involving deductive systems for modal logic, see [27, 15] for a deep overview. However, to the best of our knowledge, there is no dedicated sequent calculus presentation of constructive $\mathrm{S4}$ with the intention of human-driven proof-search or program synthesis.

Although, there are works on automated proof-search whose formalisms are therefore not suitable for high-level human-reasoning [1, 26, 13, 15, 18]. Let us review some proposed rules for $\mathrm{S4}$ present in the literature, adapted here for the case of constructive logic:

$$\frac{A, \Box A, \Gamma \vdash B}{\Box A, \Gamma \vdash B}\ (\Box\mathsf{L})\ .$$

Although this left rule is adequate for proof-search, keeping both A and $\Box A$ in the premise somehow entails redundant information and poses loop problems for automated proof-search, solved by means of sophisticated systems like the one

[1] Through the whole paper we allow ourselves to speak of values when referring to any expression which, perhaps after an adequate binding, would yield a well-defined value under any fixed dynamic semantics.

in [26]. In the case of right rules we mention two alternatives:

$$\frac{\Gamma^{\Box} \vdash A}{\Gamma^{\Box} \vdash \Box A}\ (\Box R2), \qquad \frac{\Gamma^{\circ} \vdash A}{\Gamma \vdash \Box A}\ (\Box R3),$$

where $\Gamma^{\Box}$ denotes a context with only boxed formulae and the context Γ° results from Γ by eliminating all the non modal formulae.

Apart from discarding the symmetry between left and right rules, rule $(\Box R2)$ is not suitable for proof-search, since it restricts the shape of the assumptions to be boxed formulas.

In the case of rule $(\Box R3)$, the conclusion sequent has the desired general form for proof-search, but in the generated subgoal $\Gamma^{\circ} \vdash A$, we can lose important information by passing from Γ to Γ°, although this can be alleviated by a clever decision taken by a human-agent, in order to avoid search flaws.

To mitigate the above mentioned issues and tackle the problem of type-based program synthesis we propose a sequent calculus which handles sequents of the form $\Delta \mid \Gamma \vdash M : A$ where M is a lambda-term and Δ and Γ are contexts. This kind of formalism for modal logic has its origins in systems for linear logic [3] and has been introduced in [24] for reconstructing modal logic in the light of Martin-Löf's meaning of logical constants and laws [17]. In this approach, propositions obtain their meaning through judgments without any semantic label (worlds), in particular, modal operators are defined by means of judgments over propositions.

The notion of so-called hypothetical judgments is extended to categorical judgments where a conclusion does not depend on hypotheses about the constructive truth of propositions.

Hence, a distinction of two forms of primitive judgments is essential: 'A *true*' means that we know how to verify A under hypothetical judgments, whereas 'A *valid*' represents the fact that A is a proposition whose truth does not depend on any hypotheses, thus internalizing a categorical judgment as a proposition syntactically represented by the modal formula $\Box A$.

A disengagement similar to the context separation in dual-context systems is present in several works, for instance the systems of Fitting [10]; or the work of Avron et al. [2].

We move now to the technical definitions. The types are generated by the following grammar where $\mathcal{B}$ denotes a collection of primitive basic types:

$$A, B ::= \mathcal{B} \mid A \to B \mid A \wedge B \mid A \vee B \mid \Box A.$$

Unlike other presentations, we include disjunction and conjunction. Also note that neither negation nor $\bot$ are present. Thus, we are dealing with minimal logic. Variable declaration contexts are defined by means of so-called *snoc* lists:

$$\Gamma ::= \cdot \mid \Gamma, x : A,$$

these are finite lists built from the empty list, denoted here by $\cdot$, and a binary constructor that generates a new list from a given one by adding a fresh variable x of type A to its right-end. The concatenation operation is inductively defined as expected and denoted by $\Gamma_1; \Gamma_2$.

In the below inference rules the idea behind the context separation is that hypotheses in Δ are enhanced (modal), whereas those in Γ are ordinary (intuitionistic). Nevertheless, this idea does not represent a syntactic restriction, for we can have arbitrary types, modal or intuitionistic, in both contexts. This is an important difference with other modal dual-context systems like those of [16].

Moreover, the context separation is strict, in particular it is forbidden to declare the same variable in both contexts. This is an important difference with [24] where there is only one context with two zones, a choice that requires the use of explicit labels *valid, true* in the context formulae.

Since their introduction, dual-context modal logics and their type systems have been presented in the sequent-style of natural deduction [16, 8, 22, 9].

Such formalisms are not suitable for backward proof-search, reason why we present a sequent calculus $\mathcal{GS}4$, adequate for our purposes. This system is inductively defined by the following inference rules, the corresponding lambda-terms encoding proofs are the target of the synthesis process discussed in detail in the next section:

ISSN 2007-9737

— Initial rules: we have two rules that allow to conclude a hypothesis according to the context it belongs:

$$\frac{}{\Delta \mid \Gamma, x : A; \Gamma' \vdash x : A}\ (\textsc{Thyp})\ ,$$

$$\frac{}{\Delta, x : A; \Delta' \mid \Gamma \vdash x : A}\ (\textsc{Vhyp})\ .$$

— Right rules:

$$\frac{\Delta \mid \Gamma \vdash M : A \qquad \Delta \mid \Gamma \vdash N : B}{\Delta \mid \Gamma \vdash \langle M, N\rangle : A \wedge B}\ (\wedge\,\mathsf{R})\ ,$$

$$\frac{\Delta \mid \Gamma \vdash M : A}{\Delta \mid \Gamma \vdash \mathsf{inl}\, M : A \vee B}\ (\vee\,\mathsf{R})\ ,$$

$$\frac{\Delta \mid \Gamma \vdash M : B}{\Delta \mid \Gamma \vdash \mathsf{inr}\, M : A \vee B}\ (\vee\,\mathsf{R})\ ,$$

$$\frac{\Delta \mid \Gamma, x : A \vdash N : B}{\Delta \mid \Gamma \vdash \lambda x.N : A \to B}\ (\to\mathsf{R})\ ,$$

$$\frac{\Delta \mid \cdot \vdash M : A}{\Delta \mid \Gamma \vdash \mathsf{box}\, M : \Box A}\ (\Box\,\mathsf{R})\ .$$

The rules for propositional connectives are standard. In the case of a modal formula $\Box A$ the right rule corresponds to the so-called necessitation rule and allows us to introduce the box operator on the right hand side of the turnstile, only in the absence of ordinary assumptions.

It is important to remark that this right rule, as rule $(\Box R3)$, also suffers from loss of information in its backward reading.

However, such issue can be avoided in some cases by transferring to Δ some or all the boxed assumptions in Γ, which is not possible with $(\Box R3)$.

The left rules come in two versions, one for each context.

— Left rules for the ordinary context:

$$\frac{\Delta \mid \Gamma, x : A, y : B; \Gamma' \vdash M : C}{\Delta \mid \Gamma, z : A \wedge B; \Gamma' \vdash \mathsf{letpair}(z, x.y.M) : C}\ (\wedge\mathsf{L})\ ,$$

$$\frac{\begin{array}{c}\Delta \mid \Gamma, x : A; \Gamma' \vdash M : C \\ \Delta \mid \Gamma, y : B; \Gamma' \vdash N : C\end{array}}{\Delta \mid \Gamma, z : A \vee B; \Gamma' \vdash \mathsf{case}(z, x.M, y.N) : C}\ (\vee\,\mathsf{L})\ ,$$

$$\frac{\Delta \mid \Gamma, x : A \to B; \Gamma' \vdash M : A}{\Delta \mid \Gamma, x : A \to B; \Gamma' \vdash xM : B}\ (\to\mathsf{L})\ ,$$

$$\frac{\Delta, x : A \mid \Gamma; \Gamma' \vdash M : B}{\Delta \mid \Gamma, y : \Box A; \Gamma' \vdash \mathsf{letbox}(y, x.M) : B}\ (\Box\,\mathsf{L})\ .$$

For disjunction and conjunction, the rules are standard. The left rule $(\Box L)$ represents a type transference principle between contexts: in the proof-search process we can move an encapsulated type in the ordinary context to the enhanced context by unboxing it.

The rule $(\to$ L) is not usual, for instead of decomposing the implicative hypothesis, in order to prove/use its components, like the regular left rule for implication, it only uses it to derive its consequent, once its antecedent has been derived.

This rule captures the local reasoning with implication common in informal proofs, instead the usual left rule for implication models an on the-fly prove/use of a lemma. This feature rarely figures in actual paper-and-pencil proofs and thus makes the original rule clumsy for proof-search purposes.

To the best of our knowledge this left rule, which is inspired by the `apply` tactic of the Coq proof-assistant, has been considered only by us [20, 19], though it is also related to the rule $(\to L)^\circ$ of Schroeder-Heister [28].

Let us also note that, under the presence of the cut or substitution rule, the rule $(\to$L) is equivalent to the ordinary left rule for implication. The details of this claim are omitted due to lack of space.

— Left rules for the enhanced context:

$$\frac{\Delta, x : A, y : B; \Delta' \mid \Gamma \vdash M : C}{\Delta, z : A \wedge B; \Delta' \mid \Gamma \vdash \mathsf{eletpair}(z, x.y.M) : C} \ (\wedge\mathsf{LE}),$$

$$\frac{\begin{array}{c}\Delta; \Delta' \mid \Gamma, x : A \vdash M : C \\ \Delta; \Delta' \mid \Gamma, y : B \vdash N : C\end{array}}{\Delta, z : A \vee B; \Delta' \mid \Gamma \vdash \mathsf{ecase}(z, x.M, y.N) : C} \ (\vee\mathsf{LE}),$$

$$\frac{\Delta, x : A \to B; \Delta' \mid \Gamma \vdash M : A}{\Delta, x : A \to B; \Delta' \mid \Gamma \vdash x \star M : B} \ (\to\mathsf{LE}),$$

$$\frac{\Delta, x : A; \Delta' \mid \Gamma \vdash M : B}{\Delta, y : \Box A; \Delta' \mid \Gamma \vdash \mathsf{eletbox}(y, x.M) : B} \ (\Box\,\mathsf{LE}).$$

The rule for conjunction is again standard, whereas for implication the rule is analogous to the version for ordinary contexts. In the case of an enhanced disjunctive hypothesis the case analysis on $z : A \vee B$ is performed only by ordinary hypotheses $x : A$ and $y : B$, otherwise the rule would be unsound [2]. Finally, the rule $\Box LE$, introduced by us in [19], reduces the synthesis of a program involving an enhanced and encapsulated component $\Box A$ to the search of a program involving only the non-encapsulated enhanced component A.

— Substitution or cut rules: these are essential for human-driven proof-search:

$$\frac{\Delta \mid \Gamma \vdash M : A \qquad \Delta \mid \Gamma, x : A \vdash N : B}{\Delta \mid \Gamma \vdash \mathsf{let}(M, x.N) : B} \ (\textsc{Subst}),$$

$$\frac{\Delta \mid \cdot \vdash M : A \qquad \Delta, x : A \mid \Gamma \vdash N : B}{\Delta \mid \Gamma \vdash \mathsf{elet}(M, x.N) : B} \ (\textsc{SubstE}).$$

The enhanced substitution rule (SUBSTE) is derivable from (SUBST), but we keep both rules for the sake of symmetry. Moreover, it is relevant to mention that this ordinary rule is not admissible. This is not important for our purposes since the reasoning pattern provided by the cut rule is essential for human-driven proof search. A further discussion on this matter has to be presented elsewhere, due to lack of space.

With respect to the left rules for necessity we consider important to emphasize that, since they permit to encapsulate/retrieve values at the hypotheses level, the synthesis process involving an enhanced value can be reduced to one that requires only an ordinary value. This transference process (see [11, Section 4.2]) allows us to reason in a more intuitive and straighforward way, as shown in the example of Section 3.

To finish the section is important to review the more usual elimination rule for $\Box$ in dual-context systems presented for instance in [25, 8, 16, 9]. This typing rule for a letbox operator is:

$$\frac{\Delta \mid \Gamma \vdash M : \Box A \qquad \Delta, x : A \mid \Gamma \vdash N : B}{\Delta \mid \Gamma \vdash \mathsf{letb}\, x = M \,\mathsf{in}\, N : B} \ (\Box\ \mathsf{E}).$$

We can observe that such rule entails a specific substitution (cut) process that provides the primitive way of using an ordinary value (the assumption $x : A$) retrieved from an enhanced value (the term $M : \Box A$), at the price of requiring an explicit derivation of M. Of course this is characteristic of (generalized) elimination rules in sequent-style natural deduction and can be simulated in $\mathcal{GS}4$ by means of the $(\Box L)$ and (SUBST) rules. The definition of letb, in concrete syntax, witnessing this simulation is: $\mathsf{letb}\, x = M \,\mathsf{in}\, N =_{def} \mathsf{let}\, y = M \,\mathsf{in}\, \mathsf{letbox}\, x = y \,\mathsf{in}\, N$.

It is easy to see that in a dual-context natural deduction system, with $(\Box E)$ as a primitive rule, our rules $(\Box L)$ and $(\Box LE)$ can be simulated as well. Thus, both systems turn out to be equivalent (more details are provided in [19]). Moreover, in [11] we prove that the system of dual-context natural deduction is equivalent to an S4-axiomatic system. Then we can conclude that the present system captures exactly the necessity fragment of the constructive logic S4.

3 Interactive Program Synthesis

The lambda-terms that appear in the sequent calculus typing rules of Section 2, formalize the kind of programs target of the synthesis process. They include modal constructors in the lines of some related languages [25, 8, 16, 9] as well as distinct variable binding operators (let or case expressions). Let us collect them precisely.

[2] For instance, it would allow to derive $\Box(A \vee B) \to \Box A \vee \Box B$, which is invalid in all known semantics of S4.

ISSN 2007-9737

Definition 3.1. *A pseudoterm is an expression generated by the following grammar:*

$$\begin{array}{rcl} M & ::= & x \mid X \mid R \mid O \mid E \mid P \\ R & ::= & \lambda x.M \mid \langle M, M\rangle \mid \mathsf{inl}\, M \mid \mathsf{inr}\, M \mid \mathsf{box}\, M \\ O & ::= & M\, M \mid \mathsf{letpair}(M, x.y.M) \mid \\ & & \mathsf{case}(M, x.M, y.M) \mid \mathsf{letbox}(M, y.M) \\ E & ::= & M \star M \mid \mathsf{eletpair}(M, x.y.M) \mid \\ & & \mathsf{ecase}(M, x.M, y.M) \mid \mathsf{eletbox}(M, y.M) \\ P & ::= & \mathsf{let}(M, x.M) \mid \mathsf{elet}(M, x.M) \end{array}$$

The metavariable M denotes pseudoterms which are classified as right pseudoterms, those generated by the metavariable R; ordinary pseudoterms, generated by O and enhanced pseudoterms, generated by E. A left pseudoterm is either an ordinary or an enhanced pseudoterm. Finally, those generated by P are called strong let-expressions.

As our matter of interest is to synthetize lambda-terms we define pseudoterms, which are expressions corresponding to programs with unknown templates to be filled during the synthesis process. A basic and least informative template is represented by a category of metavariables or search-variables, disjoint from the usual term variables, denoted by a capital X. A pseudoterm M is called a *term* or a *program* if M does not contain search variables.

Apart from the usual lambda term constructors for functions, sums and products, we use twin program constructors in order to make explicit the manipulation of ordinary or enhanced values, the only difference being a prefix e indicating the need for an enhanced input (we use an infix application operator $\star$ in the case of an enhanced function). There is no need to have twin constructors for right pseudoterms, due to the fact that an enhanced value can be constructed directly by the box operator.

Let us sketch next our program synthesis technique: given two contexts of type declarations for variables, Δ and Γ (for enhanced and ordinary assumptions respectively) and a type specification A, the goal is to construct a program M such that the typing $\Delta \mid \Gamma \vdash M : A$ holds. The program M is currently unknown and it is represented with a search-variable X. The task is to find a value for X such that $\Delta \mid \Gamma \vdash X : A$ holds.

Analyzing the specific form either of A or of some assumption type; and applying a backward reading of the typing rules, some restrictions on the form of M are generated in order to give a solution for X. These conditions are given by pseudoterm equations which, if solvable, will allow to construct the desired program.

For instance, the search for an X such that $\cdot \mid x : A \vdash X : A \vee B$ holds, is reduced to the search for a Y such that $\cdot \mid x : A \vdash Y : A$ holds. The search-variables X and Y are related by the equation $X \approx \mathsf{inl}\, Y$. The last goal is directly solved by the equation $Y \approx x$. By solving these two equations, we obtain that $M =_{def} \mathsf{inl}\, x$ verifies $\cdot \mid x : A \vdash M : A \vee B$.

However, let us observe that sequents involving search-variables are not derivable, for, according to Section 2, there is no typing rule for this kind of variables. Such underivable sequents represent synthesis problems, which are program search tasks formalized by the following notion of pseudosequent.

Definition 3.2. *A pseudosequent or search-sequent, $\mathcal{P}$, is a $4-$tuple of the form $\Delta \mid \Gamma \vdash_? X : A$ where X is a search-variable.*

For the synthesis process we will need to handle finite sequences of pseudosequents defined as follows:

Definition 3.3. *The set of finite sequences of pseudosequents is recursively defined with the grammar:*

$$\mathcal{S} ::= \bullet \mid \mathcal{P}\,,\, \mathcal{S}$$

where $\bullet$ denotes the empty sequence. For clarity, a singleton sequence is identified with its unique element. As for contexts, the semicolon operator ; is used for concatenation of pseudosequents sequences.

The restrictions generated during the program search will be captured by constraint sets defined as follows:

Definition 3.4. *A constraint is an equation of the form $X \approx e$ where X is a search-variable and e is a pseudoterm. A set of equations:*

$$\mathcal{R} = \{X_1 \approx e_1, X_2 \approx e_2, \ldots, X_k \approx e_k\},$$

where all X_i are different, is called a constraint set. According to the category of the pseudoterm e, a constraint can be right, ordinary, enhanced, left or strong.

Next we define what is a solution of a constraint set.

Definition 3.5. *Given a constraint set $\mathcal{R}$, a pseudoterm M is solution of the constraint $X \approx e$ in $\mathcal{R}$, if M is a term and there is a substitution*[3] *of search-variables by terms, say $\sigma = [X_1, \ldots, X_n/M_1, \ldots, M_n]$, such that $M \equiv e\sigma$ (i.e M is syntactically identical to $e\sigma$ up-to α-equivalence). In such case the solution is written as $M = Sol_{\mathcal{R}}(X \approx e)$.*

Sequences of pseudosequents interact with constraint sets by means of goals, defined as follows.

Definition 3.6. *A goal is a pair $\mathcal{S} \parallel \mathcal{R}$ consisting of a sequence of pseudosequents $\mathcal{S}$ and a constraint set $\mathcal{R}$. The set of goals is denoted by* Goal.

The program synthesis process is defined next by means of a transition system where the goals play the role of states and the transitions, which are called tactics, transform a goal into another goal according to the backward reading of the typing rules. We consider this formal definition and handling of backward lambda-term synthesis as the second main contribution of this paper.

Definition 3.7. *The transition system is defined as follows:*

- *A state is a goal $\mathcal{S} \parallel \mathcal{R}$.*
- *An initial state is a goal of the form $\Delta \mid \Gamma \vdash_? X : A \parallel \varnothing$, that is, a goal composed of a unique pseudosequent and the empty constraint set.*
- *A terminal state is a goal of the form $\bullet \parallel \mathcal{R}$, that is, a goal composed of the empty sequence of pseudosequents and an arbitrary constraint set.*
- *The transition relation $\rhd \subseteq$ Goal $\times$ Goal is inductively defined by the axioms and inference rule below, where a transition $\mathcal{S}_1 \parallel \mathcal{R}_1 \rhd \mathcal{S}_2 \parallel \mathcal{R}_2$ can be read as* to solve the current goal $\mathcal{S}_1 \parallel \mathcal{R}_1$ it suffices to solve the subgoal $\mathcal{S}_2 \parallel \mathcal{R}_2$.

In each basic transition, the search-sequent $\Delta \mid \Gamma \vdash_? X : A$ dictates the action according to the backward reading of a typing rule, updating the constraint set accordingly.

In the following, we define the transition system axioms according to four synthesis process. In each case we assume that the search-variables introduced in the pseudosequents of the reduct are fresh, that is, do not occur in the redex.

3.1 Direct Synthesis

Direct synthesis triggers the synthesis by directly analyzing the shape of the typing specification producing a right constraint:

$$\begin{array}{l}
\texttt{intro}\ x: \\
\Delta \mid \Gamma \vdash_? X : A \to B \parallel \mathcal{R} \rhd \\
\quad \Delta \mid \Gamma, x : A \vdash_? X_1 : B \parallel \mathcal{R}, X \approx \lambda x.X_1, \\
\texttt{split}: \\
\Delta \mid \Gamma \vdash_? X : A \wedge B \parallel \mathcal{R} \rhd \\
\quad \Delta \mid \Gamma \vdash_? X_1 : A\ ; \\
\quad \Delta \mid \Gamma \vdash_? X_2 : B \parallel \mathcal{R}, X \approx \langle X_1, X_2 \rangle, \\
\texttt{left}: \\
\Delta \mid \Gamma \vdash_? X : A \vee B \parallel \mathcal{R} \rhd \\
\quad \Delta \mid \Gamma \vdash_? X_1 : A \parallel \mathcal{R}, X \approx \mathsf{inl}\, X_1, \\
\texttt{right}: \\
\Delta \mid \Gamma \vdash_? X : A \vee B \parallel \mathcal{R} \rhd \\
\quad \Delta \mid \Gamma \vdash_? X_1 : B \parallel \mathcal{R}, X \approx \mathsf{inr}\, X_1, \\
\texttt{unbox}: \\
\Delta \mid \Gamma \vdash_? X : \Box A \parallel \mathcal{R} \rhd \\
\quad \Delta \mid \cdot \vdash_? X_1 : A \parallel \mathcal{R}, X \approx \mathsf{box}\, X_1,
\end{array}$$

[3] Substitution in the usual sense.

ISSN 2007-9737

3.2 Indirect Synthesis

Indirect synthesis focuses on a type in any of the contexts, generating a left constraint:

$$\begin{array}{l}
\texttt{apply } x: \\
\Delta \mid \Gamma, x : A \to B; \Gamma' \vdash_? X : B \parallel \mathcal{R} \rhd \\
\quad \Delta \mid \Gamma, x : A \to B; \Gamma' \vdash_? X_1 : A \parallel \mathcal{R}, X \approx x X_1, \\
\texttt{apply } x: \\
\Delta, x : A \to B; \Delta' \mid \Gamma \vdash_? X : B \parallel \mathcal{R} \rhd \\
\quad \Delta, x : A \to B; \Delta' \mid \Gamma \vdash_? X_1 : A \parallel \mathcal{R}, X \approx x \star X_1, \\
\texttt{destruct } z: \\
\Delta \mid \Gamma, z : A \wedge B; \Gamma' \vdash_? X : C \parallel \mathcal{R} \rhd \\
\quad \Delta \mid \Gamma, x : A, y : B; \Gamma' \vdash_? X_1 : C \parallel \mathcal{R}, X \approx \mathsf{letpair}(z, x.y.X_1), \\
\texttt{destruct } z: \\
\Delta, z : A \wedge B; \Delta' \mid \Gamma \vdash_? X : C \parallel \mathcal{R} \rhd \\
\quad \Delta, x : A, y : B; \Delta' \mid \Gamma \vdash_? X_1 : C \parallel \mathcal{R}, X \approx \mathsf{eletpair}(z, x.y.X_1), \\
\texttt{destruct } z: \\
\Delta \mid \Gamma, z : A \vee B; \Gamma' \vdash_? X : C \parallel \mathcal{R} \rhd \\
\quad \Delta \mid \Gamma, x : A; \Gamma' \vdash_? X_1 : C ; \\
\quad \Delta \mid \Gamma, y : B; \Gamma' \vdash_? X_2 : C \parallel \mathcal{R}, X \approx \mathsf{case}(z, x.X_1, y.X_2), \\
\texttt{destruct } z: \\
\Delta, z : A \vee B; \Delta' \mid \Gamma \vdash_? X : C \parallel \mathcal{R} \rhd \\
\quad \Delta; \Delta' \mid \Gamma, x : A \vdash_? X_1 : C ; \\
\quad \Delta; \Delta' \mid \Gamma, y : B \vdash_? X_2 : C \parallel \mathcal{R}, X \approx \mathsf{ecase}(z, x.X_1, y.X_2),
\end{array}$$

$$\begin{array}{l}
\texttt{retrieve } x: \\
\Delta \mid \Gamma, x : \Box A; \Gamma' \vdash_? X : B \parallel \mathcal{R} \rhd \\
\quad \Delta, y : A \mid \Gamma; \Gamma' \vdash_? X_1 : B \parallel \mathcal{R}, X \approx \mathsf{letbox}(x, y.X_1), \\
\texttt{retrieve } x: \\
\Delta, x : \Box A; \Delta' \mid \Gamma \vdash_? X : B \parallel \mathcal{R} \rhd \\
\quad \Delta, y : A; \Delta' \mid \Gamma \vdash_? X_1 : B \parallel \mathcal{R}, X \approx \mathsf{eletbox}(x, y.X_1).
\end{array}$$

In this case, distinct tactics have the same name for they share the same functionality: the `apply` tactics correspond to the application of a functional hypothesis; the `destruct` tactics trigger the destruction of a specific hypothesis, replacing it by simpler hypotheses in the subgoals. Finally, the manipulation of modal hypotheses is managed by the `retrieve` tactics.

3.3 Strong Synthesis

Strong synthesis requires to guess the type corresponding to the first premise of a substitution rule and generates a strong constraint. This calls for an explicit interaction [4] with the human agent, which is why we speak of a strong synthesis:

[4] Nevertheless, the reader can note that the indirect synthesis process also requires interactivity in order to choose an adequate hypothesis and match a particular tactic.

$$\begin{array}{l}
\texttt{assert A}: \\
\Delta \mid \Gamma \vdash_? X : C \parallel \mathcal{R} \rhd \\
\quad \Delta \mid \Gamma \vdash_? X_1 : A ; \\
\quad \Delta \mid \Gamma, x : A \vdash_? X_2 : C \parallel \mathcal{R}, X \approx \mathsf{let}(X_1, x.X_2), \\
\texttt{enough A}: \\
\Delta \mid \Gamma \vdash_? X : C \parallel \mathcal{R} \rhd \\
\quad \Delta \mid \Gamma, x : A \vdash_? X_1 : C; \\
\quad \Delta \mid \Gamma \vdash_? X_2 : A \parallel \mathcal{R}, X \approx \mathsf{let}(X_2, x.X_1), \\
\texttt{eassert A}: \\
\Delta \mid \Gamma \vdash_? X : C \parallel \mathcal{R} \rhd \\
\quad \Delta \mid \cdot \vdash_? X_1 : A ; \\
\quad \Delta, x : A \mid \Gamma \vdash_? X_2 : C \parallel \mathcal{R}, X \approx \mathsf{elet}(X_1, x.X_2), \\
\texttt{eenough A}: \\
\Delta \mid \Gamma \vdash_? X : C \parallel \mathcal{R} \rhd \\
\quad \Delta, x : A \mid \Gamma \vdash_? X_1 : C ; \\
\quad \Delta \mid \cdot \vdash_? X_2 : A \parallel \mathcal{R}, X \approx \mathsf{elet}(X_2, x.X_1).
\end{array}$$

Let us observe that each substitution rule has two corresponding tactics, namely `assert` and `enough`, the difference being only operational: either we first synthetize the auxiliar component A and then invoke it or viceversa.

3.4 Immediate Synthesis

Immediate synthesis generates a synthesis problem which is trivially solved by an initial inference rule. The constraints generated here are necessarily of the form $X \approx x$.

$$\begin{array}{l}
\texttt{assumption}: \\
\Delta \mid \Gamma, x : A; \Gamma' \vdash_? X : A \parallel \mathcal{R} \rhd \bullet \parallel \mathcal{R}, X \approx x, \\
\texttt{eassumption}: \\
\Delta, x : A; \Delta' \mid \Gamma \vdash_? X : A \parallel \mathcal{R} \rhd \bullet \parallel \mathcal{R}, X \approx x.
\end{array}$$

The names of some of the above tactics, though not the ones involving modal types, coincide with the names of analogous tactics implemented in the Coq proof assistant.

3.5 Inference Rule for Transition

All the above tactics constitute the basic axioms of the transition system. Next we give the sole inference rule for transitions.

Sequencing:

Several basic transitions cause the sequence of search-sequents in a goal to grow. In such cases we need to choose a specific search-sequent within the current goal to perform the next transition. The sequencing rule (seq) determines the choice order, namely from the first (left most) search-sequent:

$$\frac{\mathcal{S}_1 \parallel \mathcal{R}_1 \rhd \mathcal{S}_2 \parallel \mathcal{R}_2}{\mathcal{S}_1; \mathcal{S} \parallel \mathcal{R}_1 \rhd \mathcal{S}_2; \mathcal{S} \parallel \mathcal{R}_2}(\mathsf{seq}).$$

We show an example of how the transition system of tactics performs the synthesis process. Let us recall that the solution of a constraint is not unique and depends on the solutions of the later constraints found in the synthesis process. The application of any tactic guarantees that each search-variable appears in a pseudoterm in the constraint set.

Example 3.1. *A program t corresponding to the specification given by the modal scheme $\mathbb{K}$ is synthetized as follows.*

$\cdot \mid \cdot \vdash_? X : \Box(A \to B) \to \Box A \to \Box B \parallel \emptyset,$
$\rhd$ `intro` x
$\cdot \mid x : \Box(A \to B) \vdash_? X_1 : \Box A \to \Box B \parallel X \approx \lambda x.X_1,$
$\rhd$ `intro` y
$\cdot \mid x : \Box(A \to B), y : \Box A \vdash_? X_2 : \Box B \parallel \mathcal{R}, X_1 \approx \lambda y.X_2,$
$\rhd$ `retrieve` x
$x_1 : A \to B \mid y : \Box A \vdash_? X_3 : \Box B \parallel \mathcal{R}', X_2 \approx \mathsf{letbox}(x, x_1.X_3),$
$\rhd$ `retrieve` y
$x_1 : A \to B, y_1 : A \mid \cdot \vdash_? X_4 : \Box B \parallel \mathcal{R}'', X_3 \approx \mathsf{letbox}(y, y_1.X_4),$
$\rhd$ `unbox`
$x_1 : A \to B, y_1 : A \mid \cdot \vdash_? X_5 : B \parallel \mathcal{R}''', X_4 \approx \mathsf{box}\, X_5,$
$\rhd$ `apply` x_1
$x_1 : A \to B, y_1 : A \mid \cdot \vdash_? X_6 : A \parallel \mathcal{R}^{iv}, X_5 \approx x_1 \star X_6,$
$\rhd$ `eassumption`,
$\bullet \parallel \mathcal{R}''', X_4 \approx \mathsf{box}\, X_5, X_5 \approx x_1 \star X_6, X_6 \approx y_1,$

where

$$\begin{array}{rcl} \mathcal{R} & = & X \approx \lambda x.X_1, \\ \mathcal{R}' & = & X \approx \lambda x.X_1,\ X_1 \approx \lambda y.X_2, \\ \mathcal{R}'' & = & \mathcal{R}', X_2 \approx \mathsf{letbox}(x, x_1.X_3), \\ \mathcal{R}''' & = & \mathcal{R}'', X_3 \approx \mathsf{letbox}(y, y_1.X_4), \\ \mathcal{R}^{iv} & = & \mathcal{R}''', X_4 \approx \mathsf{box}\, X_5, \end{array}$$

and the solution for X is:
$t =_{def} \lambda x.\lambda y.\, \mathsf{letbox}(x, x.\, \mathsf{letbox}(y, y.\, \mathsf{box}(x_1 \star y_1)))$.

Example 3.2. *A program M corresponding to the specification given by type $\Box(A \wedge B) \to (\Box A \wedge \Box B)$ is synthetized as follows.*

$\cdot \mid \cdot \vdash_? X : \Box(A \wedge B) \to (\Box A \wedge \Box B) \parallel \emptyset,$
$\rhd$ `intro` x
$\cdot \mid x : \Box(A \wedge B) \vdash_? X_1 : \Box A \wedge \Box B \parallel X \approx \lambda x.X_1,$
$\rhd$ `retrieve` x
$x_1 : A \wedge B \mid \cdot \vdash_? X_2 : \Box A \wedge \Box B \parallel \mathcal{R}, X_1 \approx \mathsf{letbox}(x, x.X_2),$
$\rhd$ `destruct` x_1
$y_1 : A, y_2 : B \mid \cdot \vdash_? X_3 : \Box A \wedge \Box B \parallel \mathcal{R}', X_2 \approx \mathsf{eletpair}(x, y_1.y_2.X_3),$
$\rhd$ `split`
$y_1 : A, y_2 : B \mid \cdot \vdash_? X_4 : \Box A$;
$y_1 : A, y_2 : B \mid \cdot \vdash_? X_5 : \Box B \parallel \mathcal{R}'', X_3 \approx \langle X_4, X_5 \rangle,$
$\rhd$ `unbox`,
$y_1 : A, y_2 : B \mid \cdot \vdash_? X_6 : A$;
$y_1 : A, y_2 : B \mid \cdot \vdash_? X_5 : \Box B \parallel \mathcal{R}'', X_3 \approx \langle X_4, X_5 \rangle, X_4 \approx \mathsf{box}\, X_6,$
$\rhd$ `eassumption`
$\bullet$; $y_1 : A, y_2 : B \mid \cdot \vdash_? X_5 : \Box B \parallel \mathcal{R}''', X_6 \approx y_1,$
$\rhd$ `unbox`
$y_1 : A, y_2 : B \mid \cdot \vdash_? X_7 : B \parallel \mathcal{R}''', X_6 \approx y_1, X_5 \approx \mathsf{box}\, X_7,$
$\rhd$ `eassumption`
$\bullet \parallel \mathcal{R}''', X_6 \approx y_1, X_5 \approx \mathsf{box}\, X_7, X_7 \approx y_2,$

where

$$\begin{array}{rcl} \mathcal{R} & = & X \approx \lambda x.X_1, \\ \mathcal{R}' & = & \mathcal{R}, X_1 \approx \mathsf{letbox}(x, x_1.X_2), \end{array}$$

$$\begin{array}{rcl} \mathcal{R}'' & = & \mathcal{R}', X_2 \approx \mathsf{eletpair}(x_1, y_1.y_2.X_3), \\ \mathcal{R}''' & = & \mathcal{R}'', X_3 \approx \langle X_4, X_5 \rangle, X_4 \approx \mathsf{box}\, X_6, \end{array}$$

and the solution for X is:
$M =_{def} \lambda x.\, \mathsf{letbox}(x, x_1.\, \mathsf{eletpair}(x_1, y_1.y_2.\langle \mathsf{box}\, y_1, \mathsf{box}\, y_2 \rangle))$

Example 3.3. *A program M meeting the specification given by $\Box(\Box A \vee \Box B) \to \Box(A \vee B)$ is synthetized as follows:*

$\cdot \mid \cdot \vdash_? X : \Box(\Box A \vee \Box B) \to \Box(A \vee B) \parallel \emptyset,$
$\rhd$ `intro` x
$\cdot \mid x : \Box(\Box A \vee \Box B) \vdash_? X_1 : \Box(A \vee B) \parallel X \approx \lambda x.X_1,$
$\rhd$ `retrieve` x
$x_1 : \Box A \vee \Box B \mid \cdot \vdash_? X_2 : \Box(A \vee B) \parallel \mathcal{R}, X_1 \approx \mathsf{letbox}(x, x_1.X_2),$
$\rhd$ `unbox`
$x_1 : \Box A \vee \Box B \mid \cdot \vdash_? X_3 : A \vee B \parallel \mathcal{R}', X_2 \approx \mathsf{box}\, X_3,$
$\rhd$ `destruct` x_1,
$\cdot \mid x_2 : \Box A \vdash_? X_4 : A \vee B$;
$\cdot \mid x_3 : \Box B \vdash_? X_5 : A \vee B \parallel \mathcal{R}'', X_3 \approx \mathsf{ecase}(x_1, x_2.X_4, x_3.X_5),$
$\rhd$ `left`
$\cdot \mid x_2 : \Box A \vdash_? X_6 : A$;
$\cdot \mid x_3 : \Box B \vdash_? X_5 : A \vee B \parallel \mathcal{R}''', X_4 \approx \mathsf{inl}\, X_6,$
$\rhd$ `retrieve` x_2
$x_4 : A \mid \cdot \vdash_? X_7 : A$;
$\cdot \mid x_3 : \Box B \vdash_? X_5 : A \vee B \parallel \mathcal{R}^{iv}, X_6 \approx \mathsf{letbox}(x_2, x_4.X_7),$
$\rhd$ `eassumption`
$\bullet$; $\cdot \mid x_3 : \Box B \vdash_? X_5 : A \vee B \parallel \mathcal{R}^{v}, X_7 \approx x_4,$
$\rhd$ `right`
$\cdot \mid x_3 : \Box B \vdash_? X_8 : B \parallel \mathcal{R}^{v}, X_7 \approx x_4, X_5 \approx \mathsf{inr}\, X_8$
$\rhd$ `retrieve` x_3
$x_5 : B \mid \cdot \vdash_? X_9 : B \parallel \mathcal{R}^{vi}, X_8 \approx \mathsf{letbox}(x_3, x_5.X_9),$
$\rhd$ `eassumption`
$\bullet \parallel \mathcal{R}^{vi}, X_8 \approx \mathsf{letbox}(x_3, x_5.X_9), X_9 \approx x_5,$

ISSN 2007-9737

where

$$\begin{array}{rcl}\mathcal{R} & = & X \approx \lambda x.X_1,\\ \mathcal{R}' & = & \mathcal{R},\ X_1 \approx \mathsf{letbox}(x, x_1.X_2),\\ \mathcal{R}'' & = & \mathcal{R}',\ X_2 \approx \mathsf{box}\, X_3,\\ \mathcal{R}''' & = & \mathcal{R}'',\ X_3 \approx \mathsf{ecase}(x_1, x_2.X_4, x_3.X_5),\\ \mathcal{R}^{iv} & = & \mathcal{R}''',\ X_4 \approx \mathsf{inl}\, X_6,\\ \mathcal{R}^{v} & = & \mathcal{R}^{iv},\ X_6 \approx \mathsf{letbox}(x_2, x_4.X_7),\\ \mathcal{R}^{vi} & = & \mathcal{R}^{v}, X_7 \approx x_4,\ X_5 \approx \mathsf{inr}\, X_8,\end{array}$$

and the solution for X is:

$$M =_{def} \lambda x.\, \mathsf{letbox}(x, x_1.\, \mathsf{box}(\mathsf{ecase}(x_1, x_2.\, \mathsf{inl}\ \mathsf{letbox}(x_2, x_4.x_4), x_3.\, \mathsf{inr}\, \mathsf{letbox}(x_3, x_5.x_5))))$$

Let us observe that, since the `unbox` and `retrieve` tactics replace a modal with a pure propositional assumption, as announced, we are replacing a modal reasoning with a propositional inference.

4 Soundness of the Synthesis Process

In this section we prove the soundness of the synthesis process. Given an initial goal $\Delta \mid \Gamma \vdash_? X : A \parallel \varnothing$ we want to guarantee that: if the transition relation succeeds, that is, if applying the transition rules a finite number of times from this goal, we arrive to a final goal $\bullet \parallel \mathcal{R}$, then there is a program M, constructed by solving the contraints in $\mathcal{R}$, such that $\Delta \mid \Gamma \vdash M : A$ holds. Let us start by stating a convenient definition for the transitive closure of the transition relation.

Definition 4.1. *The transitive closure of the relation $\rhd$, denoted $\rhd^+$, is inductively defined by the following rules:*

$$\frac{\mathcal{S} \parallel \mathcal{R} \rhd \mathcal{S}' \parallel \mathcal{R}'}{\mathcal{S} \parallel \mathcal{R} \rhd^+ \mathcal{S}' \parallel \mathcal{R}'},$$

$$\frac{\mathcal{S} \parallel \mathcal{R} \rhd \mathcal{S}' \parallel \mathcal{R}' \qquad \mathcal{S}' \parallel \mathcal{R}' \rhd^+ \mathcal{S}'' \parallel \mathcal{R}''}{\mathcal{S} \parallel \mathcal{R} \rhd^+ \mathcal{S}'' \parallel \mathcal{R}''}.$$

Given an initial goal $\Delta \mid \Gamma \vdash_? X : A \parallel \varnothing$, let us observe that the transition process succeeds from this goal exactly when there is a constraint set $\mathcal{R}$ such that $\Delta \mid \Gamma \vdash_? X : A \parallel \varnothing \rhd^+ \bullet \parallel \mathcal{R}$ holds.

Definition 4.2. *A pseudosequent $\Delta \mid \Gamma \vdash_? X : A$ is solvable with respect to a constraint set $\mathcal{Q}$ (or $\mathcal{Q}$-solvable), if there is a constraint $X \approx e \in \mathcal{Q}$ and a program $M = Sol_{\mathcal{Q}}(X \approx e)$ such that the typing $\Delta \mid \Gamma \vdash M : A$ holds.*
Given a constraint set $\mathcal{Q}$ we say that a goal $\mathcal{S} \parallel \mathcal{R}$ is $\mathcal{Q}$-solvable if $\mathcal{R} \subseteq \mathcal{Q}$ and all pseudosequents in $\mathcal{S}$ are $\mathcal{Q}$-solvable.

We remark that the solution is not unique, moreover, there can be an infinite number of solutions. Thus, it should be the human agent who decides the desired solution as she is conducting the whole process. This rules out some relevant inquiries of automated proof-search like the question of the complexity of the proof-search space.

The next lemma guarantees that the solvability of goals is rearward preserved by the transition relation, this characteristic will imply the desired soundness property.

Lemma 1. *Let $\mathcal{Q}$ be a constraint set such that $\mathcal{R}_1, \mathcal{R}_2 \subseteq \mathcal{Q}$. If $\mathcal{S}_1 \parallel \mathcal{R}_1 \rhd^+ \mathcal{S}_2 \parallel \mathcal{R}_2$ and $\mathcal{S}_2 \parallel \mathcal{R}_2$ is solvable with respect to $\mathcal{Q}$ then $\mathcal{S}_1 \parallel \mathcal{R}_1$ is solvable with respect to $\mathcal{Q}$.*

Proof. Let $\mathcal{Q}$ be as required. The proof goes by induction on $\rhd^+$. In the base case we have $\mathcal{S}_1 \parallel \mathcal{R}_1 \rhd \mathcal{S}_2 \parallel \mathcal{R}_2$ and proceed by a nested induction on $\rhd$. We give some cases as example. The remaining are analogous:

— Case (`intro` x): We have $\Delta \mid \Gamma \vdash_? X : A \to B \parallel \mathcal{R} \rhd \Delta \mid \Gamma, x : A \vdash_? Y : B \parallel \mathcal{R}, X \approx \lambda x.Y$. Let us assume that $M = Sol_{\mathcal{Q}}(Y \approx N)$ with $\Delta \mid \Gamma, x : A \vdash M : B$. In this case we have $\Delta \mid \Gamma \vdash (\lambda x.Y)[Y/M] : A \to B$ by rule $(\to R)$. Noting that $\lambda x.M$ is a program, for so is M, and that $\lambda x.M \equiv (\lambda x.Y)[Y/M]$ we get that $\lambda x.M = Sol_{\mathcal{Q}}(X \approx \lambda x.Y)$. Thus the goal $\Delta \mid \Gamma \vdash_? X : A \to B \parallel \mathcal{R}$ is solvable with respect to $\mathcal{Q}$ and the case is done.

— Case (`vassert A :`): In this case we have $\Delta \mid \Gamma \vdash_? X : C \parallel \mathcal{R} \rhd \Delta \mid \cdot \vdash_? X_1 : A \ ;\ \Delta, x : A \mid \Gamma \vdash_? X_2 : C \parallel \mathcal{R}, X \approx \mathsf{elet}(X_1, x.X_2)$. Let us assume that $\Delta \mid \cdot \vdash M_1 : A$ and $\Delta, x : A \mid \Gamma \vdash M_2 : C$ where M_1, M_2 are programs such that there

are equations $Y_1 \approx N_1, Y_2 \approx N_2 \in \mathcal{Q}$ with $M_1 = Sol_{\mathcal{Q}}(Y_1 \approx N_1), M_2 = Sol_{\mathcal{Q}}(Y_1 \approx N_2)$. From the above typings we get, by the (SUBSTE) rule, that $\Delta \mid \Gamma \vdash \mathsf{elet}(M_1, x.M_2) : C$. Finally we observe that $\mathsf{elet}(M_1, x.M_2) = Sol_{\mathcal{Q}}(X \approx \mathsf{elet}(X_1, x.X_2))$. Thus the goal $\Delta \mid \Gamma \vdash_? X : C \parallel \mathcal{R}$ is solvable with respect to $\mathcal{Q}$ as desired.

— Case (seq): We have here that there exists a sequence $\mathcal{S}$ such that $\mathcal{S}_1; \mathcal{S} \parallel \mathcal{R}_1 \rhd \mathcal{S}_2; \mathcal{S} \parallel \mathcal{R}_2$ where $\mathcal{S}_1 \parallel \mathcal{R}_1 \rhd \mathcal{S}_2 \parallel \mathcal{R}_2$. Let us assume that $\mathcal{S}_2; \mathcal{S} \parallel \mathcal{R}_2$ is solvable with respect to $\mathcal{Q}$. This implies in particular that $\mathcal{S}_2 \parallel \mathcal{R}_2$ is $\mathcal{Q}$-solvable, from which the nested induction hypothesis now yields that $\mathcal{S}_1 \parallel \mathcal{R}_1$ is $\mathcal{Q}$-solvable. From this we can conclude that the sequence $\mathcal{S}_1; \mathcal{S}$ is $\mathcal{Q}$-solvable (observe that the part $\mathcal{S}$ was already $\mathcal{Q}$-solvable due to the original assumption). Thus, the goal $\mathcal{S}_1; \mathcal{S} \parallel \mathcal{R}_1$ is solvable with respect to $\mathcal{Q}$, as desired. This finishes the nested induction that proves the base case.

For the inductive step, we have that $\mathcal{S}_1 \parallel \mathcal{R}_1 \rhd^+ \mathcal{S}_2 \parallel \mathcal{R}_2$ where there is a goal $\mathcal{S}_3 \parallel \mathcal{R}_3$ such that $\mathcal{S}_1 \parallel \mathcal{R}_1 \rhd \mathcal{S}_3 \parallel \mathcal{R}_3$ and $\mathcal{S}_3 \parallel \mathcal{R}_3 \rhd^+ \mathcal{S}_2 \parallel \mathcal{R}_2$. Assuming that $\mathcal{S}_2 \parallel \mathcal{R}_2$ is solvable with respect to $\mathcal{Q}$ the I.H. yields $\mathcal{S}_3 \parallel \mathcal{R}_3$ is solvable with respect to $\mathcal{Q}$, the already proved base case now yields that $\mathcal{S}_1 \parallel \mathcal{R}_1$ is solvable with respect to $\mathcal{Q}$, as desired. □

Theorem 4.1 (Soundness of the synthesis process)**.** *Let $\Delta \mid \Gamma \vdash_? X : A$ be a synthesis problem. If $\Delta \mid \Gamma \vdash_? X : A \parallel \varnothing \rhd^+ \bullet \parallel \mathcal{R}$ then there is a program M such that $\Delta \mid \Gamma \vdash M : A$.*

Proof. It is clear that the goal $\bullet \parallel \mathcal{R}$ is solvable with respect to $\mathcal{R}$, hence, the Lemma 1 yields that $\Delta \mid \Gamma \vdash_? X : A \parallel \varnothing$ is also solvable with respect to $\mathcal{R}$. This fact ensures the existence of the desired program M. □

According to this theorem our type-based synthesis process is correct. This ends our exposition. Let us finish this paper with some remarks.

5 Final Remarks

In this paper we presented a dual-context sequent calculus $\mathcal{GS}4$ for the necessity fragment of the constructive modal logic S4, originated in our previous work [19], as a type system for lambda-terms. The modal types allow us to make a distinction between values with essentially the same functionality, namely ordinary values (inhabiting the type A) and enhanced values (inhabiting the type $\Box A$). This distinction is required by several applications.

The specific left rules for implication and necessity as well as the dual-context feature enable us to define a simple and intuitive sound bottom-up synthesis process involving a left-to-right depth-first proof-search, which, with the help of constraint-sets and the backward reading of the typing rules succeeds in returning a correct-by-construction modal lambda-term.

The procedure is human-driven, in the sense of modern interactive theorem provers, a feature that allows to define the synthesis process without technical modifications of the inference rules, unlike some proposals of automated proof-search [29, 1, 26].

Towards a more realistic programming environment we intend to extend the current approach to a dual-context sequent calculus for the full modal logic S4, related to our work in [12]. Another important task is to extend the here presented results to the classical version of S4. This requires a handling of classical negation suitable for proof-search, in particular the use of a traditional multi conclusion sequent calculus is not convenient.

Some other programming language features, like a detailed study of the operational semantics and the extension of $\mathcal{GS}4$ with recursion and memory references in the lines of [22, 8], have to be integrated in this quest.

Another important research topic consists of mechanizing[5] the current results, following our previous work [11]. With respect to other approaches of type-based synthesis in modal logic, we consider important to relate our approach with

[5] This is the main reason for defining contexts as lists instead of sets or multisets.

those involving intersection types, like [14, 9]. This would complicate the constraints, due to the fact that the typing rules for intersection are obviously not syntax-directed. This happens also in other richer type systems, for instance [5] involving some kind of polymorphism.

In some cases the constraints might be unsolvable, for example if the constraint-sets are cyclic or contain a recursive constraint. But even if they are solvable, their solutions would certainly require more powerful tools, such as (higher-order) unification.

Acknowledgments

This research is being supported by DGAPA-PAPIIT UNAM grant IN119920.

References

1. **Andrikonis, J. (2012).** Loop-free calculus for modal logic s4. i. Lithuanian Mathematical Journal, Vol. 52, No. 1, pp. 1–12.
2. **Avron, A., Honsell, F., Miculan, M., Paravano, C. (1998).** Encoding Modal Logics in Logical Frameworks. Studia Logica, Vol. 60, No. 1, pp. 161–208.
3. **Barber, A. G., Plotkin, G. (1997).** Dual intuitionistic linear logic. Technical Report LFCS-96-347, University of Edinburgh.
4. **Basin, D., Deville, Y., Flener, P., Hamfelt, A., Fischer Nilsson, J. (2004).** Synthesis of programs in computational logic. In **Bruynooghe, M., Lau, K.-K.**, editors, Program Development in Computational Logic: A Decade of Research Advances in Logic-Based Program Development. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 30–65.
5. **Bessai, J., Dudenhefner, A., Düdder, B., Chen, T.-C., de'Liguoro, U., Rehof, J. (2015).** Mixin Composition Synthesis Based on Intersection Types. **Altenkirch, T.**, editor, 13th International Conference on Typed Lambda Calculi and Applications (TLCA 2015), volume 38 of Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 76–91.
6. **Borghuis, T., Feijs, L. (2000).** A constructive logic for services and information flow in computer networks. The Computer Journal, Vol. 43, No. 4, pp. 274–289.
7. The Coq Development Team **(2020).** The Coq Proof Assistant Reference Manual Version 8.11.
8. **Davies, R., Pfenning, F. (2001).** A modal analysis of staged computation. J. ACM, Vol. 48, No. 3, pp. 555–604.
9. **Düdder, B., Martens, M., Rehof, J. (2014).** Staged composition synthesis. **Shao, Z.**, editor, Programming Languages and Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 67–86.
10. **Fitting, M. C. (1983).** Proof Methods for Modal and Intuitionistic Logics. Synthese Library. Springer Netherlands.
11. **González-Huesca, L., Miranda-Perea, F. E., Linares-Arévalo, P. S. (2019).** Axiomatic and dual systems for constructive necessity, a formally verified equivalence. Journal of Applied Non-Classical Logics, Vol. 29, No. 3, pp. 255–287.
12. **González Huesca, L. d. C., Miranda-Perea, F. E., Linares-Arévalo, P. S. (2020).** Dual and Axiomatic Systems for Constructive S4, a Formally Verified Equivalence. Electronic Notes in Theoretical Computer Science, Vol. 348, pp. 61 – 83. 14th International Workshop on Logical and Semantic Frameworks, with Applications (LSFA 2019).
13. **Heilala, S., Pientka, B. (2007).** Bidirectional decision procedures for the intuitionistic propositional modal logic is4. Proceedings of the 21st international conference on Automated Deduction: Automated Deduction, CADE-21, Springer-Verlag, Berlin, Heidelberg, pp. 116–131.
14. **Henglein, F., Rehof, J. (2016).** Modal intersection types, two-level languages, and staged synthesis. In **Probst, C., Hankin, C., Hansen, R.**, editors, Semantics, logics, and calculi, Lecture notes in computer science. Springer, pp. 289–312. Nielsons' Festschrift.
15. **Hudelmaier, J. (1996).** A contraction-free sequent calculus for s4. In **Wansing, H.**, editor, Proof Theory of Modal Logic, Applied Logic Series. Springer Netherlands, pp. 3–15.
16. **Kavvos, G. A. (2017).** Dual-context calculi for modal logic. 32nd Annual ACM/IEEE Symposium

on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017, pp. 1–12.

17. **Martin-Löf, P. (1996).** On the meanings of the logical constants and the justifications of the logical laws. Nordic J. Philos. Logic, Vol. 1, No. 1, pp. 11–60.
18. **Mints, G., Orevkov, V., Tammet, T. (1996).** Transfer of sequent calculus strategies to resolution for s4. In **Wansing, H.**, editor, Proof Theory of Modal Logic, Applied Logic Series. Springer Netherlands, pp. 17–31.
19. **Miranda-Perea, F. E., del Carmen González Huesca, L., Linares-Arévalo, P. S. (2020).** On interactive proof-search for constructive modal necessity. Electronic Notes in Theoretical Computer Science, Vol. 354, pp. 107 – 127. Proceedings of the Eleventh and Twelfth Latin American Workshop on Logic/Languages, Algorithms and New Methods of Reasoning (LANMR).
20. **Miranda-Perea, F. E., Linares-Arévalo, P. S., Aliseda-Llera, A. (2015).** How to prove it in natural deduction: A tactical approach. CoRR, Vol. abs/1507.03678.
21. **Miyamoto, K., Igarashi, A. (2004).** A modal foundation for secure information flow. **Sabelfeld, A.**, editor, Workshop on Foundations of Computer Security, pp. 187–203.
22. **Moody, J. (2004).** Logical mobility and locality types. **S., E.**, editor, Proceedings of the 14th International Conference on Logic Based Program Synthesis and Transformation. LOPSTR 2004, volume 3573 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, Heidelberg, pp. 69–84.
23. **Murphy VII, T., Crary, K., Harper, R., Pfenning, F. (2004).** A symmetric modal lambda calculus for distributed computing. Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, LICS '04, IEEE Computer Society, Washington, DC, USA, pp. 286–295.
24. **Pfenning, F., Davies, R. (2001).** A judgmental reconstruction of modal logic. Mathematical Structures in Comp. Sci., Vol. 11, No. 4, pp. 511–540.
25. **Pfenning, F., Wong, H. (1995).** On a modal lambda calculus for S4. **Brookes, S. D., Main, M. G., Melton, A., Mislove, M. W.**, editors, Eleventh Annual Conference on Mathematical Foundations of Programming Semantics, MFPS 1995, Tulane University, New Orleans, LA, USA, March 29 - April 1, 1995, volume 1 of Electronic Notes in Theoretical Computer Science, Elsevier, pp. 515–534.
26. **Pliuškevičius, R., Pliuškevičienė, A. (2008).** A new method to obtain termination in backward proof search for modal logic s4. Journal of Logic and Computation, Vol. 20, No. 1, pp. 353–379.
27. **Poggiolesi, F. (2010).** Gentzen Calculi for Modal Propositional Logic. Trends in Logic. Springer Netherlands.
28. **Schroeder-Heister, P. (2011).** Implications-as-Rules vs. Implications-as-Links: An Alternative Implication-Left Schema for the Sequent Calculus. J. Philosophical Logic, Vol. 40, No. 1, pp. 95–101.
29. **Stone, M. (2005).** Disjunction and modular goal-directed proof search. ACM Trans. Comput. Logic, Vol. 6, No. 3, pp. 539–577.

Article received on 09/10/2020; accepted on 11/02/2021.
Corresponding author is Favio E. Miranda-Perea.

ISSN 2007-9737

An Algebraic Study of the First Order Version of some Implicational Fragments of Three-Valued Łukasiewicz Logic

Aldo Figallo-Orellano[1], Juan Sebastián Slagter[2]

[1,2]Universidad Nacional del Sur (UNS),
Departamento de Matemática,
Argentina

[1]University of Campinas (UNICAMP),
Centre for Logic, Epistemology and The History of Science (CLE),
Brazil

aldofigallo@gmail.com, juan.slagter@uns.edu.ar

Abstract. In this paper, some implicational fragments of trivalent Łukasiewicz logic are studied and the propositional and first-order logic are presented. The maximal consistent theories are studied as Monteiro's maximal deductive systems of the Lindenbaum-Tarski algebra in both cases. Consequently, the adequacy theorems with respect to the suitable algebraic structures are proven.

Keywords. Trivalent Hilbert algebras, modals operators, 3-valued Gödel logic, first-order logics.

1 Introduction and Preliminaries

In 1923, Hilbert proposed studying the implicative fragment of *classical propositional calculus*. This fragment is well-known as *positive implicative propositional calculus* and its study was started by Hilbert and Bernays in 1934. The following axiom schemas define this calculus:

(E1) $\alpha \to (\beta \to \alpha)$,

(E2) $(\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma))$,

and the inference rule *modus ponens* is:

(MP) $\dfrac{\alpha, \alpha \to \beta}{\beta}$.

In 1950, Henkin introduced the *implicative models* as algebraic models of the positive implicative calculus. Later, A. Monteiro renamed them as *Hilbert algebras* and his Ph. D. student Diego ([8]) made one of the most important contributions to these algebraic structures.

In particular, this author proved that the class of Hilbert algebras is an equational class, that is to say, it is possible to characterize the class via certain equations.

Moreover, Diego proved that the positive implicative propositional calculus is decidable by means of using algebraic technical tools.

On the other hand, Thomas in [26] considered the n-valued positive implicative calculus, with signature $\{\to, 1\}$, as a calculus that has a characteristic matrix $\langle A, \{1\}\rangle$ where $\{1\}$ is the set of designated elements and the algebra $A = (\mathbb{C}_n, \to, 1)$ is defined as follows:

$$\mathbb{C}_n = \{0, \frac{1}{n}, \frac{2}{n}, \cdots, \frac{n-1}{n}, 1\},$$

and

$$x \to y = \begin{cases} 1 & \text{if } x \leq y \\ y & y < x \end{cases}.$$

This author proved that for this calculus we must add the following axiom to the positive implicative calculus:

(E3) $T_n(\alpha_0, \cdots, \alpha_{n-1}) = \beta_{n-2} \to (\beta_{n-3} \to (\cdots \to (\beta_0 \to \alpha_0) \cdots))$, where

$\beta_i = (\alpha_i \to \alpha_{i+1}) \to \alpha_0$ for all $i, 0 \leq i \leq n-2$.

Table 1

$\to$	0	$\frac{1}{2}$	1
0	1	1	1
$\frac{1}{2}$	0	1	1
1	0	$\frac{1}{2}$	1

The algebraic counterpart of n-valued positive implicative calculus was studied in [15] where the axiom (E3) is translated by the equation $T_n = 1$ to the ones of Hilbert algebras. In particular, in the $n = 3$ case, the variety is generated by an algebra that has this set $\mathbb{C}_3 = \{0, \frac{1}{2}, 1\}$ as support and an implication $\to$ defined by the following table 1.

It is clear that 3-valued Hilbert algebras are Hilbert algebras that verify the following identity:

(IT3) $((x \to y) \to z) \to (((z \to x) \to z) \to z) = 1$.

It is important to note that the implication defined in Table 1 characterizes the implication of 3-valued Gödel logic that we call G3.

Paraconsistent extensions of 3-valued Gödel logic were studied as a tool for knowledge representation and nonmonotonic reasoning, [21, 20]. Particularly, Osorio and his collaborators showed that some of these logics can be used to express interesting nonmonotonic semantics. In addition, these paraconsistent systems were also studied under a mathematical logic point of view as we can see in the following papers: [22, 12, 17, 19, 18]. To see other applications of three-valued logic to other fields the reader can consult [5].

In this paper, we will study implicative fragments of G3 enriched with certain modal operators that we call Moisil's operators. In this setting, recall that Moisil introduced 3-valued Łukasiewicz algebras (or 3-valued Łukasiewicz-Moisil algebras) as algebraic models of 3-valued logic proposed by Łukasiewicz. It is well-known, and part of folklore, that the class of 3-valued Łukasiewicz algebras is term equivalent to the one of 3-valued MV-algebras (see, for instance, [2]). Recall that an algebra $(A, \wedge, \vee, \sim, \nabla, 0, 1)$ is a 3-valued Łukasiewicz algebra if the following conditions hold: (L0) $x \vee 1 = 1$, (L1) $x \wedge (x \vee y) = x$, (L2) $x \wedge (y \vee z) = (z \wedge x) \vee (y \wedge x)$, (L3) $\sim\sim x = x$, (L4) $\sim (x \wedge y) = \sim x \vee \sim y$, (L5) $\sim x \vee \nabla x = 1$, (L6) $\sim x \wedge x = \sim x \wedge \nabla x$, and (L7) $\nabla(x \wedge y) = \nabla x \wedge \nabla y$. It is well known that each 3-valued Łukasiewicz algebra is a De Morgan algebra because equations (L0) to (L4) hold, [2, Definition 2.6]. In general, to see more technical aspects of Łukasiewicz-Moisil algebras, the reader can consult [2].

On the other hand, the characteristic matrix of logic from trivalent Łukasiewicz algebras has the operators $\wedge$, $\vee$, $\sim$, ∇ (possibility operator) and $\triangle$ (necessity operator) over the chain $\mathbb{C}_3 = \{0, \frac{1}{2}, 1\}$, and they are defined by the next table:

Table 2

x	$\sim x$	∇x	$\triangle x$
0	1	0	0
$\frac{1}{2}$	$\frac{1}{2}$	1	0
1	0	1	1

In addition, the implication $\to$ defined in Table 1 can be obtained from the operators $\wedge$, $\vee$, $\sim$, ∇ and $\triangle$ by the following formula:

$$x \to y = \triangle \sim x \vee y \vee (\nabla \sim x \wedge \nabla y).$$

Moreover, it is not hard to see that $\nabla x = (x \to \triangle x) \to \triangle x$. In this setting, the algebraic structures in the signature $\{\to, \triangle\}$ were defined and studied by Canals-Frau and Figallo in [6, 7]; these structures can be seen as certain $\{\to, \triangle\}$-fragments of 3-valued Łukasiewicz algebras.

The rest of the paper is organized as follows: in section 2, we introduce and study the class of modal 3-valued Hilbert algebras with supremum and also, as an application of our algebraic work, we present a Hilbert calculus for the fragment with disjunction soundness and completeness, in a strong version, with respect to this class of algebras. In Section 3, we study the first-order logic for the fragment with disjunction by means of an adaptation of the Rasiowa's technique ([25]) using our algebraic work for the propositional case. In the last Section, we discuss the possibility to applied our proofs to other classes of algebras.

ISSN 2007-9737

2 Trivalent Modal Hilbert Algebras With Supremum

In this section, we will introduce and study algebraically the trivalent modal Hilbert algebra with supremum that we denote $H_3^{\vee,\triangle}$-algebras. From this algebraic work, we present a sound and complete calculus w.r.t. the class of $H_3^{\vee,\triangle}$-algebras in propositional case.

For the sake of brevity, we only introduce those essential notions of *Hilbert algebras* that we need, thought not in full detail. Anyway, for more information about these algebras, the reader can consult the bibliography.

Now, recall that a Hilbert algebra is an algebra $(A, \to, 1)$ such that for all $x, y, z \in A$ verifies:

(H1) $x \to (y \to x) = 1$,

(H2) $(x \to (y \to z)) \to ((x \to y) \to (x \to z)) = 1$,

(H3) if $x \to y = 1$, $y \to x = 1$, then $x = y$.

Furthermore, we say $(A, \to, 1)$ is a 3-valued Hilbert algebra if verifies the following equation:
(IT3) $((x \to y) \to z) \to (((z \to x) \to z) \to z) = 1$.

The following lemma is well-known and the proof can be found in [8].

Lemma 2.1 *Let A be a Hilbert algebra. The following properties are satisfied for every $x, y, z \in A$:*

(H4) *if $x = 1$ and $x \to y = 1$, then $y = 1$;*

(H5) *the relation $\leq$ defined by $x \leq y$ iff $x \to y = 1$, which is an order relation on A and 1 is the last element;*

(H6) $x \to x = 1$;

(H7) $x \leq y \to x$;

(H8) $x \to (y \to z) \leq (x \to y) \to (x \to z)$;

(H9) $x \to 1 = 1$;

(H10) $x \leq y$ *implies* $z \to x \leq z \to y$;

(H11) $x \leq y \to z$ *implies* $y \leq x \to z$;

(H12) $x \to ((x \to y) \to y) = 1$,

(H13) $1 \to x = x$;

(H14) $x \leq y$ *implies* $y \to z \leq x \to z$,

(H15) $x \to (y \to z) = y \to (x \to z)$;

(H16) $x \to (x \to y) = x \to y$;

(H17) $(x \to y) \to ((y \to x) \to x) = (y \to x) \to ((x \to y) \to y)$;

(H18) $x \to (y \to z) = (x \to y) \to (x \to z)$;

(H19) $((x \to y) \to y) \to y = x \to y$.

In the following, we present a definition of the equational class of 3-valued modal Hilbert algebra that was introduced in [6].

Definition 2.2 *An algebra $(A, \to, \triangle, 1)$ is said to be a 3-valued modal Hilbert algebra if its reduct $(A, \to, 1)$ is a 3-valued Hilbert algebra and $\triangle$ verifies the following identities:*

(M1) $\triangle x \to x = 1$,

(M2) $((y \to \triangle y) \to (x \to \triangle\triangle x)) \to \triangle(x \to y) = \triangle x \to \triangle\triangle y$, *and*

(M3) $(\triangle x \to \triangle y) \to \triangle x = \triangle x$.

Moreover, we define a new connective by $\nabla x = (x \to \triangle x) \to \triangle x$.

Now, consider the following Definition that we introduce for the first time.

Definition 2.3 *An algebra $\mathbf{A} = \langle A, \to, \vee, \triangle, 1\rangle$ is said to be a trivalent modal Hilbert algebra with supremum if the following properties hold:*

(1) *the reduct $\langle A, \vee, 1\rangle$ is a join-semilattice with greatest element 1, and the conditions* (a) $x \to (x \vee y) = 1$ *and* (b) $(x \to y) \to ((x \vee y) \to y) = 1$ *hold. Besides, given $x, y \in A$ such that there exists the infimum of $\{x, y\}$, denoted by $x \wedge y$, then $\triangle(x \wedge y) = \triangle x \wedge \triangle y$.*

(2) *The reduct $\langle A, \to, \triangle, 1\rangle$ is a $\triangle H_3$-algebra.*

From now on, we denote with $\mathbf{A}$ the $H_3^{\vee,\triangle}$-algebra $\langle A, \to, \vee, \triangle, 1\rangle$ and with A its support. Next, we will show some properties that will be very useful for the rest of this section.

Let us notice that there is an $H_3^{\vee,\triangle}$-algebra A in which the infimum can not be defined. To see that, take some subalgebras of $\mathbb{C}_3^{\to,\vee} \times \mathbb{C}_3^{\to,\vee}$ where $\times$ is the direct product.

The fragment with infimum has been studied in [24] that will comment in the following Remark.

Remark 2.4 *In [24], the class of 3-valued modal Hilbert algebra with imfimum ($i\triangle H_3$-algebra) was defined as follows: An algebra $\langle A, \to, \wedge, \triangle, 1\rangle$ is said to be an $i\triangle H_3$-algebra if the following conditions hold: (1) the reduct $\langle A, \to, \triangle, 1\rangle$ is a 3-valued modal Hilbert algebra; (2) the following identities hold: (iH_1) $x \wedge (y \wedge z) = (x \wedge y) \wedge z$, (iH_2) $x \wedge x = x$, (iH_3) $x \wedge (x \to y) = x \wedge y$, and (iH_4) $(x \to (y \wedge z)) \to ((x \to z) \wedge (x \to y)) = 1$.*

Let us observe that for each $i\triangle H_3$-algebra $\mathbf{A}$ and for every $x, y \in A$, we can define the supremum of $\{x, y\}$ in the following way:

$$x \vee y \stackrel{def}{=} ((x \to y) \to y) \wedge ((y \to x) \to x).$$

Indeed, let $a, b \in A$ and put $c = ((a \to b) \to b) \wedge ((b \to a) \to a)$. Since $x \leq (x \to y) \to y$ and $x \leq (y \to x) \to x$ hold and there exists the infimum $((x \to y) \to y) \wedge ((y \to x) \to x)$, then c is upper bound of the set $\{a, b\}$. Now, let us suppose that d is another upper bound of $\{a, b\}$ such that $c \not\leq d$. Thus, there exists an irreducible deductive system P such that $c \in P$ and $d \notin P$ [8, Corolario 1]. Besides, since $a, b \leq d$ then $a, b \notin P$. On the other hand, as A is a trivalent Hilbert algebra and according to [14, Théorème 4.1], we have $a \to b \in P$ or $b \to a \in P$. Now, if we suppose that $a \to b \in P$ and since $c \leq (b \to a) \to a$, then we can infer that $a \in P$, which is a contradiction. If we consider the case $b \to a \in P$, we also obtain a contradiction. Thus, c is the supremum of $\{a, b\}$. Therefore, each $i\triangle H_3$-algebra is a relatively pseudocomplemented lattice *since $x \wedge z \leq y$ iff $x \leq z \to y$, see [25]. From the latter, we have that each $i\triangle H_3$-algebra is a distributive lattice. It is possible to see that every finite and complete $i\triangle H_3$-algebra is a 3-valued Łukasiewicz algebra. To see the details, the reader can consult Section 3 of [24].*

Lemma 2.5 *For a given $H_3^{\vee,\triangle}$-algebra $\mathbf{A}$ and $x, y, z \in A$, then the following properties hold:*

(1) $\triangle 1 = 1$;

(2) $\triangle(x \to y) \to (\triangle x \to \triangle y) = 1$;

(3) *if* $x \to y = 1$, *then* $x \vee y = y$;

(4) *if* $x \to z = 1$ *and* $y \to z = 1$, *then* $(z \vee y) \to z = 1$;

(5) $x \to (x \vee y) = 1$,

(6) $(x \to z) \to ((y \to z) \to ((x \vee y) \to z)) = 1$;

(7) $\triangle(x \vee y) = \triangle x \vee \triangle y$;

(8) $\nabla(x \vee y) = \nabla x \vee \nabla y$.

Proof. It is routine. □

Definition 2.6 *For a given $H_3^{\vee,\triangle}$-algebra $\mathbf{A}$ and $D \subseteq A$. Then, D is said to be a deductive system if (D1) $1 \in D$, and (D2) if $x, x \to y \in D$ imply $y \in D$. Additionally, we say that D is a modal if: (D3) $x \in D$ implies $\triangle x \in D$. Moreover, D is said to be maximal if for every modal deductive system M such that $D \subseteq M$ implies $M = A$ or $M = D$.*

Given a $H_3^{\vee,\triangle}$-algebra-algebra $\mathbf{A}$ and $\{H_i\}_{i\in I}$ a family of modal deductive systems of A, then it is easy to see that $\bigcap_{i\in I} H_i$ is a modal deductive system. Thus, we can consider the notion of modal deductive system generated by H, denoted $[H)_m$, as an intersection of all modal deductive system D such that $D \subseteq H$. The deductive system generated by H, denoted $[H)$, verify that $[H) = \{x \in A : \text{there exist } h_1, \cdots, h_k \in H \text{ such as } h_1 \to (h_2 \to \cdots \to (h_k \to x) \cdots) = 1\}$ where k is a finite integer, see [8]. Now, we will introduce the following notation:

$$(x_1, \ldots, x_{n-1}; x_n) = \begin{cases} x_n, & if\ n = 1, \\ x_1 \to (x_2, \ldots, x_{n-1}; x_n), & if\ n > 1. \end{cases}$$

Hence, we can write:

$$[H) = \{x \in A : \text{there exist } h_1, \ldots, h_k \in D_1 : (h_1, \ldots, h_k; x) = 1\}.$$

ISSN 2007-9737

Then, we have the following result:

Proposition 2.7 *Let* $\mathbf{A}$ *be a* $H_3^{\vee,\triangle}$*-algebra, suppose that* $H \subseteq A$ *and* $a \in A$*. Then the following properties hold:*

(i) $[H)_m = \{x \in A : \text{there exist } h_1, \cdots, h_k \in H : (\triangle h_1, \ldots, \triangle h_k; x) = 1\}$;

(ii) $[a)_m = [\triangle a)$, *where* $[b)$ *is the set* $[\{b\})$;

(iii) $[H \cup \{a\})_m = \{x \in A : \triangle a \to x \in [H)_m\}$.

Proof. It is routine. □

Lemma 2.8 *Given a* $H_3^{\vee,\triangle}$*-algebra* $\mathbf{A}$*, there exists a lattice-isomorphism between the poset of congruences of* A *and the poset of the modal deductive systems of* A*.*

Proof. It is well-known that the set of congruences of Hilbert algebra A is lattice-isomorphic to the set of all deductive systems. For each deductive system D we have the relation $R(D) = \{(x, y) : x \to y, y \to x \in D\}$ which is a congruence of A, such that the class of 1 verifies $|1|_{R(D)} = D$. In addition, for each congruence θ of A, the class of $|1|_\theta$ is a deductive system and $R(|1|_\theta) = \theta$. From the latter and Lemma 2.5 (1) and (2), we can infer that every congruence θ for a given A respect $\triangle$ and $|1|_\theta$ is a modal deductive system. □

For each $H_3^{\vee,\triangle}$-algebra $\mathbf{A}$, we can define a new binary operation $\rightarrowtail$ named weak implication such that: $x \rightarrowtail y = \triangle x \to y$.

Lemma 2.9 *Let* $\mathbf{A}$ *be a* $H_3^{\vee,\triangle}$*-algebra, for any* $x, y, z \in A$ *the following properties hold:*

(wi1) $1 \rightarrowtail x = x$;

(wi2) $x \rightarrowtail x = 1$;

(wi3) $x \rightarrowtail \triangle x = 1$;

(wi4) $x \rightarrowtail (y \rightarrowtail z) = (x \rightarrowtail y) \rightarrowtail (x \rightarrowtail z)$;

(wi5) $x \rightarrowtail (y \rightarrowtail x) = 1$;

(wi6) $((x \rightarrowtail y) \rightarrowtail x) \rightarrowtail x = 1$.

Proof. The proof immediately follows from the very definitions; and, it can be consulted [24, Lema 2.4.2]. □

Let $\mathbf{A}$ an $H_3^{\vee,\triangle}$-algebra and suppose a subset $D \subseteq A$, we say that D is a weak deductive system (w.d.s.) if $1 \in D$, and $x, x \rightarrowtail y \in D$ imply $y \in D$. It is not hard to see that the set of modal deductive systems is equal to the set of weak deductive systems. We denote by $\mathcal{D}_w(A)$ the set of weak deductive systems of a Hilbert algebra.

Now, for a given $H_3^{\vee,\triangle}$-algebra $\mathbf{A}$ and a (weak) deductive system D of A, D is said to be a maximal if for every (weak) deductive system M such that $D \subseteq M$, then $M = A$ or $M = D$. Besides, let us consider the set of all maximal w.d.s. $\mathcal{E}_w(A)$. A. Monteiro gave the following definition in order to characterize maximal deductive systems:

Definition 2.10 *(A. Monteiro) Let* $\mathbf{A}$ *be a* $H_3^{\vee,\triangle}$*-algebra,* $D \in \mathcal{D}_w(A)$ *and* $p \in A$*. We say that* D *is a weak deductive system tied to* p *if* $p \notin D$ *and for any* $D' \in \mathcal{D}_w(A)$ *such that* $D \subsetneq D'$*, then* $p \in D'$*.*

The importance for introducing the notion of weak deductive systems is to prove that every maximal weak deductive system is a weak deductive system tied to some element of a given $H_3^{\vee,\triangle}$-algebra, A. Conversely, and using (wi6), we can prove every w.d.s is a maximal weak deductive systems. Moreover, from (wi4), (wi5) and (wi1) and using A. Monteiro's techniques, we also can prove that $\{1\} = \bigcap_{M \in \mathcal{E}_w(A)} M$. To see the details of the proof, see Sections 2.4, 2.5 and 2.6 of [24].

In what follows, we will consider the quotient algebra $\mathbf{A}/M$ defined by $a \equiv_M b$ iff $a \to b, b \to a \in M$ and the canonical projection $q_M : \mathbf{A} \to \mathbf{A}/M$ defined by $q_M = |x|_M$ where $|x|_M$ denotes the equivalence class of x generated by M.

Lemma 2.11 *Let* $\mathbf{A}$ *be a* $H_3^{\vee,\triangle}$*-algebra. Then, the map* $\Phi : \mathbf{A} \longrightarrow \prod_{M \in \mathcal{E}_w(A)} A/M$ *defined by* $\Phi(x)(M) = q_M(x)$ *is a homomorphism; that is to say, the variety of* $H_3^{\vee,\triangle}$*-algebras is semisimple.*

Proof. Taking $\prod_{\alpha\in\mathcal{E}_w(A)} A/M_\alpha = \{f : \mathcal{A} \to \bigcup_{\alpha\in\mathcal{E}_w(A)} A/M_\alpha : f(\alpha) \in A/M_\alpha \text{ for every } \alpha \in \mathcal{E}_w(A)\}$ and $\mathcal{E}_w(A)$ is the set of maximal w.d.s. defined before. Let us define $\Phi : A \to \prod_{\alpha\in\mathcal{E}_w(A)} A/M_\alpha$ such that for every α we have that $\Phi(\alpha) = f_a$ where $f_a(\alpha) = q_\alpha(a) = |a|_\alpha \in A/M_\alpha$ with $a \in A$. It is not hard to see that Φ is a homomorphism in view of the fact that $\equiv_{M_\alpha}$ is a congruence relation. Now, from the fact that $\{1\} = \bigcap_{M\in\mathcal{E}_w(A)} M$, it is possible to see that Φ is one-to-one function which completes the proof. □

The construction of the following homomorphism is fundamental to obtaining the generating algebras of the variety of $H_3^{\vee,\triangle}$-algebra. Moreover, this homomorphism will play a central *role* in the adequacy theorems in a propositional and first-order version of logic, as we will see later on.

In the next, we consider the algebras $\mathbb{C}_3^{\to,\vee} = \langle\{0, \frac{1}{2}, 1\}, \to, \vee, \triangle, 1\rangle$ and $\mathbb{C}_2^{\to,\vee} = \langle\{0, 1\}, \to, \vee, \triangle, 1\rangle$. We denote $\mathbb{C}_3$ and $\mathbb{C}_2$ the support of $\mathbb{C}_3^{\to,\vee}$ and $\mathbb{C}_2^{\to,\vee}$, respectively; besides, the operation $\vee$ is the maximum on the corresponding chain.

Theorem 2.12 *Let M be a non-trivial maximal modal deductive system of an $H_3^{\vee,\triangle}$-algebra $\mathbf{A}$. Let us consider the sets $M_0 = \{x \in A : \nabla x \notin M\}$ and $M_{1/2} = \{x \in A : x \notin M, \nabla x \in M\}$, and the map $h : A \longrightarrow \mathbb{C}_3$ defined by*

$$h(x) = \begin{cases} 0 & \text{if } x \in M_0 \\ 1/2 & \text{if } x \in M_{1/2} \\ 1 & \text{if} x \in M. \end{cases}$$

Then, h is a homomorphism from $\mathbf{A}$ into $\mathbb{C}_3^{\to,\vee}$ such that $h^{-1}(\{1\}) = M$.

Proof. We shall prove only that $h(x \vee y) = h(x) \vee h(y)$, for the rest of the proof can be done in a similar manner.

(1) Let $x \in M$ and $y \in A$. Taking into account (5) of Lemma 2.5, we have that $x \to (x \vee y) = 1$. Thus, from D_1) and D_2) then $x \vee y \in M$.

(3) Let us consider $x, y \in M_0$ and suppose that $\nabla(x \vee y) \in M$, then by (8) of Lemma 2.5, we have that $\nabla x \vee \nabla y \in M$. Thus, according to (6) of Lemma 2.5, we infer that $(\nabla x \to \nabla x) \to ((\nabla y \to \nabla x) \to ((\nabla x \vee \nabla y) \to \nabla x)) = 1$. So, from D_1), D_2) and (H6), we can obtain that $(\nabla y \to \nabla x) \to ((\nabla x \vee \nabla y) \to \nabla x) \in M$. Since $\nabla x \notin M$, we can infer that $\triangle\nabla y \to \nabla x \in M$ and so, we have $\nabla y \to \nabla x \in M$. Form the latter and D_2), we can write $(\nabla x \vee \nabla y) \to \nabla x \in M$. Therefore, $\nabla x \in M$ which is impossible, then $\nabla(x \vee y) \notin M$.

(4) If $x \in M_0$ and $y \in M_{1/2}$, since $\nabla y \to (\nabla x \vee \nabla y) = 1$ and $\nabla y \in M$, we can infer that $\nabla x \vee \nabla y \in M$. Now, let us suppose that $x \vee y \in M$. From (6) of Lemma 2.5, we can write $(x \to y) \to ((y \to y) \to ((x \vee y) \to y)) = 1$. Thus, $x \to y \in M$ and then, $y \in M$ which is a contradiction. Therefore, $x \vee y \in M_{1/2}$.

(5) If $x \in M_{1/2}$ and $y \in M_0$, we can prove that $x \vee y \in M_{1/2}$ in a similar way to (4).

(6) Suppose that $x \in M_{1/2}$ and $y \in M_{1/2}$, then from (8) of Lemma 2.5, we have that $\nabla(x \vee y) \in M$. On the other hand, let us suppose $x \vee y \in M$, thus by (6) of Lemma 2.5, we infer that $(x \to x) \to ((x \to y) \to ((x \vee y) \to x)) = 1$. Hence, since $x \to y \in M$, we can write $x \in M$ which is a contradiction. Therefore, $x \vee y \in M_{1/2}$.

□

According to Lemma 2.11 and Theorem 2.12, and well-known facts about universal algebra, we have proved the following Corollary.

Corollary 2.13 *The variety of $H_3^{\vee,\triangle}$-algebras is semisimple. Moreover, the algebras:*

$$\mathbb{C}_3^{\to,\vee} = \langle\{0, \frac{1}{2}, 1\}, \to, \vee, \triangle, 1\rangle,$$

and

$$\mathbb{C}_2^{\to,\vee} = \langle\{0, 1\}, \to, \vee, \triangle, 1\rangle.$$

are the unique simple algebras.

ISSN 2007-9737

2.1 Propositional Calculus for $H_3^{\vee,\triangle}$-Algebras

Let $\mathfrak{Fm}_s = \langle Fm, \vee, \to, \triangle\rangle$ be the absolutely free algebra over $\Sigma = \{\to, \vee, \triangle\}$ generated by a set $Var = \{p_1, p_2, \cdots\}$ of numerable variables. As usual, we say that $\mathfrak{Fm}_s$ is a language over Var and Σ. Consider now the following logic:

Definition 2.14 *We denote by $\mathcal{H}^3_{\vee,\triangle}$ the Hilbert calculus determined by the following axioms and inference rules, where $\alpha, \beta, \gamma, ... \in Fm$:*

Axiom schemas

(Ax1) $\alpha \to (\beta \to \alpha)$,

(Ax2) $(\alpha \to (\beta \to \gamma) \to ((\alpha \to \beta) \to (\alpha \to \gamma))$,

(Ax3) $((\alpha \to (\beta \to \gamma)) \to (((\gamma \to \alpha) \to \gamma) \to \gamma)$,

(Ax4) $\alpha \to (\alpha \vee \beta)$,

(Ax5) $\beta \to (\alpha \vee \beta)$,

(Ax6) $(\alpha \to \gamma) \to ((\beta \to \gamma) \to ((\alpha \vee \beta) \to \gamma))$,

(Ax7) $\triangle\alpha \to \alpha$,

(Ax8) $\triangle(\triangle\alpha \to \beta) \to (\triangle\alpha \to \triangle\beta)$,

(Ax9) $((\beta \to \triangle\beta) \to (\alpha \to \triangle(\alpha \to \beta))) \to \triangle(\alpha \to \beta)$,

(Ax10) $((\triangle\alpha \to \beta) \to \gamma) \to ((\triangle\alpha \to \gamma) \to \gamma)$.

Inference Rules

(MP) $\dfrac{\alpha, \alpha \to \beta}{\beta}$, (NEC) $\dfrac{\alpha}{\triangle\alpha}$.

Assume that $\nabla\alpha := (\alpha \to \triangle\alpha) \to \triangle\alpha$.

Let $\Gamma \cup \{\alpha\}$ be a set formulas of $\mathcal{H}^3_{\vee,\triangle}$, we define the derivation of α from Γ in usual a way and denote it by $\Gamma \vdash_\vee \alpha$.

Lemma 2.15 *The following rules are derivable in $\mathcal{H}^3_{\vee,\triangle}$:*

(P_s1) $\vdash_\vee (x \vee y) \to (y \vee x)$;

(P_s2) $\{x \to y\} \vdash_\vee (x \vee z) \to (y \vee z)$;

(P_s3) $\{x \to y, u \to v\} \vdash_\vee (x \vee u) \to (y \vee v)$;

$(R_\vee 3)$ $\dfrac{\alpha \to \beta}{(\alpha \vee \beta) \to \beta}$.

Proof. It is routine. □

Now, we denote by $\alpha \equiv_\vee \beta$ if conditions $\vdash_\vee \alpha \to \beta$ and $\vdash_\vee \beta \to \alpha$ hold. Then,

Lemma 2.16 *$\equiv_\vee$ is a congruence on $\mathfrak{Fm}_s$.*

Proof. We only have to prove that if $\alpha \equiv_\vee \beta$ and $\gamma \equiv_\vee \delta$, then $\alpha \vee \gamma \equiv_\vee \beta \vee \delta$, which follows immediately from (P_s3). □

Since the $\equiv_\vee$ is a congruence, it allows us to define the quotient algebra $\mathfrak{Fm}_s/\equiv_\vee$ that is so-called the Lindenbaum-Tarski algebra.

Theorem 2.17 *The algebra $\mathfrak{Fm}_s/\equiv_\vee$ is a $H_3^{\vee,\triangle}$-algebra by defining: $|\alpha| \to |\beta| = |\alpha \to \beta|$, $|\alpha| \vee |\beta| = |\alpha \vee \beta|$ and $1 = |\beta \to \beta| = \{\alpha \in \mathfrak{Fm}_s : \vdash_\vee \alpha\}$, where $|\delta|$ denotes the equivalence class of the formula δ.*

Proof. We only have to prove $\mathfrak{Fm}_s/\equiv_\vee$ is a join-semilattice and the axioms (a) and (b) from Definition 2.3 (2). So, the first part follows from (Ax4), (Ax5) and (Ax6), and the second one follows from axioms (Ax4) and $(R_\vee 3)$. □

Now, we will introduce some useful notions in order to prove a strong version of Completeness Theorem for $\mathcal{H}^3_{\vee,\triangle}$ w.r.t. the class of $H_3^{\vee,\triangle}$-algebras.

Recall that a logic defined over a signature $\mathcal{S}$ is a system $\mathcal{L} = \langle For, \vdash\rangle$ where For is the set of formulas over $\mathcal{S}$ and the relation $\vdash \subseteq \mathcal{P}(For) \times For$, $\mathcal{P}(A)$ is the set of all subsets of A. The logic $\mathcal{L}$ is said to be a Tarskian if it satisfies the following properties, for every set $\Gamma \cup \Omega \cup \{\varphi, \beta\}$ of formulas:

(1) if $\alpha \in \Gamma$, then $\Gamma \vdash \alpha$,

(2) if $\Gamma \vdash \alpha$ and $\Gamma \subseteq \Omega$, then $\Omega \vdash \alpha$,

(3) if $\Omega \vdash \alpha$ and $\Gamma \vdash \beta$ for every $\beta \in \Omega$, then $\Gamma \vdash \alpha$.

A logic $\mathcal{L}$ is said to be finitary if it satisfies the following:

(4) if $\Gamma \vdash \alpha$, then there exists a finite subset Γ_0 of Γ such that $\Gamma_0 \vdash \alpha$.

Definition 2.18 *Let $\mathcal{L}$ be a Tarskian logic and let $\Gamma \cup \{\varphi\}$ be a set of formulas, we say that Γ is a theory. In addition, Γ is said to be a consistent theory if there is φ such that $\Gamma \nvdash_{\mathcal{L}} \varphi$. Furthermore, we say that Γ is a maximal consistent theory if $\Gamma, \psi \vdash_{\mathcal{L}} \varphi$ for any $\psi \notin \Gamma$; and, in this case, we also say Γ non-trivial maximal respect to φ.*

A set of formulas Γ is closed in $\mathcal{L}$ if the following property holds for every formula φ: $\Gamma \vdash_{\mathcal{L}} \varphi$ if and only if $\varphi \in \Gamma$. It is easy to see that any maximal consistent theory is a closed one.

Lemma 2.19 (Lindenbaum-Łos) *Let $\mathcal{L}$ be a Tarskian and finitary logic. Let $\Gamma \cup \{\varphi\}$ be a set of formulas such that $\Gamma \nvdash \varphi$. Then, there exists a set of formulas Ω such that $\Gamma \subseteq \Omega$ with Ω maximal non-trivial with respect to φ in $\mathcal{L}$.*

Proof. It can be found [27, Theorem 2.22]. □

It is worth mentioning that, by the very definitions, $\mathcal{H}^3_{\vee,\triangle}$ is a Tarskian and finitary logic and then, we have the following:

Theorem 2.20 *Let $\Gamma \cup \{\varphi\} \subseteq \mathfrak{Fm}_s$, with Γ non-trivial maximal respect to φ in $\mathcal{H}^3_{\vee,\triangle}$. Let $\Gamma/\equiv_\vee = \{\overline{\alpha} : \alpha \in \Gamma\}$ be a subset of the trivalent modal Hilbert algebra with supremum $\mathfrak{Fm}_s/\equiv_\vee$, then:*

1. *If $\alpha \in \Gamma$ and $\overline{\alpha} = \overline{\beta}$ then $\beta \in \Gamma$,*
2. *$\Gamma/\equiv_\vee$ is a modal deductive system of $\mathfrak{Fm}/\equiv_\vee$. Also, if $\overline{\varphi} \notin \Gamma/\equiv_\vee$ and for any modal deductive system $\overline{D}$ which contains properly to $\Gamma/\equiv_\vee$, then $\overline{\varphi} \in \overline{D}$.*

Proof. Taking into account $\alpha \in \Gamma$ and $\alpha \equiv_\vee \beta$, we have that $\vdash \alpha \to \beta$ and $\vdash \beta \to \alpha$. Therefore, $\beta \in \Gamma$. Besides, it is not hard to see that D_1), D_2) and D_3) are valid, see Definition 2.6.

On the other hand, let $\overline{D}$ be mds that contains $\Gamma/\equiv_\vee$ and so, there is $\overline{\gamma} \in \overline{D}$ such that $\overline{\gamma} \notin \Gamma/\equiv_\vee$. Now, we have that $\gamma \notin \Gamma$ and therefore, $\Gamma \cup \{\gamma\} \vdash \varphi$. From the latter and taking into account $D = \{\alpha : \overline{\alpha} \in \overline{D}\}$, we can infer that $D \vdash \varphi$. Now, let us suppose that $\alpha_1, ..., \alpha_n$ is a derivation from D. We shall prove by induction over the length of the derivation that $\overline{\alpha_n} \in \overline{D}$. Indeed:

If $n = 1$, then α_1 is an instance of an axiom or otherwise $\alpha_1 \in D$. From the first case, we have $\vdash \alpha_1$ and then $\Gamma \vdash \alpha_1$ which is a contradiction. Then, it only can occur that $\alpha_1 \in D$ which implies $\overline{\varphi} \in \overline{D}$.

Suppose that $\overline{\alpha_k} \in \overline{D}$ if k is less than n. Then, we have the following cases:

1. If φ be the instance of an axiom, then $\Gamma \vdash \varphi$ which is a contradiction. This case can not occur.
2. If $\varphi \in D$, then $\overline{\varphi} \in \overline{D}$.
3. If there exists $\{j, t_1, ..., t_m\} \subseteq \{1, ..., k-1\}$ such that $\alpha_{t_1}, ..., \alpha_{t_m}$ is a derivation of $\alpha_j \to \varphi$, then we have $\overline{\alpha_j \to \varphi} \in \overline{D}$ by induction hypothesis. So, $\overline{\alpha_j} \to \overline{\varphi} \in \overline{D}$. From the latter and since $j < k$, we have $\overline{\alpha_j} \in \overline{D}$ and therefore, $\overline{\varphi} \in \overline{D}$.
4. If there exists $\{j, t_1, ..., t_m\} \subseteq \{1, ..., k-1\}$ such that $\alpha_{t_1}, ..., \alpha_{t_m}$ is a derivation of α_j and suppose that α_n is $\triangle\alpha_j$, then $\overline{\alpha_j} \in \overline{D}$. Now, since $\overline{D}$ is a mds, we have that $\triangle\overline{\alpha_j} \in \overline{D}$. Thus, $\overline{\varphi} \in \overline{D}$, which completes the proof. □

The notion of deductive systems considered in the last Theorem, part 2, was named *Systèmes deductifs liés à "a"* by A. Monteiro, where a is an element of some given algebra such that the congruences are determined by deductive systems [13, pag. 19], see also Definition 2.10.

Recall that for a given $H_3^{\vee,\triangle}$-algebra $\mathbf{A}$, a *logical matrix* for $\mathcal{H}^3_{\vee,\triangle}$ is a pair $\langle A, \{1\}\rangle$ where $\{1\}$ is the set of designated elements. In addition, we say that a homomorphism $v : \mathfrak{Fm}_s \to A$ is a valuation. Then, we say that φ is a *semantical consequence* of Γ and we denote by $\Gamma \vDash_{\mathcal{H}^3_{\vee,\triangle}} \varphi$, if for every $H_3^{\vee,\triangle}$-algebra $\mathbf{A}$ and every valuation v, if $v(\Gamma) = \{1\}$ then $v(\varphi) = 1$.

Corollary 2.21 *Let $\Gamma \cup \{\varphi\}$ be a set of formulas such that Γ non-trivial maximal respect to φ in $\mathcal{H}^3_{\vee,\triangle}$. Then, there exists a valuation $v : \mathfrak{Fm}_s \to \mathbb{C}_3^{\to,\vee}$ such that $v(\varphi) = 1$ iff $\varphi \in \Gamma$.*

Proof. Taking into account Theorem 2.20, we known that $\Gamma/\equiv_\vee$ is a maximal modal deductive system of $\mathfrak{Fm}_s/\equiv_\vee$. Then, by Theorem 2.12, there is a homomorphism $h : \mathfrak{Fm}_s/\equiv_\vee \to \mathbb{C}_3^{\to,\vee}$ (see Corollary 2.13) such that $h^{-1}(\{1\}) = \Gamma/\equiv_\vee$. Now, consider the canonical projection $\pi : \mathfrak{Fm}_s \to \mathfrak{Fm}_s/\equiv$ defined by $\pi(\alpha) = |\alpha|$, see Theorem 2.17. Now, it is enough to take $v = h \circ \pi$ to end the proof. □

Theorem 2.22 *(Soundness and Completeness of $\mathcal{H}^3_{\vee,\triangle}$ w.r.t. the class of $H_3^{\vee,\triangle}$-algebras) Let $\Gamma \cup \{\varphi\} \subseteq \mathfrak{Fm}_s$, $\Gamma \vdash_\vee \varphi$ if and only if $\Gamma \vDash_{\mathcal{H}^3_{\vee,\triangle}} \varphi$.*

Proof. Soundness: It is not hard to see that every axiom is valid for every $H_3^{\vee,\triangle}$-algebra A. In addition, satisfaction is preserved by the inference rules.

Completeness: Suppose $\Gamma \vDash_{\mathcal{H}^3_{\vee,\triangle}} \varphi$ and $\Gamma \nvdash_\vee \varphi$. Then, according to Lemma 2.19, there is maximal consistent theory M such that $\Gamma \subseteq M$ and $M \nvdash_\vee \varphi$. From the latter and Corollary 2.21, there is a valuation $\mu : \mathfrak{Fm}_s \to \mathbb{C}_3^{\to,\vee}$ such that $\mu(\Delta) = \{1\}$ but $\mu(\varphi) \neq 1$. □

3 Model Theory and First Order version of the logic of $\mathcal{H}_3^{\vee,\triangle}$ Without Identities

In this section, we will define the first order logic of $\mathcal{H}_3^{\vee,\triangle}$. First, let $\Sigma = \{\to, \vee, \triangle\}$ be the propositional signature of $\mathcal{H}_3^{\vee,\triangle}$, the symbols $\forall$ (universal quantifier) and $\exists$ (existential quantifier), with the punctuation marks (commas and parentheses). Let $Var = \{v_1, v_2, ...\}$ be a numerable set of individual variables. A first order signature Θ is composed of the following elements:

— a set $\mathcal{C}$ of individual constants,

— for each $n \geq 1$, $\mathcal{F}$ a set of functions of arity n,

— for each $n \geq 1$, $\mathcal{P}$ a set of predicates of arity n.

The notions of bound and free variables inside a formula, closed terms, closed formulas (or sentences), and of the term free for a variable in a formula are defined as usual, see [23]. We will denote by T_Θ and $\mathfrak{Fm}_\Theta$ the sets of all terms and formulas, respectively. Given a formula φ, the formula obtained from φ by substituting every free occurrence of a variable x by a term t will be denoted by $\varphi(x/t)$.

Definition 3.1 *Let Θ be a first order signature. The logic $\mathcal{QH}_3^{\vee,\triangle}$ over Θ is defined by Hilbert calculus obtained by extending $\mathcal{H}_3^{\vee,\triangle}$ expressed in the language $\mathfrak{Fm}_\Theta$ by adding the following:*

Axioms Schemas

(Ax11) $\varphi(x/t) \to \exists x\varphi$, *if t is a term free for x in φ,*

(Ax12) $\forall x\varphi \to \varphi(x/t)$, *if t is a term free for x in φ,*

(Ax13) $\triangle\exists x\varphi \leftrightarrow \exists x\triangle\varphi$,

(Ax14) $\triangle\forall x\varphi \leftrightarrow \forall x\triangle\varphi$,

Inferences Rules

(R3) $\dfrac{\varphi \to \psi}{\exists x\varphi \to \psi}$ *where x does not occur free in ψ,*

(R4) $\dfrac{\varphi \to \psi}{\varphi \to \forall x\psi}$ *where x does not occur free in φ.*

We denote by $\vdash \alpha$ the derivation of a formula α in $\mathcal{QH}_3^{\vee,\triangle}$ and with $\Gamma \vdash \alpha$ the derivation of α from a set of premises Γ. These notions are defined as the usual way. Furthermore, we denote $\vdash \varphi \leftrightarrow \psi$ as an abbreviation of $\vdash \varphi \to \psi$ and $\vdash \varphi \to \psi$.

Definition 3.2 *Let Θ be a first-order signature. A Θ-structure is a triple $\mathfrak{S} = \langle \mathbf{A}, S, \cdot^{\mathfrak{S}} \rangle$ such that $\mathbf{A}$ is a complete $H_3^{\vee,\triangle}$-algebra, and S is a non-empty set and $\cdot^{\mathfrak{S}}$ is an interpretation mapping defined on Θ as follows:*

1. *for each individual constant symbol c of Θ, $c^{\mathfrak{S}}$ of S,*
2. *for each function symbol f n-ary of Θ, $f^{\mathfrak{S}} : S^n \to S$,*
3. *for each predicate symbol P n-ary of Θ, $P^{\mathfrak{S}} : S^n \to A$.*

Given a Θ-structure $\mathfrak{S} = \langle \mathbf{A}, S, \cdot^{\mathfrak{S}} \rangle$, an $\mathfrak{S}$-valuation (or simply valuation) is a function $v : Var \to S$. Given $a \in S$ and $\mathfrak{S}$-valuation v, by $v[x \to a]$ we denote the following $\mathfrak{S}$-valuation, $v[x \to a](x) = a$ and $v[x \to a](y) = v(y)$ for any $y \in V$ such that $y \neq x$.

Let $\mathfrak{S} = \langle \mathbf{A}, S, \cdot^{\mathfrak{S}} \rangle$ be a Θ-structure and v an $\mathfrak{S}$-valuation. A Θ-structure $\mathfrak{S} = \langle \mathbf{A}, S, \cdot^{\mathfrak{S}} \rangle$ and an $\mathfrak{S}$-valuation v induce an interpretation map $||\cdot||_v^{\mathfrak{S}}$ for terms and formulas that can be defined as follows:

$$||c||_v^{\mathfrak{S}} = c^{\mathfrak{S}}, \text{ if } c \in \mathcal{C}$$

$$||x||_v^{\mathfrak{S}} = v(x), \text{ if } x \in Var$$

$$||f(t_1, \cdots, t_n)||_v^{\mathfrak{S}} = f^{\mathfrak{S}}(||t_1||_v^{\mathfrak{S}}, \cdots, ||t_n||_v^{\mathfrak{S}}), \text{ for any } f \in \mathcal{F},$$

$$||P(t_1, \cdots, t_n)||_v^{\mathfrak{S}} = P^{\mathfrak{S}}(||t_1||_v^{\mathfrak{S}}, \cdots, ||t_n||_v^{\mathfrak{S}}), \text{ for any } P \in \mathcal{P},$$

$$||\alpha \to \beta||_v^{\mathfrak{S}} = ||\alpha||_v^{\mathfrak{S}} \to ||\beta||_v^{\mathfrak{S}},$$

$$||\alpha \vee \beta||_v^{\mathfrak{S}} = ||\alpha||_v^{\mathfrak{S}} \vee ||\beta||_v^{\mathfrak{S}},$$

$$||\triangle\alpha||_v^{\mathfrak{S}} = \triangle||\alpha||_v^{\mathfrak{S}},$$

$$||\forall x\alpha||_v^{\mathfrak{S}} = \bigwedge_{a\in S} ||\alpha||_{v[x\to a]}^{\mathfrak{S}},$$

$$||\exists x\alpha||_v^{\mathfrak{S}} = \bigvee_{a\in S} ||\alpha||_{v[x\to a]}^{\mathfrak{S}}.$$

We say that $\mathfrak{S}$ and v *satisfy* a formula φ, denoted by $\mathfrak{S} \vDash \varphi[v]$, if $||\varphi||_v^{\mathfrak{S}} = 1$. Besides, we say that φ is *true* in $\mathfrak{S}$ if $||\varphi||_v^{\mathfrak{S}} = 1$ for each $\mathfrak{S}$-valuation v and denoted by $\mathfrak{S} \vDash \varphi$. We say that φ is a *semantical consequence* of Γ in $\mathcal{QH}_3^{\vee,\triangle}$, if, for any structure $\mathfrak{S}$: if $\mathfrak{S} \vDash \gamma$ for each $\gamma \in \Gamma$, then $\mathfrak{S} \vDash \varphi$. For a given set of formulas Γ, we say that the structure $\mathfrak{S}$ is a *model* of Γ iff $\mathfrak{S} \vDash \gamma$ for each $\gamma \in \Gamma$.

Now, it is worth mentioning that the following property $||\varphi(x/t)||_v^{\mathfrak{S}} = ||\varphi||_{v[x\to||t||_v^{\mathfrak{S}}]}^{\mathfrak{S}}$ holds. Another important aspect of the definition of *semantical consequence* is that it is different to the propositional case because if we use the definition of valuation for this case, we are unable to prove an important rule as $\alpha(x) \vDash \forall x\alpha(x)$.

In addition, we need to recall an important property of complete $H_3^{\vee,\triangle}$-algebra.

Lemma 3.3 [16, Lemma 0.1.21] *Let A be a complete $H_3^{\vee,\triangle}$-algebra and the set $\{a_i\}_{i\in I}$ of element of A for any non-empty set I. Then if there exists $\bigvee_{i\in I} a_i$ ($\bigwedge_{i\in I} a_i$), then there exists $\bigvee_{i\in I} \triangle a_i$ ($\bigwedge_{i\in I} \triangle a_i$) and also, $\bigvee_{i\in I} \triangle a_i = \triangle \bigvee_{i\in I} a_i$ and $\bigwedge_{i\in I} \triangle a_i = \triangle \bigwedge_{i\in I} a_i$.*

This property is useful to prove the following theorem:

Theorem 3.4 *(Soundness Theorem) Let $\Gamma \cup \{\varphi\} \subseteq \mathfrak{Fm}_\Theta$, if $\Gamma \vdash_\vee \varphi$ then $\Gamma \vDash \varphi$.*

Proof. In what follows we will consider an arbitrary but fixed structure $\mathfrak{S} = \langle A, S, \cdot^{\mathfrak{S}}\rangle$. It is clear that the propositional axioms are true in $\mathfrak{S}$. Now, we have to prove that the new axioms (Ax11) and (Ax12) are true in $\mathfrak{S}$, and the new inference rules (R3) and (R4) preserve trueness in $\mathfrak{S}$.

(Ax11) Suppose that φ is $\alpha(x/t) \to \exists x\alpha$. Then, $||\varphi||_v^{\mathfrak{S}} = ||\alpha||_{v[x\to||t||_v^{\mathfrak{S}}]}^{\mathfrak{S}} \to ||\exists x\alpha||_v^{\mathfrak{S}}$. It is clear that $||\alpha||_{v[x\to||t||_v^{\mathfrak{m}}]}^{\mathfrak{S}} \leq \bigvee_{a\in S} ||\alpha||_{v[x\to a]}^{\mathfrak{S}}$ and then, $||\alpha||_{v[x\to||t||_v^{\mathfrak{S}}]}^{\mathfrak{S}} \leq ||\exists x\alpha||_v^{\mathfrak{S}}$. Therefore $||\alpha(x/t) \to \exists x\alpha||_v^{\mathfrak{S}} = 1$ for every $\mathfrak{S}$-valuation v. (Ax12) is analogous to (Ax11). Now, according to Lemma 3.3, the axioms (Ax13) (Ax14) are true in $\mathfrak{S}$.

(R4) Let $\alpha \to \beta$ such that x is not free in α, and let $\alpha \to \forall x\beta$. Let us suppose that $||\alpha \to \beta||_v^{\mathfrak{S}} = 1$ for every $\mathfrak{S}$-valuation v. Now, consider a fix valuation v, then $||\alpha \to \forall x\beta||_v^{\mathfrak{S}} = ||\alpha||_v^{\mathfrak{S}} \to ||\forall x\beta||_v^{\mathfrak{S}} = ||\alpha||_v^{\mathfrak{S}} \to \bigwedge_{a\in S} ||\beta||_{v[x\to a]}^{\mathfrak{S}}$. On the other hand, by hypothesis, we know that $||\alpha||_u^{\mathfrak{S}} \leq ||\beta||_u^{\mathfrak{S}}$ for every $\mathfrak{S}$-valuation u. In particular, $||\alpha||_v^{\mathfrak{S}} = ||\alpha||_{v[x\to a]}^{\mathfrak{S}} \leq ||\beta||_{v[x\to a]}^{\mathfrak{S}}$ for every $\mathfrak{S}$-valuation v. Then, $||\alpha||_v^{\mathfrak{S}} \leq \bigwedge_{a\in S} ||\beta||_{v[x\to a]}^{\mathfrak{S}}$ and so, $||\alpha||_v^{\mathfrak{S}} \to \bigwedge_{a\in S} ||\beta||_{v[x\to a]}^{\mathfrak{S}} = 1$ for every $\mathfrak{S}$-valuation v. The proof of preservation of trueness for (R3) is analogous to (R4). □

In what follows, we will prove a strong version of Completeness Theorem for $\mathcal{QH}_{\vee,\triangle}^3$ using the Lindenbaum-Tarski algebra in a similar way to the propositional case. Let us observe that the algebra of formulas is an absolutely free algebra generated by the atomic formulas and its quantified formulas.

Now, let us consider the relation $\equiv$ defined by $\alpha \equiv \beta$ iff $\vdash \alpha \to \beta$ and $\vdash \alpha \to \beta$, then we have the algebra $\mathfrak{Fm}_\Theta/\equiv$ is a $H_3^{\vee,\triangle}$-algebra and the proof is exactly the same as in the propositional case (see, for instance, [1]). On the other hand, it is clear that $\mathcal{QH}_3^{\vee,\triangle}$ is a Tarskian and finitary logic. So, we can consider the notion of (maximal) consistent and closed theories with respect to some formula in the same way as the propositional case. Therefore, we have that Lindenbaum- Łos' Theorem holds for $\mathcal{QH}_3^{\vee,\triangle}$. Then, we have the following:

ISSN 2007-9737

Theorem 3.5 *Let $\Gamma \cup \{\varphi\} \subseteq \mathfrak{Fm}_\Theta$, with Γ non-trivial maximal respect to φ in $\mathcal{QH}_3^{\vee,\triangle}$. Let $\Gamma/\equiv = \{\overline{\alpha} : \alpha \in \Gamma\}$ be a subset of $\mathfrak{Fm}_\Theta/\equiv$, then:*

1. *If $\alpha \in \Gamma$ and $\overline{\alpha} = \overline{\beta}$, then $\beta \in \Gamma$. Besides, it is verified that $\Gamma/\equiv = \{\overline{\alpha} : \Gamma \vdash \alpha\}$, which, in this case, we say it is closed.*
2. *$\Gamma/\equiv$ is a modal deductive system of $\mathfrak{Fm}_\Theta/\equiv$. Also, if $\overline{\varphi} \notin \Gamma/\equiv$ and for any modal deductive system $\overline{D}$ which is closed in the sense of 1 and properly contains to $\Gamma/\equiv$, then $\overline{\varphi} \in \overline{D}$.*

Proof. According to the proof of Theorem 2.20, we only have to consider the rules $(R3)$ and $(R4)$. The fact that $\Gamma/\equiv$ is closed follows immediately.

In order to complete the proof, we have to consider two new cases 5 and 6. It is clear that $\Gamma/\equiv$ is a subset of $\overline{D}$. Now, let us consider $\overline{\phi} \in \overline{D}$ then $\overline{\phi} \notin \Gamma/\equiv$ and remember $D = \{\alpha : \overline{\alpha} \in \overline{D}\}$.

Case 5: There exists $\{j, t_1, ..., t_m\} \subseteq \{1, ..., k-1\}$ such that $\alpha_{t_1}, ..., \alpha_{t_m}$ is a derivation of $\alpha_j = \theta \to \beta$. Let us suppose that $\alpha_n = \exists x\theta \to \beta$ is obtained by α_j applying $(R3)$. From induction hypothesis, we have that $\overline{\theta \to \beta} \in \overline{D}$. From the latter, we obtain $\overline{\exists x\theta \to \beta} \in \overline{D}$.

Case 6: There exists $\{j, t_1, ..., t_m\} \subseteq \{1, ..., k-1\}$ such that $\alpha_{t_1}, ..., \alpha_{t_m}$ is a derivation of $\alpha_j = \theta \to \beta$. Let us suppose that $\alpha_n = \theta \to \forall x\beta$ is obtained by α_j applying $(R4)$. From induction hypothesis, we have $\overline{\theta \to \beta} \in \overline{D}$ and then, $\overline{\theta \to \forall x\beta} \in \overline{D}$. □

We note that for a given maximal consistent theory Γ of $\mathfrak{Fm}_\Theta$ we have $\Gamma/\equiv$ is a maximal modal deductive system of $\mathfrak{Fm}_\Theta/\equiv$. By well-known results of Universal Algebras, if we denote $A := \mathfrak{Fm}_\Theta/\equiv$ and $\theta := \Gamma/\equiv$, we have the quotient algebra A/θ is a simple algebra, see Corollary 2.13. From the latter and by adapting the first isomorphism theorem for Universal Algebras, we have that A/θ is isomorphic to $\mathfrak{Fm}_\Theta/\Gamma$ which is defined by the congruence $\alpha \equiv_\Gamma \beta$ iff $\alpha \to \beta, \beta \to \alpha \in \Gamma$.

Theorem 3.6 *(Completeness Theorem) Let $\Gamma \cup \{\varphi\}$ be a set of sentences, then $\Gamma \vDash \varphi$ then $\Gamma \vdash \varphi$.*

Proof. Let us suppose $\Gamma \vDash \varphi$ and $\Gamma \nvdash \gamma$. Then, by Lindenbaum- Łos' Lemma, there exists Δ maximal consistent theory such that $\Gamma \subseteq \Delta$. Now, consider the algebra $\mathfrak{Fm}_\Theta/\Delta$ defined by the congruence $\alpha \equiv_\Delta \beta$ iff $\alpha \to \beta, \beta \to \alpha \in \Delta$. In view of the above observations, we know that $\mathfrak{Fm}_\Theta/\Delta$ is isomorphic to a subalgebra of $\mathbb{C}_3^{\to,\vee}$ and so, complete as lattice.

Now, let us take the canonical projection $\pi_\Delta : \mathfrak{Fm} \to \mathfrak{Fm}_\Theta/\Delta$ defined by $\pi_\Delta(\alpha) = |\alpha|$ where $|\alpha|$ denotes the equivalence class of $\alpha \in \mathfrak{Fm}$. In this sense, consider the structure $\mathfrak{M} = \langle \mathfrak{Fm}_\Theta/\Delta, T_\Theta, \cdot^{T_\Theta} \rangle$ where T_Θ is a set of terms. It is clear that for every $t \in T_\Theta$ we have an associated constant $\hat{t}$ of Θ. Now, let us take a function $\mu : Var \to T_\Theta$ defined by $\mu(x) = x$. Then, we have the interpretation $||\cdot||_\mu^{\mathfrak{M}} : \mathfrak{Fm} \to \mathfrak{Fm}_\Theta/\Delta$ defined by if $\hat{t}$ is a constant, then $||\hat{t}||_\mu^{\mathfrak{M}} := t$; if $f \in \mathcal{F}$, then $||f(t_1, \cdots, t_n)||_\mu^{\mathfrak{M}} = f(t_1, \cdots, t_n)$; if $P \in \mathcal{P}$, then $||P(t_1, \cdots, t_n)||_\mu^{\mathfrak{M}} = \pi_\Delta(P(t_1, \cdots, t_n))$. Our interpretation is defined for atomic formulas but it is easy to see that $||\alpha||_\mu^{\mathfrak{M}} = \pi_\Delta(\alpha)$ for every quantifier-free formula α. Moreover, it is easy to see that for every formula $\phi(x)$ and every term t, we have $||\phi(x/\hat{t})||_\mu^{\mathfrak{M}} = ||\phi(x/t)||_\mu^{\mathfrak{M}}$. Therefore, from the latter property and by (Ax12) and (R4), we have $||\forall x\alpha||_\mu^{\mathfrak{M}} = \bigwedge_{a \in T_\Theta} ||\alpha||_{\mu[x \to a]}^{\mathfrak{M}}$ and now using (Ax11) and (R3), we obtain $||\exists x\alpha||_\mu^{\mathfrak{M}} = \bigvee_{a \in T_\Theta} ||\alpha||_{\mu[x \to a]}^{\mathfrak{M}}$. So, $||\cdot||_\mu^{\mathfrak{M}}$ is an interpretation map such that $||\alpha||_\mu^{\mathfrak{M}} = 1$ iff $\alpha \in \Delta$. On the other hand, it is not hard to see for every closed formula (sentence) β, we have $||\beta||_\mu^{\mathfrak{M}} = ||\beta||_v^{\mathfrak{M}}$ for every $\mathfrak{M}$-valuation v. Therefore, $\mathfrak{M} \vDash \gamma$ for every $\gamma \in \Gamma$ but $\mathfrak{M} \nvDash \varphi$ which is a contradiction. □

Given a formula φ and suppose $\{x_1, \cdots, x_n\}$ is the set of variables of φ, the *universal closure* of φ is defined by $\forall x_1 \cdots \forall x_n \varphi$. Thus, it is clear that if φ is a sentence, then the universal closure of φ is itself. Now, we are in condition to prove the following Completeness Theorem for formulas:

Theorem 3.7 *Let $\Gamma \cup \{\varphi\}$ be a set of formulas, then $\Gamma \vDash \varphi$ then $\Gamma \vdash \varphi$.*

Proof. Let us suppose $\Gamma \vDash \varphi$ and consider the set $\forall\Gamma$ all universal closure of Γ. From the latter and definition of $\vDash$, we have $\forall\Gamma \vDash \forall x_1 \cdots \forall x_n\varphi$. Then, according to Theorem 3.6, $\forall\Gamma \vdash \forall x_1 \cdots \forall x_n\varphi$. Now, from the latter, (Ax12) and (R4), we have $\Gamma \vdash \varphi$ as desired. □

4 Final Comments and Future Work

As final comments, we can say that our proof of the Completeness Theorem is different from the ones we can find in the literature (see for instance [1, 25]) because we use an algebraic technique developed by A. Monteiro, [13]. This technique can be used in the class studied in [6]. Indeed, consider the class of 3-valued modal Hilbert algebra ($\triangle H_3$-algebras) of Definition 2.2. From Lemma 2.11, it is possible to see that this class constitutes a semisimple variety. Now, let us consider the logic $\triangle\mathcal{H}_3$ over the signature $\{\rightarrow, \triangle\}$ defined by the axiom schemas (Ax1) to (Ax3) and (Ax7) to (Ax10), as well as the rules (MP) and (NEC). Taking in mind, the corresponding definitions of Section 2.1, it is possible to prove the following theorem:

Theorem 4.1 *(Soundness and Completeness of $\triangle\mathcal{H}_3$ w.r.t. the class of $\triangle H_3$-algebras) Let $\Gamma \cup \{\varphi\} \subseteq \mathfrak{Fm}_s$, $\Gamma \vdash_{\triangle\mathcal{H}_3} \varphi$ if and only if $\Gamma \vDash_{\triangle\mathcal{H}_3} \varphi$.*

Now, consider the first order version of $\triangle\mathcal{H}_3$ that we denote $\mathcal{Q}\triangle\mathcal{H}_3$. For $\mathcal{Q}\triangle\mathcal{H}_3$ we use the axioms (Ax11), (Ax12), rules of Definition 3.1 and notation of Section 3. Then, we have the following Theorem:

Theorem 4.2 *Let $\Gamma \cup \{\varphi\}$ be a set of formulas of $\mathcal{Q}\triangle\mathcal{H}_3$, then $\Gamma \vDash_{\mathcal{Q}\triangle\mathcal{H}_3} \varphi$ if only if $\Gamma \vdash_{\triangle H_3} \varphi$.*

The two last Theorems can be proved in the same way as the corresponding ones of the logic $\mathcal{H}^3_{\vee,\triangle}$ and $\mathcal{QH}_3^{\vee,\triangle}$. Yet this technique can not be applied to any logics from non-semisimple varieties, such as (n-valued) Heyting algebras, MV-algebras, Hilbert algebras, residuated lattices and so on.

As future work, we will present a study of logics from semisimple varieties of algebras studied in the Monteiro's school. All these systems will allow us to apply the technique presented in this paper.

Acknowledgments

The first author acknowledges the support of a grant 2016/21928-0 from São Paulo Research Foundation (FAPESP), Brazil. Also, Slagter was financially supported by a Ph.D. grant from CONICET, Argentina.

References

1. **Bell, J., Slomson, A. (1971).** Models and Ultraproducts: An Introduction. North Holland, Amsterdam.
2. **Boicescu, V., Filipoiu, A., Georgescu, G., Rudeanu, S. (1991).** Lukasiewicz - Moisil Algebras. Annals of Discrete Mathematics, Vol. 49, North - Holland.
3. **Cignoli, R. (1984).** An algebraic approach to elementary theories based on n-valued Łukasiewicz logics. Z. Math. Logik Grundlag. Math., Vol. 30 , No. 1, pp. 87–96.
4. **Cignoli, R. (1982).** Proper n-valued Łukasiewicz algebras as S-algebras of Łukasiewicz n-valued propositional calculi. Studia Logica, Vol. 41, No. 1, pp. 3–16.
5. **Ciucci, D., Dubois, D. (2012).** Three-valued logics for incomplete information and epistemic logic. European Workshop on Logics in Artificial Intelligence, Springer, pp. 147–159. DOI: 10.1007/978-3-642-33353-8_12.
6. **Canals Frau, M., Figallo, A.V., Saad, S. (1990)**. Modal three-valued Hilbert algebras. Preprints del Instituto de Ciencias Básicas, Universidad Nacional de San Juan, pp. 1–21.
7. **Canals Frau, M., Figallo, A.V., Saad, S. (1992).** Modal 3-valued implicative semilattices. Preprints del Instituto de Ciencias Básicas, Universidad Nacional de San Juan, pp. 1–24.
8. **Diego, A. (1966).** Sur les algèbres de Hilbert. Colléction de Logique Mathèmatique, ser. A, fasc. 21, Gouthier-Villars.

9. **Figallo, A.V., Ramón, G., Saad, S. (2003).** A note on the Hilbert algebras with infimum. Mat. Contemp., Vol. 24, pp. 23–37.

10. **Figallo, A.V., Ramón, G., Saad, S. (2006).**, iH-Propositional calculus. Bull. Sect. Logic Univ. Lódz, Vol. 35, No. 4, pp. 157–162.

11. **Figallo Jr., A., Ziliani, A. (2005).** Remarks on Hertz algebras and implicative semilattices. Bull. Sect. Logic Univ. Lódz, Vol. 34, No. 1, pp. 37–42.

12. **Hernández-Tello, A., Arrazola-Ramírez, J.R., Osorio-Galindo, M.J. (2017).** The pursuit of an implication for the logics L3A and L3B. Logica Universalis, Vol. 11, pp. 507–524.

13. **Monteiro, A. (1980).** Sur les algèbres de Heyting simetriques. Portugaliae Math., Vol. 39, No. 1-4 , pp. 1–237.

14. **Monteiro, A. (1996).** Les algèbras de Hilbert linaires, Unpublished papers I. Notas de Lógica Matemtica, Vol. 40, pp. 114–127.

15. **Monteiro, L. (1977).** Algèbres de Hilbert $n-$valentes. Portugaliae Math., Vol. 36, pp. 159–174.

16. **Monteiro, L. (1973).** Algebras de Łukasiewicz trivalentes monádicas. Ph. D. thesis, Universidad Nacional del Sur.

17. **Pérez-Gaspar, M., Hernández-Tello, A., Arrazola-Ramírez, J., Osorio-Galindo, M. (2020).** An axiomatic approach to CG'3 logic. Logic Journal of the IGPL, Vol. 28, No. 6, pp. 1218–1232.

18. **Osorio, M.,Carballido, J., Zepeda, C. (2018).** SP3B as an extension of C1. South American Journal of Logic, Vol. 4, No. 1, pp. 1–27.

19. **Osorio, M., Carballido, J.L. (2008).** Brief study of G'3 logic. Journal of Applied Non-Classical Logic, Vol. 18, No. 4, pp. 475–499.

20. **Osorio, M., Zepeda, C., Nieves, J.C., Carballido, J.L. (2009).** G'_3–stable semantics and inconsistency. Computación y Sistemas, Vol. 13, pp. 75–86.

21. **Osorio, M., Navarro, J., Arrazola, J., Borja, V. (2006).** Logics with common weak completions. Journal of Logic and Computation, Vol. 16, No. 6, pp. 867–890.

22. **Macías, V.B., Pérez-Gaspar, M. (2016).** Kripke-type semantics for CG'3. Electronic Notes in Theoretical Computer Science, Vol. 328, pp. 17–29.

23. **Mendelson, E. (2009).** Introduction to Mathematical Logic. CRC Press.

24. **Slagter, J.S. (2016).** Reductos hilbertianos de las álgebras de Łukasiewicz-Moisil de orden 3. Master's thesis, Universidad Nacional del Sur.

25. **Rasiowa, H. (1974).** An algebraic approach to non-clasical logics. Studies in logic and the foundations of mathematics, North-Holland Publishing Company, Amsterdam and London, and American Elsevier Publishing Company, Inc., New York, Vol. 78.

26. **Thomas, I. (1962).** Finite limitations on Dummett's LC. Notre Dame Journal of Formal Logic, Vol. 3, pp. 170–174.

27. **Wójcicki, R. (1984).** Lectures on propositional calculi. Ossolineum, Warsaw.

Article received on 11/10/2020; accepted on 20/02/2021.
Corresponding author is Aldo Figallo-Orellano.

ISSN 2007-9737

Computing the Clique-Width on Series-Parallel Graphs

Marco Antonio López-Medina, J. Leonardo González-Ruiz,
J.Raymundo Marcial-Romero, J. A. Hernández

Universidad Autónoma del Estado de México,
Facultad de Ingeniería,
Mexico

valgirmanda@gmail.com, {jlgonzalezru,jrmarcialr,xoseahernandez}@uaemex.mx

Abstract. The clique-width (cwd) is an invariant of graphs which, similar to other invariants like the tree-width (twd) establishes a parameter for the complexity of a problem. For example, several problems with bounded clique-width can be solved in polynomial time. There is a well known relation between tree-width and clique-width denoted as $cwd(G) \leq 3 \cdot 2^{twd(G)-1}$. Serial-parallel graphs have tree-width of at most 2, so its clique–width is at most 6 according to the previous relation. In this paper, we improve the bound for this particular case, showing that the clique-width of series-parallel graphs is smaller or equal to 5.

Keywords. Graph theory, clique-width, tree-width, complexity, series-parallel.

1 Introduction

The clique-width is an invariant which set up a parameter to measure the complexity of a problem. Computing the clique-width consists on finding an algebraic finite term which represents in a succinct way the graph, meaning that its operations establishes how to built the graph. Courcelle et al. [3] present a set of four operations to built the algebraic expression called a term: 1) label creations which represent a vertex, 2)disjoint unions among graphs, 3) edge creation and 4) vertex re-label. The number of labels used to built a finite term is commonly denoted by k. The minimum number k used to built the term, also called k-expression, defines the clique-width.

Finding the smallest k which minimize the k-expression is an NP-Complete problem [7].

It has been observed that if the clique-width increases for a certain class of graphs then the complexity of a given problem for such a class of graphs also increases since the difficulty to decompose the graph increases. In recent years, clique-width has been studied in different class of graphs showing the behaviour of this invariant under certain operations.

Recent research shows how to calculate the clique-width in special types of graphs, for example in [12] prove that $(4k_1, C_4, C_5, C_7)$-free graphs that are not chordal have unbounded clique-width. Also in [5] a complete classification of graphs H was obtained, they shown that for these graph classes, a well-quasi-orderability implies boundedness of clique-width.

In [10], it is shown that the clique-width of Cactus graphs is smaller or equal to 4 and is presented a polynomial time algorithm which computes exactly a 4-expression. Also in [9] it is shown how to compute the cwd of Polygonal Tree Graphs and is presented a polynomial time algorithm which computes the 5-expression.

In a similar way, another invariant of graphs is tree-width [8], however, cwd is more general than tree width in the sense that, graphs with small tree-width also have small cwd.

A special class of graphs are the so called series-parallel graphs which can be obtained by recursive applications of series and parallel connections [6, 11]. This kind of graphs are a subclass of what are called planar graphs.

In this paper we show how to built a series-parallel graph and later on the algebraic

5-expression which defines the cwd, so we show that the cwd of a series-parallel graph is 5 improving the best known bound known of 6 [2].

The structure of the paper is as follows: section 2 presents the preliminaries of the paper, in section 3 the main result is demonstrated, an algorithm to compute the clique-width is shown in section 4. Finally, the conclusions are established in section 5.

2 Preliminaries

2.1 Graph

A graph G is denoted by $G = (V(G), E(G))$, where $V(G)$ is the set of vertices in G and $E(G)$ the set of edges in G. A *path graph* is denoted as a set of connected vertices that have two end points and every inner vertex x_i have exactly two incident edges, $d(x_i) = 2$.

2.2 Series-Parallel Graph

A graph is series-parallel if it can be built from a single edge and the following two operations:

1. series construction: subdividing an edge in the graph.
2. parallel construction: duplicating an edge in the graph.

Another characterization of a series-parallel graph is that it do not contain a subdivision of k_4 (complete graph of 4 vertices).

As the first characterization of series-parallel graphs implies, a series-parallel graph always has a vertex of degree two, although series-parallel operations may construct multiple edges, in this paper we only work with simple graphs.

2.3 Clique-Width

We now introduce the notion of clique-width (*cwd*, for short). Let $\mathscr{C}$ be a countable set of labels. A *labeled* graph is a pair (G, γ) where γ maps each element of $V(G)$ into $\mathscr{C}$. A labeled graph can also be defined as a triple $G = (V(G), E(G), \gamma(G))$ and its labeling function is denoted by $\gamma(G)$. We say that G is C-labeled if C is finite and $\gamma(G)(V) \subseteq C$. We denote by $\mathscr{G}(C)$ the set of undirected C-labeled graphs.A vertex with label a will be called an a-port. We introduce the following symbols:

— a nullary symbol $a(v)$ for every $a \in \mathscr{C}$ and $v \in V$;

— a unary symbol $\rho_{a\to b}$ for all $a, b \in \mathscr{C}$, with $a \neq b$;

— a unary symbol $\eta_{a,b}$ for all $a, b \in \mathscr{C}$, with $a \neq b$;

— a binary symbol $\oplus$.

These symbols are used to denote operations on graphs as follows: $a(v)$ creates a vertex with label a corresponding to the vertex v, $\rho_{a\to b}$ renames the vertex a by b, $\eta_{a,b}$ creates an edge between a and b, and $\oplus$ is a disjoint union of graphs.

For $C \subseteq \mathscr{C}$ we denote by $T(C)$ the set of finite well-formed terms written with the symbols $\oplus, a, \rho_{a\to b}, \eta_{a,b}$ for all $a, b \in C$, where $a \neq b$. Each term in $T(C)$ denotes a set of labeled undirected graphs. Since any two graphs denoted by the same term t are isomorphic, one can also consider that t defines a unique abstract graph.

The following definitions are given by induction on the structure of t. We let $val(t)$ be the set of graphs denoted by t.

If $t \in T(C)$ we have the following cases:

1. $t = a \in C$: $val(t)$ is the set of graphs with a single vertex labeled by a;
2. $t = t_1 \oplus t_2$: $val(t)$ is the set of graphs $G = G_1 \cup G_2$ where G_1 and G_2 are disjoint and $G_1 \in val(t_1)$, $G_2 \in val(t_2)$;
3. $t = \rho_{a\to b}(t') : val(t) = \{\rho_{a\to b}(G) | G \in val(t')\}$ where for every graph G in $val(t')$, the graph $\rho_{a\to b}(G)$ is obtained by replacing in G every vertex label a by b;

4. $t = \eta_{a,b}(t') : val(t) = \{\eta_{a,b}(G) | G \in val(t')\}$ where for every undirected labeled graph $G = (V, E, \gamma)$ in $val(t')$, we let $\eta_{a,b}(G) = (V, E', \gamma)$ such that:
   $E' = E \cup \{\{x, y\} | x, y \in V, x \neq y, \gamma(x) = a, \gamma(y) = b\}$, e.g. $\eta_{a,b}(G)$ adds an edge between each pair of vertices a and b in G.

For every labeled graph G we let:

$$cwd(G) = min\{|C| | G \in val(t), t \in T(C)\}.$$

A term $t \in T(C)$ such that $|C| = cwd(G)$ and $G = val(t)$ is called optimal *expression of* G [4] and written as $|C|$-expression.

In other words, the clique-width of a graph G is the minimum number of different labels needed to construct a vertex-labeled graph isomorphic to G using the four mentioned operations [1].

3 Computing $cwd(G)$ when G is a Series-Parallel Graph

In this section we show the k-expression for series and parallel graphs independently and later on how to combine them in order to present the 5-expression for series-parallel graphs. We firstly begins with series graphs. Although the result for this kind of graphs is well-known, we need a special construction to combine them with parallel graphs.

Lemma 1 *If G is a series graphs (a path graph) then $cwd(G) \leq 4$.*

Proof 1 *Let G be a series graph, which is denoted as follows:*

1 —— 2 —— 3 —— 4 —— 5 ---- n

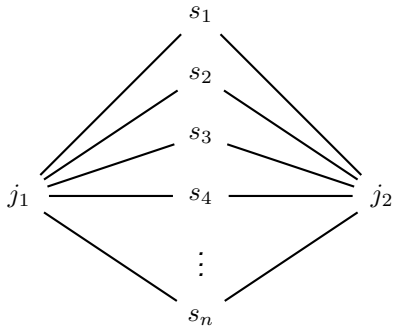
The k-expression is built as follows:

$k-expression$	*Graph G*	*Labels*
	$a(1)$ —— $b(2)$	
$k_G = \eta_{(a,b)}(a(1) \oplus b(2))$		*2*
	$a(1) \cdot b(2) \cdot c(3)$	
$k_G = \eta_{(b,c)}(k_G \oplus c(3))$		*3*
	$a(1) \cdot b(2) \cdot c(3) \cdot d(4)$	
$k_G = \eta_{(c,d)}(k_G \oplus d(4))$		*4*
	$a(1) \cdot b(2) \cdot b(3) \cdot d(4)$	
$k_G = \rho_{c \to b}(k_G)$		*3*
	$a(1) \cdot b(2) \cdot b(3) \cdot c(4)$	
$k_G = \rho_{d \to c}(k_G)$		*3*
	$a(1) \cdot b(2) \cdot b(3) \cdot c(4) \cdot d(5)$	
$k_G = \eta_{(c,d)}(k_G \oplus d(5))$		*4*
	$a(1) \cdot b(2) \cdot b(3) \cdot b(4) \cdot d(5)$	
$k_G = \rho_{c \to b}(k_G)$		*3*
	$a(1) \cdot b(2) \cdot b(3) \cdot b(4) \cdot c(5)$	
$k_G = \rho_{d \to c}(k_G)$		*3*
$\vdots$		
	$a(1) \cdot b(2) \cdot b(3) \cdot b(4) \cdot c(5) \cdot d(n)$	
$k_G = \eta_{(c,d)}(k_G \oplus d(n))$		*4*
	$a(1) \cdot b(2) \cdot b(3) \cdot b(4) \cdot b(5) \cdot d(n)$	
$k_G = \rho_{c \to b}(k_G)$		*3*
	$a(1) \cdot b(2) \cdot b(3) \cdot b(4) \cdot b(5) \cdot c(n)$	
$k_G = \rho_{d \to c}(k_G)$		*3*

4 labels are used to built a series graph. At the end of the process we relabel the end vertices as a and c respectively, while the rest of the vertices are assigned label b, this assignment will be used at the end of each proof in the rest of the paper.

Lemma 2 *If G is a parallel graph formed by series subgraphs then $cwd(G) \leq 5$.*

Proof 2 *Let n be the number of series subgraphs which forms the parallel graph:*



ISSN 2007-9737

By lemma 1, each k-expression of $s_1, s_2, s_3 \dots s_n$ requires 3 labels, let says a, b and c. Let a and c be the end vertices of each one. If j_1 and j_2 are the union vertices the final k-expression is given by:

$k_G = \eta_{(c,e)}(\eta_{(a,d)}(k_{s_1} \oplus k_{s_2} \oplus k_{s_3} \oplus k_{s_4} \oplus \cdots \oplus k_{s_n} \oplus d(j_1) \oplus e(j_2)))$

$k_G = \rho_{e\to c}((\rho_{c\to b}((\rho_{d\to a}((\rho_{a\to b}(k_G))))$

Although 5 labels are needed, in the last steps the joint vertices j_1 and j_2 are labeled with a and c respectively and the rest of the vertices are labeled with b.

A series-parallel graph can be composed by the following rules:

— A simple path is series-parallel (SP), Lemma 1.

— A parallel graph formed by series subgraphs is series parallel (SP). Lemma 2

— if SP_1 and SP_2 are series parallel graphs then:

  – The path graph formed by $SP_1, SP_2, ..., SP_n$ is series parallel (SP). Lemma 5.
  – The parallel graph formed by $SP_1, SP_2, ..., SP_n$ with union points j_1, j_2 is series parallel (SP). Lemma 3
  – The parallel graph formed by $SP_2, SP_3, ..., SP_n$ with union points SP_1, j_1 is series parallel (SP). Lemma 4

Lemma 3 *Let G a series-parallel graph which is connected to an other series-parallel graph, then the $cwd(G) \leq 5$.*

Proof 3 *Let G a parallel graph as follows:*

SP_1 ——————— SP_2

Where SP_1 and SP_2 are series-parallel graphs and j_1 is a joint vertex. By lemma 2 shows how to build the k-expression of SP_1 and SP_2 respectively.

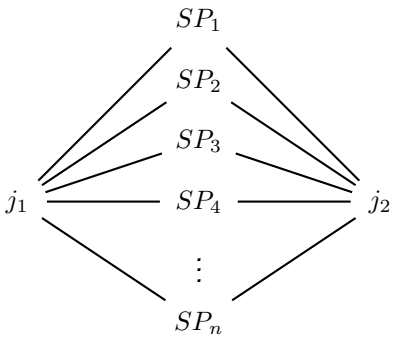
$k_G = \eta_{(d,e)}((\rho_{c\to d}(k_{SP_1})) \oplus (\rho_{a\to e}(k_{SP_2})))$

$k_G = \rho_{d\to b}(\rho_{e\to b}(k_G))$

The initial vertex of SP_1 and the final vertex of SP_2 are labelled by a and c respectively, while the rest of the vertices correspond to the label b.

Lemma 4 *If G is a graph which contains series-parallel subgraphs then $cwd(G) \leq 5$.*

Proof 4 *Let n be the number of series-parallel subgraphs which forms the parallel graph where $n \geq 0$:*



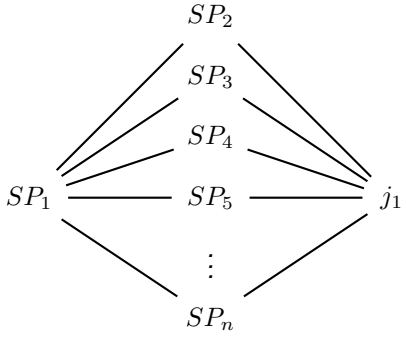
By lemmas 1, 2, 3, each k-expression of $SP_1, \dots, SP_n$ requires 3 labels, let says a, b and c. The end vertices of each one are a and c. If j_1 and j_2 are the union vertices the final k-expression is given by:

$k_G = \eta_{(c,e)}(\eta_{(a,d)}(k_{SP_1} \oplus \cdots \oplus k_{SP_n} \oplus d(j_1) \oplus e(j_2)))$

$k_G = \rho_{e\to c}((\rho_{c\to b}((\rho_{d\to a}((\rho_{a\to b}(k_G))))$

The end vertices j_1 and j_2 are labeled with a and c respectively and the rest of the vertices are labeled with b.

Lemma 5 *Let G be a parallel graph with end points SP_1 and j_1 and elements $SP_2, SP_3, ..., SP_n$.*



Proof 5

By lemmas 1, 2, 3 and 4, we know the k-expression of SP_1 and each k-expression of $SP_1, \ldots, SP_n$ requires 3 labels, let says a, b and c. The end vertices of each one are a and c:

$k_G = \eta_{(e,d)}(\rho_{a\to d}(k_{SP_2} \oplus \cdots \oplus k_{SP_n})) \oplus (\rho_{c\to e}(k_{SP_1}))$,
$k_G = \rho_{d\to c}(\rho_{c\to b}(\eta_{(c,d)}((\rho_{d\to b}(\rho_{e\to b}(k_G))) \oplus d(j_1))))$.

The initial vertex of SP and the joint vertex j_1 are labelled by a y c respectively, while the rest of the vertices correspond to the label b.

Lemma 5 can be applied transitively, e.g. j_1 to the left and SP_1 to the right.

Theorem 1 *Let G a series-parallel graph, the $cwd(G) \leq 5$.*

Proof 6 *By series-parallel definition lemmas 1, 2 , 3, 4 and 5 allow to built any series parallel graph so $cwd(G)$ is ≤ 5*

4 Algorithm to Compute cwd of Series-Parallel Graphs

The construction of the k-expression of a series-parallel graph is presented in Algorithm 1 and 2.

Algorithm 1 Construction of the k-expression of a series-parallel graph (Part1)

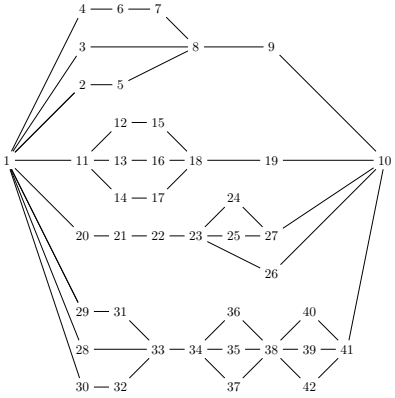
Require: A series-parallel graph G
Ensure: k-expression of a series-parallel graph
Construct the adjacency matrix A of G
Construct the incidence matrix I of G
An empty set SPs of tuples of the form (sp, k_{sp}), where sp is a subgraph of G and k_{sp} is the k-expression of sp
Find the series subgraphs $sp_i \in G$ (paths of vertices with degree two) and construct k_{sp_i} (lemma 1)
for each sp_i **do**
  Add the tuple (sp_i, k_{sp_i}) to SPs
  Remove from A all edges forming the sp_i subgraph
end for
Remove from I all vertices with degree two

Algorithm 2 Construction of the k-expression of a series-parallel graph (Part2)

while $A \neq \emptyset$ **do**
  Find the subgraphs sp_k in SPs connected to the same vertices $i, j \in I$ (to form a parallel subgraph sp_p)
  Construct the k-expressions of the parallel subgraphs formed by the sp_k subgraphs (lemma 2 and 5)
  for each sp_p **do**
    Add the tuple (sp_p, k_{sp_p}) to SPs
    Remove sp_k from SPs
    Remove the edges on sp_p from A
    Remove the vertices i, j from I
  end for
  Find the subgraphs sp_k in SPs connected to the vertex $j \in I$ and a vertex $i \in sp_u \in SPs$ (to form a parallel subgraph sp_p)
  if $|sp_k| - d(j) \leq 1$ and $|sp_k| - d(i) \leq 1$ **then**
    Construct the k-expression of the parallel subgraph formed by the sp_k subgraphs (lemma 4)
    for each sp_p **do**
      Add the tuple (sp_p, k_{sp_p}) to SPs
      Remove sp_k from SPs
      Remove the edges on sp_p from A
      Remove the vertex j from I
      Remove sp_u from SPs
    end for
  end if
  Find the subgraphs sp_i, sp_j connected with an edge $e \in A$ (to form a series subgraph sp_e)
  for each pair sp_i and sp_j **do**
    Construct the k-expresion of the subgraph formed by $sp_i \cup sp_j \cup e$ (lemma 5)
    Add the tuple (sp_e, k_{sp_e}) to SPs
    Remove the edge e from A
    Remove sp_i and sp_j from SPs
  end for
end while
return k-expression of the remaining element in the set SPs

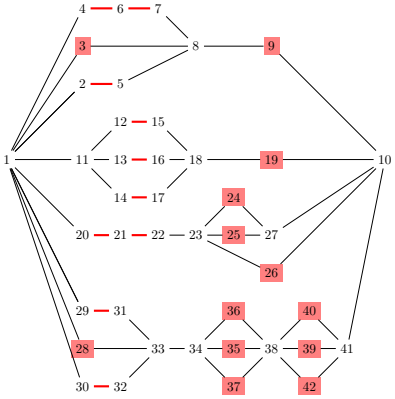
We explain the algorithm with the following example:

Given a series-parallel graph:

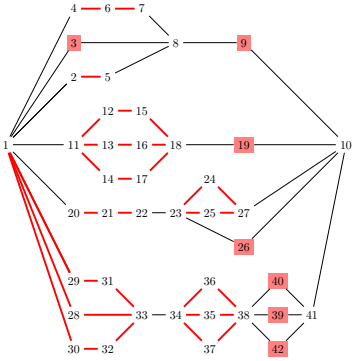


With the adjacency matrix A, the incidence matrix I and the set SPs.

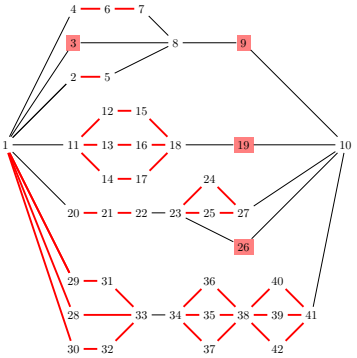
First lines from 3 to 9 allow to construct the sp_i subgraphs, formed by paths of vertices with degree two, using lemma 1.



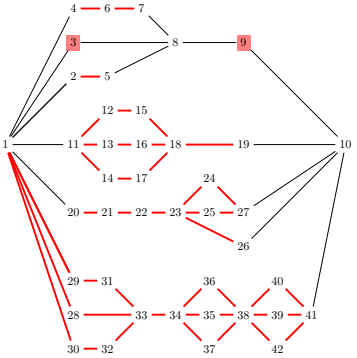
From line 11 to 18 we construct the parallel graphs with the joint vertices we have in I (lemma 2 and 5).



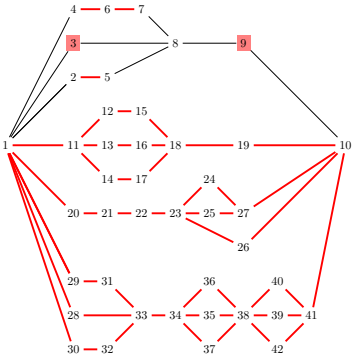
From lines 19 to 29 we can construct a parallel graph with joint vertex and a vertex on a sp_k subgraph (lemma 4). Notice that the end point 1 and 8 cannot be added at this time since the degree of 1 will not be 0 after joining it to the subgraphs.



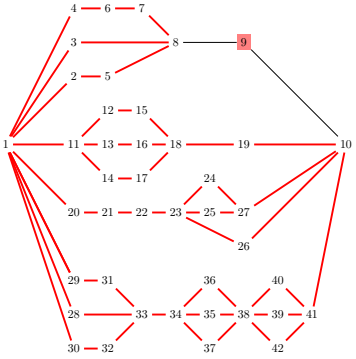
From lines 30 to 36 we can connect two sp_i and sp_k subgraphs by an edge in A (lemma 5).



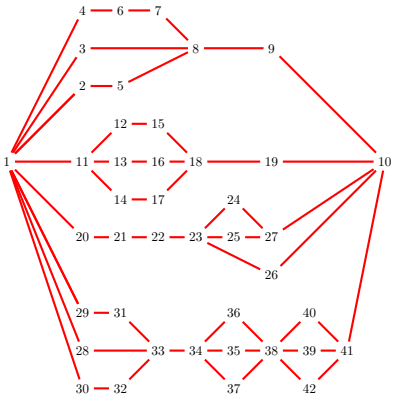
From lines 19 to 29 we can construct a parallel graph with joint vertex and a vertex on a sp_k subgraph (lemma 4).



Again, from lines 19 to 29 we can construct a parallel graph with joint vertex and a vertex on a sp_k subgraph (lemma 4).



Finally, from lines 30 to 36 we can connect two sp_i and sp_k subgraphs by an edge in A (lemma 5).



As a result of the algorithm we have a unique element $sp \in SPs$ with the k-expression that represents it.

5 Conclusions

In this paper we show that five labels are enough to compute the clique-width of series-parallel graphs instead of six labels as Courcelle et al. [2] shown. Our main proof is based on the series-parallel graph's definition which consists on building this kind of graph from series subgraphs joined by vertices which form parallel components. An algorithm was presented with time complexity $O(n^2)$.

References

1. **Bonomo, F., Grippo, L. N., Milanic, M., Safe, M. D. (2016).** Graph classes with and without powers of bounded clique-width. Discrete Applied Mathematics, Vol. 199, pp. 3–15. Sixth Workshop on Graph Classes, Optimization, and Width Parameters, Santorini, Greece, October 2013.
2. **Corneil, D. G., Rotics, U. (2001).** Graph-Theoretic Concepts in Computer Science: 27th InternationalWorkshop, WG 2001 Boltenhagen, Germany, June 14–16, 2001 Proceedings, chapter On the Relationship between Clique-Width and Treewidth. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 78–90.
3. **Courcelle, B., Engelfriet, J., Rozenberg, G. (1993).** Handle-rewriting hypergraph grammars. Journal of Computer and System Sciences, Vol. 46, No. 2, pp. 218–270.
4. **Courcelle, B., Olariu, S. (2000).** Upper bounds to the clique width of graphs. Discrete Applied Mathematics, Vol. 101, pp. 77–114.
5. **Dabrowski, K. K., Lozin, V. V., Paulusma, D. (2020).** Clique-width and well-quasi-ordering of triangle-free graph classes. Journal of Computer and System Sciences, Vol. 108, pp. 64–91.
6. **Dieter, J. (2013).** Graphs, Networks and Algorithms. Springer Publishing Company, Incorporated, 4th edition.
7. **Fellows, M. R., Rosamond, F. A., Rotics, U., Szeider, S. (2009).** Clique-width is np-complete. SIAM Journal on Discrete Mathematics, Vol. 23, No. 2, pp. 909–939.
8. **Fomin, F. V., Golovach, P. A., Lokshtanov, D., Saurabh, S. (2010).** Intractability of clique-width parameterizations. SIAM Journal on Computing, Vol. 39, No. 5, pp. 1941–1956.
9. **González-Ruiz, J. L., Marcial-Romero, J. R., Hernández, J. A., De Ita, G. (2017).** Computing the clique-width of polygonal tree graphs. **Pichardo-Lagunas, O., Miranda-Jiménez, S.**, editors, Advances in Soft Computing, Springer International Publishing, Cham, pp. 449–459.
10. **González-Ruiz, J. L., Marcial-Romero, J. R., Hernández-Servín, J. (2016).** Computing the clique-width of cactus graphs. Electronic Notes in Theoretical Computer Science, Vol. 328, pp. 47–57. Tenth Latin American Workshop on

Logic/Languages, Algorithms and New Methods of Reasoning (LANMR).

11. **Gross, J. L., Yellen, J., Zhang, P. (2013).** Handbook of Graph Theory, Second Edition. Chapman & Hall/CRC, 2nd edition.
12. **Penev, I. (2020).** On the clique-width of (4k1,c4,c5,c7)-free graphs. Discrete Applied Mathematics, Vol. 285, pp. 688–690.

Article received on 15/10/2020; accepted on 20/02/2021.
Corresponding author is Marco Antonio López-Medina.

ISSN 2007-9737

Intuitionistic Epistemic Logic with Distributed Knowledge

Ryo Murai[1], Katsuhiko Sano[2]

[1] Hokkaido University,
Japan

[2] Hokkaido University,
Faculty of Humanities and Human Sciences,
Japan

ryo.murai11@gmail.com, v-sano@let.hokudai.ac.jp

Abstract. We develop intuitionistic epistemic logics with distributed knowledge, which is more general than a logic proposed by (Jäger & Marti 2016) in that a distributed knowledge operator is parameterized by a group of agents. Specifically, we present Hilbert systems of intuitionistic K, KT, KD, K4, K4D, and S4 with distributed knowledge. The semantic completeness of the logics with regard to suitable Kripke frames is shown by modifying the standard argument of the semantic completeness of classical distributed knowledge logics via the concept of pseudo-model. We also present cut-free sequent calculi for the logics, based on which we establish Craig interpolation theorem and decidability.

Keywords. Intuitionistic logic, epistemic logic, distributed knowledge.

1 Introduction

'Distributed knowledge' is one of the notions of group knowledge studied in multi-agent epistemic logic [6, 18]. A typical example of distributed knowledge is the following: a group consisting of a and b has distributed knowledge of a fact q when a knows that $p \to q$ and b knows that p. According to [1, Section 1], "distributed knowledge is the knowledge of a third party, someone 'outside the system' who somehow has access to the epistemic states of all the group members". Fagin et al. [6, p. 3] stated as an intuitive description for distributed knowledge "a group has distributed knowledge of a fact φ if the knowledge of φ is distributed among its members, so that by pooling their knowledge together the members of the group can deduce φ". At first sight, the latter description seems clearer than the former. Ågotnes and Wáng [1] state, however, that the above intuitive description by Fagin et al. [6, p. 3] is inappropriate by an illustrative example given in [1, Section 1].

Formally, distributed knowledge is expressed as a modal operator D_G, parameterized by a finite group G of agents and the satisfaction of $D_G\varphi$ at a state w is defined as: φ holds at all states v such that v can be reached in a single step from w for all agents in G, i.e., wR_av for all agents $a \in G$, where R_a is a binary relation on the states. As for the model-theoretic study of distributed knowledge, we can cite [1, 25, 10, 28]. Proof-theoretic study is relatively less active, but there have been proposed several sequent calculi [12, 23, 11, 19]. However, those cited here are all on the basis of classical logic.

Not to mention distributed knowledge, epistemic logic as a whole has been studied mainly in the classical setting. However, several kinds of intuitionistic epistemic logics have been proposed from different perspectives. Several philosophical logicians have proposed intuitionistic epistemic logics [31, 24, 3] for the sake of analysis of Fitch's knowability paradox [7], from the verificationist point of view.

Another kind of intuitionistic epistemic logic [14] is proposed for the analysis of distributed computing in the sense of [13, 26]. Also, [27] develops an intuitionistic epistemic logic from the

ISSN 2007-9737

game-theoretical point of view. The intuitionistic aspect of the logic is required for describing the property of asynchronous communication among agents in distributed computing.

Jäger and Marti [15] formulate intuitionistic epistemic logic with distributed knowledge for the first time, as far as the authors know, and prove semantic completeness of Hilbert systems of intuitionistic $\mathbf{K}$ and $\mathbf{KT}$ with distributed knowledge. Logics we investigate in the present paper is basically based on theirs, but differs in the following respects: firstly, in our logics, distributed knowledge operator is parameterized by a group, i.e., a subset of whole agents, while [15] deals with only distributed knowledge for the whole agents. Secondly, we handle more axioms than [15], that is, we propose intuitionistic $\mathbf{K}$, $\mathbf{KT}$, $\mathbf{KD}$, $\mathbf{K4}$, $\mathbf{K4D}$, and $\mathbf{S4}$ with distributed knowledge. One point to note here. Axioms (K), (T), and (4) in our logics are simply a D_G-version of the respective axioms in the basic modal logic.

However, our axiom (D) is restricted to a single agent (i.e., $\neg D_{\{a\}}\bot$). This is because seriality for each R_a is generally not preserved under taking intersection among a group (refer to [2]), while reflexivity and transitivity are always preserved. As for proof of the semantic completeness, we adopt a more standard method via the concept of "pseudo-model" than [15]. We also propose cut-free sequent calculi for our logics, based on the idea introduced in [19] and prove Craig interpolation theorem by Maehara's method [16, 21]. Also, we establish decidability of the sequent calculi by the standard argument [8, 9] on a cut-free derivation of a sequent, while [15] does not show it for their Hilbert systems.

The paper is organized as follows. In Section 2, we introduce syntax and semantics for intuitionistic epistemic logic with distributed knowledge. Section 3 defines Hilbert systems of the logics, and state soundness results. In Section 4, strong completeness of the Hilbert systems of the logics is shown, via a notion of "pseudo-model".

In Section 5, we introduce sequent calculi for the logics and prove the cut-elimination theorem, Craig interpolation theorem, and decidability.

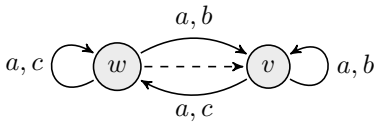


Fig. 1. Example of a frame

2 Syntax and Semantics of Intuitionistic Epistemic Logics with Distributed Knowledge Operators

We denote a finite set of agents by Agt. We call a *nonempty* subset of Agt "group" and denote it by G, H, etc. We denote by Grp the set of all groups, i.e., the set $\wp(\mathsf{Agt}) \setminus \{\varnothing\}$ of all non-empty subset of Agt. Let Prop be a countable set of propositional variables and Form be the set of formulas defined inductively by the following clauses:

$$\mathsf{Form} \ni \varphi ::= p \in \mathsf{Prop} \mid \bot \mid \varphi \to \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid D_G\varphi.$$

We read $D_G\varphi$ as "φ is distributed knowledge among a group G". We define $\neg\varphi$ as $\varphi \to \bot$ and the epistemic operator $K_a\varphi$ (read "agent a knows that φ") as $D_{\{a\}}\varphi$. As noted above, an expression of the form $D_\varnothing\varphi$ is *not* a well-formed formula, since we have excluded $\varnothing$ from our definition of groups.

We introduce Kripke semantics for intuitionistic multi-agent epistemic logic with distributed knowledge, along the lines of [15].

Definition 2.1 (Frame, Model)**.** A tuple $F = (W, \leqslant, (R_a)_{a\in\mathsf{Agt}})$ is a *frame* if: W is a set of states; $\leqslant$ is a preorder on W; $(R_a)_{a\in\mathsf{Agt}}$ is a family of binary relations on W, indexed by agents; and $\leqslant; R_a \subseteq R_a$ (for all $a \in \mathsf{Agt}$), where $R_1; R_2 := \{(x, z) \mid$ there exists y such that xR_1y and $yR_2z\}$.

A pair $M = (F, V)$ is a *model* if F is a frame, and a valuation function $V\colon \mathsf{Prop} \to \mathcal{P}(W)$ satisfies the heredity condition, i.e., if $w \in V(p)$ and $w \leqslant v$, then $v \in V(p)$. We denote an underlying set of states of a frame F or a model M by $|F|$ or $|M|$.

For a model $M = (W, \leqslant, (R_a)_{a\in\mathsf{Agt}}, V)$ and a state $w \in W$, a pair (M, w) is called a *pointed model*.

Satisfaction relation $M, w \Vdash \varphi$ on pointed models and formulas is defined recursively as follows:

$$
\begin{array}{lll}
M, w \Vdash p & \text{iff} & w \in V(p), \\
M, w \Vdash \bot & & \text{Never}, \\
M, w \Vdash \varphi \to \psi & \text{iff} & \text{for all } v \in W, \\
 & & \text{if } w \leqslant v \text{ then } M, v \nVdash \varphi \\
 & & \text{or } M, v \Vdash \psi, \\
M, w \Vdash \varphi \wedge \psi & \text{iff} & M, w \Vdash \varphi \text{ and } M, w \Vdash \psi, \\
M, w \Vdash \varphi \vee \psi & \text{iff} & M, w \Vdash \varphi \text{ or } M, w \Vdash \psi, \\
M, w \Vdash D_G\varphi & \text{iff} & \text{for all } v \in W, \\
 & & \text{if } (w, v) \in \bigcap_{a \in G} R_a \\
 & & \text{then } M, v \Vdash \varphi.
\end{array}
$$

It is noted from our definition of $K_a\varphi \coloneqq D_{\{a\}}\varphi$ that the satisfaction of $K_a\varphi$ at a state w of a model M is given as follows:

$$
\begin{array}{ll}
 & M, w \Vdash K_a\varphi, \\
\text{iff} & \text{for all } v \in W, \text{ if } (w, v) \in R_a \text{ then } M, v \Vdash \varphi.
\end{array}
$$

As is the case with ordinary intuitionistic logic, we have the following heredity property for a formula.

Proposition 2.2 (Heredity)**.** *If* $M, w \Vdash \varphi$ *and* $w \leqslant v$*, then* $M, v \Vdash \varphi$*.*

Proof. By induction on φ. For the case where $\varphi \equiv D_G\psi$, it is noted that the condition $\leqslant; R_a \subseteq R_a$ of a frame implies that $\leqslant; \bigcap_{a \in G} R_a \subseteq \bigcap_{a \in G} R_a$. □

Fig. 1 is an example of a frame. The preorder is depicted by a dotted arrow. Note that we omit reflexive arrows for the preorder. If a valuation is defined by, for example, $V(p) = \{v\}$ for any $p \in \mathsf{Prop}$, V satisfies the heredity condition. In this model, it can be seen that different groups have different distributed knowledge even at the same state. Indeed, $D_{\{a,b\}}p$ is true at w, but $D_{\{a,c\}}p$ is false at w. Further, we can also see that seriality for each agent's relation is not always preserved under taking intersection among a group. Namely, R_b and R_c are serial but $R_b \cap R_c$ is not in the example. This is why we should restrict (D) axiom to $\neg D_{\{a\}}\bot$, as defined in Table 1. Given a frame $F = (W, \leqslant, (R_a)_{a \in \mathsf{Agt}})$, we say that a formula φ is *valid* in F (notation: $F \Vdash \varphi$) if $(F, V), w \Vdash \varphi$ for every valuation function V and every $w \in W$. Moreover, a formula φ is valid in a class $\mathbb{F}$ of frames (notation: $\mathbb{F} \Vdash \varphi$) if $F \Vdash \varphi$ for every $F \in \mathbb{F}$.

Definition 2.3. A formula φ is a *semantic consequence* of Γ in a frame class $\mathbb{F}$ if for all frame $F \in \mathbb{F}$, a valuation V on F, a state $w \in |F|$, if $(F, V), w \Vdash \Gamma$, then $(F, V), w \Vdash \varphi$. We write it as "$\Gamma \Vdash_{\mathbb{F}} \varphi$".

3 Hilbert Systems

Hilbert systems for intuitionistic epistemic logics with D_G operators are constructed from axioms and rules shown in Table 1.

Table 1. Axioms and Rules for Hilbert-style Axiomatizations

Axioms and Rules for Intuitionistic Logic	
$(\mathbf{k})$	$\varphi \to (\psi \to \varphi)$
$(\mathbf{s})$	$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$
$(\vee\mathbf{i}_1)$	$\varphi \to (\varphi \vee \psi)$
$(\vee\mathbf{i}_2)$	$\psi \to (\varphi \vee \psi)$
$(\vee\mathbf{e})$	$(\varphi \to \chi) \to ((\psi \to \chi) \to ((\varphi \vee \psi) \to \chi))$
$(\wedge\mathbf{e}_1)$	$(\varphi \wedge \psi) \to \varphi$
$(\wedge\mathbf{e}_2)$	$(\varphi \wedge \psi) \to \psi$
$(\wedge\mathbf{i})$	$\varphi \to (\psi \to (\varphi \wedge \psi))$
$(\bot)$	$\bot \to \varphi$
$(\mathbf{MP})$	From φ and $\varphi \to \psi$, infer ψ
Axioms and Rules for H($\mathbf{IK}$)	
(Incl)	$D_G\varphi \to D_H\varphi \ \ (G \subseteq H)$
(K)	$D_G(\varphi \to \psi) \to (D_G\varphi \to D_G\psi)$
(Nec)	From φ, infer $D_G\varphi$
Additional Axioms for D_G operators	
(T)	$D_G\varphi \to \varphi$
(D)	$\neg D_{\{a\}}\bot$
(4)	$D_G\varphi \to D_G D_G\varphi$

A Hilbert system H($\mathbf{IK}$) consists of axioms and rules for intuitionistic logic, axioms (Incl) and (K), and a rule (Nec). Hilbert systems H($\mathbf{IKT}$), H($\mathbf{IKD}$), H($\mathbf{IK4}$), H($\mathbf{IK4D}$), and H($\mathbf{IS4}$) are defined as axiomatic expansions of H($\mathbf{IK}$) with (T), (D), (4), (4) and (D), and (T) and (4), respectively. Let $\mathbf{X}$ be any of $\mathbf{IK}$, $\mathbf{IKT}$, $\mathbf{IKD}$, $\mathbf{IK4}$, $\mathbf{IK4D}$, and $\mathbf{IS4}$ in what follows. The notion of provability in each system is defined as usual, and the fact that a formula φ is provable in H($\mathbf{X}$) is denoted by "$\vdash_{\mathsf{H}(\mathbf{X})} \varphi$". We also define derivability relation between a set Γ of formulas and a formula φ as below.

Definition 3.1. A formula φ is *derivable* from Γ in a logic $\mathbf{X}$ if $\vdash_{\mathsf{H}(\mathbf{X})} \bigwedge \Gamma' \to \varphi$ for some finite set Γ' which is a subset of Γ. We write it as "$\Gamma \vdash_{\mathsf{H}(\mathbf{X})} \varphi$".

We introduce a class of frames corresponding to each logic, in order to state soundness of our axiomatization.

Definition 3.2. A class of frames $\mathbb{F}(\mathbf{X})$ is defined as follows:

— $\mathbb{F}(\mathbf{IK})$ is the class of all frames.

— $\mathbb{F}(\mathbf{IKT})$ is the class of all frames such that R_a is reflexive ($a \in \mathsf{Agt}$).

— $\mathbb{F}(\mathbf{IKD})$ is the class of all frames such that R_a is serial ($a \in \mathsf{Agt}$).

— $\mathbb{F}(\mathbf{IK4})$ is the class of all frames such that R_a is transitive ($a \in \mathsf{Agt}$).

— $\mathbb{F}(\mathbf{IK4D})$ is the class of all frames such that R_a is transitive and serial ($a \in \mathsf{Agt}$).

— $\mathbb{F}(\mathbf{IS4})$ is the class of all frames such that R_a is reflexive and transitive ($a \in \mathsf{Agt}$).

Here, reflexivity, seriality, and transitivity are defined ordinarily.

We can prove the following soundness theorem by induction on φ. Note that axioms (T) and (4) are valid in reflexive and transitive frames, respectively, because if R_a is reflexive or transitive for any $a \in G$, $\bigcap_{a\in G} R_a$ is also reflexive or transitive, respectively.

Theorem 3.3. *If* $\vdash_{\mathsf{H}(\mathbf{X})} \varphi$*, then* $\mathbb{F}(\mathbf{X}) \Vdash \varphi$*.*

4 Completeness

In the present section, we explain a proof of the strong completeness theorem of our logic. Let Γ be a set of formulas and φ be a formula. The strong completeness theorem is stated as follows.

Theorem 4.1. *Let* $\mathbf{X}$ *be any of* $\mathbf{IK}$*,* $\mathbf{IKT}$*,* $\mathbf{IKD}$*,* $\mathbf{IK4}$*,* $\mathbf{IK4D}$*, and* $\mathbf{IS4}$*. Then, if* $\Gamma \Vdash_{\mathbb{F}(\mathbf{X})} \varphi$*, then* $\Gamma \vdash_{\mathsf{H}(\mathbf{X})} \varphi$*.*

As in [5], we show the theorem in two steps via the notion of "pseudo-model", that is, we first construct a canonical pseudo-model satisfying truth lemma, and then transform it into an equivalent pseudo-model which can be regarded as a model in the sense of Definition 2.1.

Definition 4.2 (Pseudo-frame, Pseudo-model)**.** A tuple $F = (W, \leqslant, (R_G)_{G\in\mathsf{Grp}})$ is a *pseudo-frame* if: $\leqslant; R_G \subseteq R_G$ for any $G \in \mathsf{Grp}$ and $R_H \subseteq R_G$ if $G \subseteq H$.

A pair $M = (F, V)$ is a *pseudo-model* if F is a pseudo-frame, and a valuation function $V\colon \mathsf{Prop} \to \mathcal{P}(W)$ satisfies the heredity condition, i.e., if $w \in V(p)$ and $w \leqslant v$, then $v \in V(p)$.

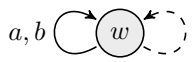


Fig. 2. Example of a pseudo-frame

Example 4.3. Fig. 2 is an example of a pseudo-frame. We name it F_{ex}. Note that $\{a\}$ is written as "a" and $R_{\{a,b\}}$ is defined as $\emptyset$ here. Since $R_{\{a,b\}} = \emptyset$, the condition of "$R_H \subseteq R_G$ if $G \subseteq H$" is self-evidently satisfied, i.e., $R_{\{a,b\}} \subseteq R_{\{a\}}$ and $R_{\{a,b\}} \subseteq R_{\{b\}}$. Note that $R_{\{a\}} \cap R_{\{b\}} \not\subseteq R_{\{a,b\}}$ in F_{ex}, while the contrary is guaranteed by the condition of "$R_H \subseteq R_G$ if $G \subseteq H$". Any frame can be regarded as a pseudo-frame with only relations for singleton groups, as in F_{ex}.

Definition 4.4 (Pseudo-satisfaction Relation)**.** For a pseudo-model M, a state $w \in |M|$, and a formula φ, a *pseudo-satisfaction relation* $M, w \Vdash^{ps} \varphi$ is defined the same as the satisfaction relation $\Vdash$, except for the clause for $D_G\varphi$: that is:

$$\begin{array}{ll} & M, w \Vdash^{ps} D_G\varphi, \\ \text{iff} & \text{for all } v \in W, \text{ if } (w, v) \in R_G \text{ then } M, v \Vdash^{ps} \varphi. \end{array}$$

Namely, in a pseudo-model, an operator D_G is treated like a primitive box operator, parameterized by a group.

Considering the definition of satisfaction relation for $D_G\varphi$, a pseudo-frame can be seen as a frame if the condition $R_G = \bigcap_{a\in G} R_{\{a\}}$ is satisfied for any group G.

So, we can prove the strong completeness theorem by transforming a canonical pseudo-model into a pseudo-model enjoying the condition above without changing satisfaction. We do this by a method of "tree unraveling".

4.1 Canonical Pseudo-Model

We define a canonical pseudo-model of our logics and state some properties of it in the present subsection. Since D_G operators are interpreted as primitive box-like operators indexed by a group in a pseudo-model, a canonical pseudo-model defined here is essentially the same as the canonical model of intuitionistic epistemic logics without distributed knowledge, which is described in detail e.g., in [17, Chapter 1]. Let $\mathbf{X}$ be any of $\mathbf{IK}$, $\mathbf{IKT}$, $\mathbf{IKD}$, $\mathbf{IK4}$, $\mathbf{IK4D}$, and $\mathbf{IS4}$ below.

Definition 4.5 (consistency)**.** A set Γ of formulas is $\mathbf{X}$-*consistent* if $\Gamma \nvdash_{\mathsf{H}(\mathbf{X})} \bot$.

Definition 4.6 (prime theory)**.** Γ is an $\mathbf{X}$-*prime* theory if:

1. Γ is prime, i.e., if $\varphi_1 \vee \varphi_2 \in \Gamma$, then $\varphi_1 \in \Gamma$ or $\varphi_2 \in \Gamma$.
2. Γ is a $\mathbf{X}$-theory, i.e., if $\Gamma \vdash_{\mathsf{H}(\mathbf{X})} \varphi$, then $\varphi \in \Gamma$.

The following are useful properties of a consistent and prime theory.

Lemma 4.7. *Let a set Γ of formulas be an $\mathbf{X}$-consistent and $\mathbf{X}$-prime theory:*

1. $\Gamma \vdash_{\mathsf{H}(\mathbf{X})} \varphi$ *iff* $\varphi \in \Gamma$.
2. *If* $\{\varphi, \varphi \to \psi\} \subseteq \Gamma$, *then* $\psi \in \Gamma$.
3. $\bot \notin \Gamma$.
4. $\varphi \wedge \psi \in \Gamma$ *iff* $\varphi \in \Gamma$ *and* $\psi \in \Gamma$.
5. $\varphi \vee \psi \in \Gamma$ *iff* $\varphi \in \Gamma$ *or* $\psi \in \Gamma$.
6. *If* $\varphi \to \psi \notin \Gamma$, *then* $\Gamma \cup \{\varphi\} \nvdash_{\mathsf{H}(\mathbf{X})} \psi$.
7. *If* $D_G\psi \notin \Gamma$, *then* $D_G^{-1}\Gamma \nvdash_{\mathsf{H}(\mathbf{X})} \psi$.

Lemma 4.8 (Lindenbaum)**.** *Let $\Gamma \cup \{\varphi\}$ be a set of formulas. If $\Gamma \nvdash_{\mathsf{H}(\mathbf{X})} \varphi$, then there is an $\mathbf{X}$-consistent and $\mathbf{X}$-prime theory Γ^+ such that $\Gamma \subseteq \Gamma^+$ and $\Gamma^+ \nvdash_{\mathsf{H}(\mathbf{X})} \varphi$.*

Definition 4.9. Given a set Γ of formulas, we define $D_G^{-1}\Gamma := \{\varphi \in \mathsf{Form} \mid D_G\varphi \in \Gamma\}$. A *canonical pseudo-model:*

$$M^{\mathbf{X}} = (W^{\mathbf{X}}, \leqslant^{\mathbf{X}}, (R_G^{\mathbf{X}})_{G\in\mathsf{Grp}}, V^{\mathbf{X}}),$$

is defined as follows:

— $W^{\mathbf{X}} :=$
$\{\Gamma \mid \Gamma$ is an $\mathbf{X}$-consistent and $\mathbf{X}$-prime theory$\}$.

— $\Gamma \leqslant^{\mathbf{X}} \Delta$ iff $\Gamma \subseteq \Delta$.

— $\Gamma R_G^{\mathbf{X}} \Delta$ iff $D_G^{-1}\Gamma \subseteq \Delta$.

— $V^{\mathbf{X}}(p) := \{\Gamma \in W^{\mathbf{X}} \mid p \in \Gamma\}$.

The definition is well-defined:

Proposition 4.10. $M^{\mathbf{X}}$ is *a pseudo-model.*

Lemma 4.11 (Truth Lemma)**.** *Let Γ be an $\mathbf{X}$-consistent and $\mathbf{X}$-prime theory. Then, $\varphi \in \Gamma$ if and only if $M^{\mathbf{X}}, \Gamma \Vdash^{ps} \varphi$.*

Proof. By induction on φ. We show the case $\varphi \equiv D_G\psi$. First, we show the left-to-right. Assume $D_G\psi \in \Gamma$ and fix any $\Delta \in W^{\mathbf{X}}$ such that $\Gamma R_G^{\mathbf{X}} \Delta$, i.e., $D_G^{-1}\Gamma \subseteq \Delta$. Clearly, $\psi \in \Delta$, and by the induction hypothesis, we have $M^{\mathbf{X}}, \Delta \Vdash \psi$. Next, We show the contraposition of the right-to-left. Assume $D_G\psi \notin \Gamma$. By item 7 of Lemma 4.7, and Lemma 4.8, there is an $\mathbf{X}$-consistent and $\mathbf{X}$-prime theory Δ such that $D_G^{-1}\Gamma \subseteq \Delta$ and $\Delta \nvdash_{\mathsf{H}(\mathbf{X})} \psi$. By item 1 of Lemma 4.7 and induction hypothesis, we have $M^{\mathbf{X}}, \Delta \nVdash \psi$, which shows $M^{\mathbf{X}}, \Gamma \nVdash D_G\psi$. □

For each axiom, the canonical pseudo-model satisfies the corresponding property on relations for D_G.

Proposition 4.12. *1. If $\mathbf{X}$ has the axiom (T), $R_G^{\mathbf{X}}$ is reflexive in $M^{\mathbf{X}}$.*

2. If $\mathbf{X}$ has the axiom (D), $R_{\{a\}}^{\mathbf{X}}$ is serial in $M^{\mathbf{X}}$.

3. If $\mathbf{X}$ has the axiom (4), $R_G^{\mathbf{X}}$ is transitive in $M^{\mathbf{X}}$.

Proof. We only show item 2. Fix any $\mathbf{X}$-consistent and $\mathbf{X}$-prime theory Γ. The aim is to find an $\mathbf{X}$-consistent and $\mathbf{X}$-prime theory Δ such that $D_{\{a\}}^{-1}\Gamma \subseteq \Delta$. By Lemma 4.8, it suffices to show $D_{\{a\}}^{-1}\Gamma \nvdash_{\mathsf{H}(\mathbf{X})} \bot$. Assuming the contrary, we have $\vdash_{\mathsf{H}(\mathbf{X})} \bigwedge_{i=1}^n \varphi_i \to \bot$ for some $\varphi_i \in D_{\{a\}}^{-1}\Gamma$. By (Nec), (K), and intuitionistic propositional tautologies, $\vdash_{\mathsf{H}(\mathbf{X})} \bigwedge_{i=1}^n D_{\{a\}}\varphi_i \to D_{\{a\}}\bot$. Since $D_{\{a\}}\varphi_i \in \Gamma$, it means $\Gamma \vdash_{\mathsf{H}(\mathbf{X})} D_{\{a\}}\bot$. However, we also have $\Gamma \vdash_{\mathsf{H}(\mathbf{X})} \neg D_{\{a\}}\bot$ by the assumption, which leads to contradiction by item 1 to 3 of Lemma 4.7. □

4.2 Tree Unraveling

We introduce a method called "tree unraveling", which transforms a pseudo-model into another pseudo-model satisfying $\bigcap_{a\in G} R_{\{a\}} = R_G$ (i.e., a model in the sense of Definition 2.1). Our definitions below are intuitionistic generalizations of definitions proposed in [5] over classical logic.

Definition 4.13. Let $M = (W, \leqslant, (R_G)_{G\in\mathsf{Grp}}, V)$ be a pseudo-model. A pseudo-model $M' = (W', \leqslant \cap(W'\times W'), (R_G \cap (W'\times W'))_{G\in\mathsf{Grp}}, V')$ is a *generated submodel* of M if: $W' \subseteq W$; If $w \in W'$ and $w \leqslant w'$ then $w' \in W'$; If $w \in W'$ and wR_Gw' then $w' \in W'$; and $V'(p) = V(p) \cap W'$ for any $p \in \mathsf{Prop}$.

For $X \subseteq |M|$, we define M_X as the smallest generated submodel containing X. If $M = M_X$, we say that M is generated by X.

Definition 4.14. Let $M = (F, V)$ be a pseudo-model generated by $w \in W$, where $F = (W, \leqslant, (R_G)_{G\in\mathsf{Grp}})$:

— We put $w_0 := w$ and define $\mathrm{Finpath}(F, w)$ as
$\{\langle w_0, L_1, w_1, L_2, \cdots, L_n, w_n\rangle \mid n \geq 0,$
$L_i \in \{\leqslant, R_G\}_{G\in\mathsf{Grp}}, w_{i-1}L_iw_i$ for all $1 \leq i \leq n\}$.
We call an element of $\mathrm{Finpath}(F, w)$ "a path (from a state w)" and denote it by $\overrightarrow{u}, \overrightarrow{v}$, etc.

— For $\overrightarrow{u} = \langle w_0, L_1, w_1, L_2, \cdots, L_{n-1}, w_{n-1}, L_n, w_n\rangle \in \mathrm{Finpath}(F, w)$, $\mathsf{tail}(\overrightarrow{u})$ is defined as w_n.

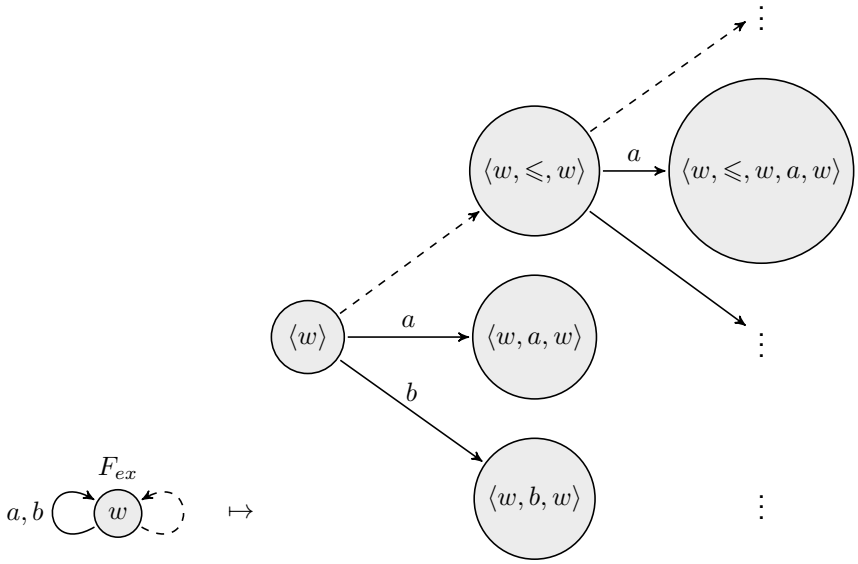


Fig. 3. Tree unraveling

— We say that paths $\overrightarrow{u}, \overrightarrow{v} \in \mathrm{Finpath}(F, w)$ satisfy a relation $\overrightarrow{u} \preccurlyeq \overrightarrow{v}$ if and only if $\overrightarrow{v} \equiv \overrightarrow{u}^\frown\langle \leqslant, w' \rangle$, where $^\frown$ means concatenation of two tuples.

— We say that paths $\overrightarrow{u}, \overrightarrow{v} \in \mathrm{Finpath}(F, w)$ satisfy a relation $\overrightarrow{u} \mathcal{R}_G \overrightarrow{v}$ if and only if $\overrightarrow{v} \equiv \overrightarrow{u}^\frown\langle R_H, w' \rangle$ and $G \subseteq H$.

— A *valuation* $\mathcal{V}\colon \mathsf{Prop} \to \mathcal{P}(\mathrm{Finpath}(F, w))$ is defined by:

$$\mathcal{V}(p) = \{ \overrightarrow{u} \in \mathrm{Finpath}(W, w) \mid \mathsf{tail}(\overrightarrow{u}) \in V(p) \}\,.$$

Take F_{ex} in Fig. 2 as an example. The set $\mathrm{Finpath}(F_{ex}, w)$ of paths on F_{ex} and $\preccurlyeq$ and $\mathcal{R}_G$ on this set are drawn in Fig. 3. The point is that the a-arrow and b-arrow on w in F_{ex} are transformed into two arrows with different destinations, so that the condition "$R_{\{a\}} \cap R_{\{b\}} = R_{\{a,b\}}$" is not satisfied in F_{ex} but becomes satisfied in $\mathrm{Finpath}(F_{ex}, w)$. However, as it is, $(\mathrm{Finpath}(F_{ex}, w), \preccurlyeq, (\mathcal{R}_G)_{G \in \mathsf{Grp}})$ is not a pseudo-frame, since $\preccurlyeq$ itself is not a preorder and the condition "$\leqslant ; R_G \subseteq R_G$" is not satisfied because, for example, there is no a-arrow from $\langle w \rangle$ to $\langle w, \leqslant, w, a, w \rangle$. Therefore, a preorder and relations for D_G on $\mathrm{Finpath}(F, w)$ in general should be defined as follows.

Definition 4.15 (Tree Unraveling)**.** Let $M = (F, V)$ be a pseudo-model generated by $w \in W$, where $F = (W, \leqslant, (R_G)_{G \in \mathsf{Grp}})$. A *tree unraveling pseudo-model* $\mathrm{Tree}(M, w)$ of a pointed pseudo-model (M, w) is defined as a tuple:

$$(\mathrm{Finpath}(F, w), \preccurlyeq^*, (\preccurlyeq^* ; \mathcal{R}_G)_{G \in \mathsf{Grp}}, \mathcal{V})\,,$$

where R^* is defined as the reflexive and transitive closure of a relation R.

We can easily show that $\mathrm{Tree}(M, w)$ is indeed a pseudo-model. Moreover, as explained above with Fig. 3, $\bigcap_{a \in G} \mathcal{R}_{\{a\}} = \mathcal{R}_G$ holds, from which it is also shown that $\bigcap_{a \in G} \preccurlyeq^* ; \mathcal{R}_{\{a\}} = \preccurlyeq^* ; \mathcal{R}_G$ by a simple argument using property of a tree unraveling pseudo-model. Therefore, $\mathrm{Tree}(M, w)$ can be seen as a model in the sense of Definition 2.1. The following is a key property of tree unraveling.

Lemma 4.16. *Let $M = (F, V)$ be a pseudo-model generated by $w \in W$, where $F = (W, \leqslant, (R_G)_{G \in \mathsf{Grp}})$. Then, $M, w \Vdash^{ps} \varphi$ iff $\mathrm{Tree}(M, w), \langle w \rangle \Vdash^{ps} \varphi$ for any formula φ.*

Proof. The function $\overrightarrow{u} \mapsto \mathsf{tail}(\overrightarrow{u})$ is a bounded morphism (which takes not only relations for D_G but also a preorder into account) from $\mathrm{Tree}(M, w)$ to M. □

We end the present section by proving Theorem 4.1.

Proof. (Outline) First, we show the case of $\mathbf{IK}$. We show the contraposition. Assume $\Gamma \not\vdash_{\mathsf{H}(\mathbf{X})} \varphi$. By Lemma 4.8, We can find an $\mathbf{X}$-prime and $\mathbf{X}$-consistent theory Γ^+ such that $\Gamma \subseteq \Gamma^+$ and $\Gamma^+ \not\vdash_{\mathsf{H}(\mathbf{X})} \varphi$. Since $\Gamma \subseteq \Gamma^+$, $M^{\mathbf{X}}, \Gamma^+ \Vdash^{ps} \Gamma$ by the left-to-right of Lemma 4.11. On the other hand, $M^{\mathbf{X}}, \Gamma^+ \not\Vdash^{ps} \varphi$ by the right-to-left of Lemma 4.11 and item 1 of Lemma 4.7. We can take $\mathrm{Tree}\left(M^{\mathbf{X}}_{\Gamma^+}, \Gamma^+\right)$, because, by Proposition 4.10, $M^{\mathbf{X}}_{\Gamma^+}$ is a pseudo-model generated by Γ^+. Since any tree unraveling pseudo-model can be seen as a model in the sense of Definition 2.1, it suffices to show that $(M^{\mathbf{X}}, \Gamma^+)$ satisfies exactly the same formulas as $\left(\mathrm{Tree}\left(M^{\mathbf{X}}_{\Gamma^+}, \Gamma^+\right), \langle\Gamma^+\rangle\right)$. First, $(M^{\mathbf{X}}, \Gamma^+)$ satisfies exactly the same formulas as $(M^{\mathbf{X}}_{\Gamma^+}, \Gamma^+)$. Then, by Lemma 4.16, $(M^{\mathbf{X}}_{\Gamma^+}, \Gamma^+)$ satisfies exactly the same formulas as $\left(\mathrm{Tree}\left(M^{\mathbf{X}}_{\Gamma^+}, \Gamma^+\right), \langle\Gamma^+\rangle\right)$.

For the remaining logics, basically, a similar argument can be applied, but definitions and proofs become more involved. In order to make relations for D_G have the desired property, such as reflexivity or transitivity, the relation $\preccurlyeq^*; \mathcal{R}_G$ should be replaced by $\preccurlyeq^*; \mathcal{R}^\circ_G$, $(\preccurlyeq^*; \mathcal{R}^+_G)^+$, and $(\preccurlyeq^*; \mathcal{R}^*_G)^*$ for $\mathbf{IKT}$, $\mathbf{IK4}$ and $\mathbf{IK4D}$, and $\mathbf{IS4}$, respectively, in the definition of tree unraveling. Here, R° and R^+ are defined as the reflexive closure and transitive closure of a relation R, respectively. Also, note that $\preccurlyeq^*; \mathcal{R}_G$ and $\left(\preccurlyeq^*; \mathcal{R}^+_G\right)^+$ are serial if R_G is serial and that $R^{\mathbf{X}}_{\{a\}}$ is serial if $\mathbf{X}$ has the axiom (D) (by Proposition 4.12). The resulting tree unravelings are also easily shown to be pseudo-models. The condition "$\bigcap_{a\in G} R_{\{a\}} = R_G$" in the tree unraveling pseudo-models also can be shown to be satisfied, by using the property of a tree unraveling pseudo-model. Therefore, from the above argument, $\mathrm{Tree}\left(M^{\mathbf{X}}_{\Gamma^+}, \Gamma^+\right)$ can be seen as a model, whose underlying frame is an element of $\mathbb{F}(\mathbf{X})$. The fact that the function $\overrightarrow{u} \mapsto \mathsf{tail}(\overrightarrow{u})$ is a bounded morphism also in the respective tree unraveling pseudo-models is needed, and can be shown straightforwardly. □

5 Sequent Calculi of Intuitionistic Epistemic Logics with Distributed Knowledge

5.1 Equipollence and Cut-Elimination

A *sequent* is a pair of finite multisets of formulas Γ and Δ denoted by "$\Gamma \Rightarrow \Delta$", where $\#\Delta \leq 1$. The multiset Γ is called an"antecedent" of a sequent $\Gamma \Rightarrow \Delta$, and Δ a "succedent". A sequent is intuitively interpreted as "if all formulas in Γ hold, then a formula in Δ holds." The reason why the number of Δ is restricted is that we build our calculus on the basis of Gentzen's **LJ** [8, 9] for intuitionistic propositional logic. Our sequent calculi for the intuitionistic epistemic logics with distributed knowledge are presented in Table 2. Axioms, structural rules, and propositional logical rules are common to $\mathbf{LJ}$. The other rules are the same as the ones in [19], except that rules for (D) axiom, i.e., $(D_{\mathbf{IKD}})$ and $(D_{\mathbf{IK4D}})$ are added, in order to construct calculi for the logics $\mathbf{IKD}$ and $\mathbf{IK4D}$.

We note that when $n = 0$, e.g., in the rule (D) of Table 2, the multiset is regarded as the empty multiset and thus $\bigcup_{i=1}^n G_i$ is regarded as $\varnothing$. A sequent $\Gamma \Rightarrow \Delta$ is *derivable* in each calculus $\mathsf{G}(\mathbf{X})$ if there exists a finite tree of sequents, whose root is $\Gamma \Rightarrow \Delta$ and each node of which is inferred by some rule (including axioms) in $\mathsf{G}(\mathbf{X})$. We write it as $\vdash_{\mathsf{G}(\mathbf{X})} \Gamma \Rightarrow \Delta$.

Example 5.1. The following is an application of rule (D), which captures typical inference involving distributed knowledge mentioned in Introduction:

$$\frac{p \to q, p \Rightarrow q}{D_{\{a\}}(p \to q), D_{\{b\}}p \Rightarrow D_{\{a,b\}}q}\ (D).$$

We note that for any logic $\mathbf{X}$ under consideration, $\mathsf{H}(\mathbf{X})$ and $\mathsf{G}(\mathbf{X})$ are equipollent in the following sense.

Theorem 5.2 (Equipollence)**.** *Let* $\mathbf{X}$ *be any of* $\mathbf{IK}$, $\mathbf{IKT}$, $\mathbf{IKD}$, $\mathbf{IK4}$, $\mathbf{IK4D}$, *and* $\mathbf{IS4}$. *Then, the following hold. 1. If* $\vdash_{\mathsf{H}(\mathbf{X})} \varphi$, *then* $\vdash_{\mathsf{G}(\mathbf{X})} \Rightarrow \varphi$. *2. If* $\vdash_{\mathsf{G}(\mathbf{X})} \Gamma \Rightarrow \Delta$, *then* $\vdash_{\mathsf{H}(\mathbf{X})} \bigwedge \Gamma \to \bigvee \Delta$, *where* $\bigwedge \varnothing := \top$ *and* $\bigvee \varnothing := \bot$.

Proof. We show the case of $\mathbf{IK}$. The idea for proof is common to the rest. Here we focus on item 2 alone. We show item 2 by induction on the structure of the derivation for the sequent $\Gamma \Rightarrow \Delta$. We deal with the case for the rule (D) only. Suppose we have a derivation:

$$\frac{\dfrac{\mathcal{D}}{\varphi_1, \ldots, \varphi_n \Rightarrow \psi} \quad \left(\bigcup_{i=1}^n G_i \subseteq G\right)}{D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow D_G\psi}\ (D).$$

We show $\vdash_{\mathsf{H}(\mathbf{X})} \bigwedge_{i=1}^n D_{G_i}\varphi_i \to D_G\psi$. We have $\vdash_{\mathsf{H}(\mathbf{X})} \bigwedge_{i=1}^n \varphi_i \to \psi$ as the induction hypothesis for the derivation $\mathcal{D}$. From this, we can infer by necessitation $\vdash_{\mathsf{H}(\mathbf{X})} D_G(\bigwedge_{i=1}^n \varphi_i \to \psi)$. By this and axiom (K), we have $\vdash_{\mathsf{H}(\mathbf{X})} D_G(\bigwedge_{i=1}^n \varphi_i) \to D_G\psi$, which is equivalent to $\vdash_{\mathsf{H}(\mathbf{X})} \bigwedge_{i=1}^n D_G\varphi_i \to D_G\psi$. Therefore, it suffices to show that $\vdash_{\mathsf{H}(\mathbf{X})} \bigwedge_{i=1}^n D_{G_i}\varphi_i \to \bigwedge_{i=1}^n D_G\varphi_i$, which is equivalent to $\vdash_{\mathsf{H}(\mathbf{X})} \bigwedge_{i=1}^n D_{G_i}\varphi_i \to D_G\varphi_i$ for any $i \in \{1, \ldots, n\}$. This is evident because we have a theorem in intuitionistic propositional logic $\vdash_{\mathsf{H}(\mathbf{X})} \bigwedge_{i=1}^n D_{G_i}\varphi_i \to D_{G_i}\varphi_i$ and the axiom (Incl) $\vdash_{\mathsf{H}(\mathbf{X})} D_{G_i}\varphi_i \to D_G\varphi_i$. □

Table 2. Sequent Calculi for $\mathbf{IK}$, $\mathbf{IKT}$, $\mathbf{IKD}$, $\mathbf{IK4}$, $\mathbf{IK4D}$, and $\mathbf{IS4}$

Axioms

$$\frac{}{\varphi \Rightarrow \varphi}\ (Id) \qquad \frac{}{\bot \Rightarrow}\ (\bot)$$

Structural Rules

$$\frac{\Gamma \Rightarrow}{\Gamma \Rightarrow \varphi}\ (\Rightarrow w) \qquad \frac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\ (w \Rightarrow) \qquad \frac{\varphi, \varphi, \Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\ (c \Rightarrow)$$

$$\frac{\Gamma \Rightarrow \varphi \quad \varphi, \Pi \Rightarrow \Sigma}{\Gamma, \Pi \Rightarrow \Sigma}\ (Cut)$$

Propositional Logical Rules

$$\frac{\varphi, \Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \to \psi}\ (\Rightarrow\to) \qquad \frac{\Gamma_1 \Rightarrow \varphi \quad \psi, \Gamma_2 \Rightarrow \Delta}{\varphi \to \psi, \Gamma_1, \Gamma_2 \Rightarrow \Delta}\ (\to\Rightarrow)$$

$$\frac{\Gamma \Rightarrow \varphi \quad \Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \wedge \psi}\ (\Rightarrow \wedge)$$

$$\frac{\varphi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta}\ (\wedge \Rightarrow^1) \qquad \frac{\psi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta}\ (\wedge \Rightarrow^2)$$

$$\frac{\Gamma \Rightarrow \varphi}{\Gamma \Rightarrow \varphi \vee \psi}\ (\Rightarrow \vee^1) \qquad \frac{\Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \vee \psi}\ (\Rightarrow \vee^2)$$

$$\frac{\varphi, \Gamma \Rightarrow \Delta \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \vee \psi, \Gamma \Rightarrow \Delta}\ (\vee \Rightarrow)$$

Logical Rules for D_G of $\mathbf{IK}$

$$\frac{\varphi_1, \ldots, \varphi_n \Rightarrow \psi \quad (\bigcup_{i=1}^n G_i \subseteq G)}{D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow D_G\psi}\ (D)$$

Logical Rules for D_G of $\mathbf{IKT}$

$$\frac{\varphi_1, \ldots, \varphi_n \Rightarrow \psi \quad (\bigcup_{i=1}^n G_i \subseteq G)}{D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow D_G\psi}\ (D) \qquad \frac{\varphi, \Gamma \Rightarrow \Delta}{D_G\varphi, \Gamma \Rightarrow \Delta}\ (D \Rightarrow)$$

Logical Rules for D_G of $\mathbf{IKD}$

$$\frac{\varphi_1, \ldots, \varphi_n \Rightarrow \psi \quad (\bigcup_{i=1}^n G_i \subseteq G)}{D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow D_G\psi}\ (D) \qquad \frac{\Gamma \Rightarrow}{D_{\{a\}}\Gamma \Rightarrow}\ (D_{\mathbf{IKD}})$$

Logical Rules for D_G of $\mathbf{IK4}$

$$\frac{\varphi_1, \ldots, \varphi_n, D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow \psi \quad (\bigcup_{i=1}^n G_i \subseteq G)}{D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow D_G\psi}\ (\Rightarrow D_{\mathbf{IK4}})$$

Logical Rules for D_G of $\mathbf{IK4D}$

$$\frac{\varphi_1, \ldots, \varphi_n, D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow \psi \quad (\bigcup_{i=1}^n G_i \subseteq G)}{D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow D_G\psi}\ (\Rightarrow D_{\mathbf{IK4}})$$

$$\frac{\Gamma, D_{\{a\}}\Gamma \Rightarrow}{D_{\{a\}}\Gamma \Rightarrow}\ (\Rightarrow D_{\mathbf{IK4D}})$$

Logical Rules for D_G of $\mathbf{IS4}$

$$\frac{D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow \psi \quad (\bigcup_{i=1}^n G_i \subseteq G)}{D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow D_G\psi}\ (\Rightarrow D_{\mathbf{IS4}})$$

$$\frac{\varphi, \Gamma \Rightarrow \Delta}{D_G\varphi, \Gamma \Rightarrow \Delta}\ (D \Rightarrow)$$

We have the cut-elimination theorem for all of the logics in consideration.

Theorem 5.3 (Cut-Elimination)**.** *Let $\mathbf{X}$ be any of $\mathbf{IK}$, $\mathbf{IKT}$, $\mathbf{IKD}$, $\mathbf{IK4}$, $\mathbf{IK4D}$, and $\mathbf{IS4}$. Then, the following holds: If $\vdash_{\mathsf{G}(\mathbf{X})} \Gamma \Rightarrow \Delta$, then $\vdash_{\mathsf{G}^-(\mathbf{X})} \Gamma \Rightarrow \Delta$, where $\mathsf{G}^-(\mathbf{X})$ denotes a system "$\mathsf{G}(\mathbf{X})$ minus the cut rule".*

Proof. First, we introduce a notion of "principal formula". A principal formula is defined for each inference rule, except for the axioms and (Cut) rule and is informally expressed as "a formula, on which the inference rule acts".

Definition 5.4. A *principal formula* of the structural rules, the propositional logical rules, and the rule $(D \Rightarrow)$ is a formula appearing in the lower sequent, which is not contained in Γ_1, Γ_2, Γ, or Δ. A *principal formula* of the rules for D_G operator other than $(D \Rightarrow)$ is every formula in the lower sequent.

To prove the theorem, we consider a system $\mathsf{G}^*(\mathbf{X})$, in which the cut rule is replaced by a "extended" cut rule defined as:

$$\frac{\Gamma \Rightarrow \varphi^n \quad \varphi^m, \Sigma \Rightarrow \Theta}{\Gamma, \Sigma \Rightarrow \Theta}\ (ECut),$$

where φ^n denotes the multi-set of n-copies of φ and $n = 0, 1$ and $m \geq 0$. Since $(ECut)$ is the same as (Cut) when we set $n = m = 1$, it is obvious that if $\vdash_{\mathsf{G}(\mathbf{X})} \Gamma \Rightarrow \Delta$, then $\vdash_{\mathsf{G}^*(\mathbf{X})} \Gamma \Rightarrow \Delta$, so it suffices to show that if $\vdash_{\mathsf{G}^*(\mathbf{X})} \Gamma \Rightarrow \Delta$, then $\vdash_{\mathsf{G}^-(\mathbf{X})} \Gamma \Rightarrow \Delta$.

Suppose $\vdash_{\mathsf{G}^*(\mathbf{X})} \Gamma \Rightarrow \Delta$ and fix one derivation for the sequent. To obtain an $(ECut)$-free derivation of $\Gamma \Rightarrow \Delta$, it is enough to concentrate on a derivation whose root is derived by $(ECut)$ and which has no other application of $(ECut)$. In what follows, we let $\mathbf{X}$ be $\mathbf{IK}$. Let us suppose that $\mathcal{D}$ has the following structure:

$$\dfrac{\dfrac{\mathcal{L}}{\Gamma \Rightarrow \varphi^n}\ (\mathbf{rule}_{\mathcal{L}}) \quad \dfrac{\mathcal{R}}{\varphi^m, \Sigma \Rightarrow \Theta}\ (\mathbf{rule}_{\mathcal{R}})}{\Gamma, \Sigma \Rightarrow \Theta}\ (ECut),$$

where the derivations $\mathcal{L}$ and $\mathcal{R}$ has no application of $(ECut)$ and $\mathbf{rule}_{\mathcal{L}}$ and $\mathbf{rule}_{\mathcal{R}}$ are meta-variables for the name of rule applied there. Let the number of logical symbols (including D_G) appearing in φ be $c(\mathcal{D})$, and the number of sequents in $\mathcal{L}$ and $\mathcal{R}$ be $w(\mathcal{D})$. We show the lemma by double induction on $(c(\mathcal{D}), w(\mathcal{D}))$. If $n = 0$ or $m = 0$, we can derive the root sequent of $\mathcal{D}$ without using $(ECut)$ by weakening rules. So, in what follows

we assume $n = 1$ and $m > 0$. Then, it is sufficient to consider the following four cases: [1]

1. $\mathbf{rule}_{\mathcal{L}}$ or $\mathbf{rule}_{\mathcal{R}}$ is an axiom.
2. $\mathbf{rule}_{\mathcal{L}}$ or $\mathbf{rule}_{\mathcal{R}}$ is a structural rule.
3. $\mathbf{rule}_{\mathcal{L}}$ or $\mathbf{rule}_{\mathcal{R}}$ is a logical rule and a cut formula φ is not principal (in the sense we have specified above) for that rule.
4. $\mathbf{rule}_{\mathcal{L}}$ and $\mathbf{rule}_{\mathcal{R}}$ are both logical rules (including (D)) for the same logical symbol and a cut formula φ is principal for each rule.

We concentrate on a rule (D) and the case involving the rule (D) is case 4 only, so we only comment on case 4 where both $\mathbf{rule}_{\mathcal{L}}$ and $\mathbf{rule}_{\mathcal{R}}$ are rules (D). In that case, the given derivation $\mathcal{D}$ has the following structure:

$$\dfrac{\dfrac{\mathcal{L}}{\Gamma \Rightarrow D_G\psi} \quad \dfrac{\mathcal{R}}{(D_G\psi)^m, \Sigma \Rightarrow D_H\chi}}{\Gamma, \Sigma \Rightarrow D_H\chi}\ (ECut),$$

where

$$\mathcal{L} \equiv \dfrac{\dfrac{\mathcal{L}'}{\varphi_1, \ldots, \varphi_n \Rightarrow \psi} \quad (\bigcup_{i=1}^{n} G_i \subseteq G),}{D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow D_G\psi}\ (D)$$

and

$$\mathcal{R} \equiv \dfrac{\dfrac{\mathcal{R}'}{\psi^m, \psi_1, \ldots, \psi_m \Rightarrow \chi} \quad (G \cup \bigcup_{j=1}^{m} H_j \subseteq H)}{(D_G\psi)^m, D_{H_1}\psi_1, \ldots, D_{H_m}\psi_m \Rightarrow D_H\chi}\ (D).$$

The derivation $\mathcal{D}$ can be transformed into the following derivation $\mathcal{E}$:

$$\dfrac{\dfrac{\dfrac{\mathcal{L}'}{\varphi_1, \ldots, \varphi_n \Rightarrow \psi} \quad \dfrac{\mathcal{R}'}{\psi^m, \psi_1, \ldots, \psi_m \Rightarrow \chi}}{\varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m \Rightarrow \chi}\ (ECut)}{D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n, D_{H_1}\psi_1, \ldots, D_{H_m}\psi_m \Rightarrow D_H\chi}\ (D),$$

where the rule (D) is applicable because we have $\bigcup_{i=1}^{n} G_i \cup \bigcup_{j=1}^{m} H_j \subseteq H$ by $\bigcup_{i=1}^{n} G_i \subseteq G$ and $G \cup \bigcup_{j=1}^{m} H_j \subseteq H$. We call $\mathcal{E}'$ its subderivation whose root sequent is $\varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m \Rightarrow \chi$. The derivation $\mathcal{E}'$ have no application of $(ECut)$ and $c(\mathcal{E}') < c(\mathcal{D})$. Hence, by induction hypothesis, there exists an $(ECut)$-free derivation $\tilde{\mathcal{E}}'$ having the same root sequent. Replacing the derivation $\mathcal{E}'$ by $\tilde{\mathcal{E}}'$ in $\mathcal{E}$, we obtain an $(ECut)$-free derivation for the sequent $D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n, D_{H_1}\psi_1, \ldots, D_{H_m}\psi_m \Rightarrow D_H\chi$ as required. □

[1] In case 4, we assume the condition for both rule applications, because if the one of the two rule applications does not satisfy the condition, the whole derivation should be categorized into one of the rest cases.

The following subformula property is an important corollary of the cut-elimination theorem, and later used in a proof of decidability.

Corollary 5.5 (Subformula Property)**.** *Let* $\mathbf{X}$ *be any of* $\mathbf{IK}$*,* $\mathbf{IKT}$*,* $\mathbf{IKD}$*,* $\mathbf{IK}4$*,* $\mathbf{IK}4\mathbf{D}$*, and* $\mathbf{IS}4$ *and suppose* $\vdash_{\mathsf{G}(\mathbf{X})} \Gamma \Rightarrow \Delta$*. Then, there exists a derivation of* $\Gamma \Rightarrow \Delta$ *satisfying a condition that any formula occurring in the derivation is a subformula of certain formula in* Γ *or* Δ*.*

Proof. A cut-free derivation of $\Gamma \Rightarrow \Delta$ satisfies the condition, because any formula in the upper sequent is a subformula of certain formula in the lower sequent in every inference rules of our calculi except (Cut). □

5.2 Craig Interpolation Theorem and Decidability

In many logics, the Craig interpolation theorem can be derived as an application of the cut-elimination theorem, using a Maehara method originally described in [16]. An application of the method to basic modal logic can also be found in [21]. Unlike [19], the concept of 'partition' is simplified, because we do not allow multiple formulas to appear in the succedent of a sequent.

Definition 5.6 (Partition)**.** A *partition* for a sequent $\Gamma \Rightarrow \Delta$ is defined as a tuple $\langle \Gamma_1; \Gamma_2 \rangle$, such that $\Gamma = \Gamma_1, \Gamma_2$.

Definition 5.7. For a formula φ, $\mathsf{Prop}(\varphi)$ is defined as the set of all propositional variables appearing in φ. For a multiset Γ of formulas, $\mathsf{Prop}(\Gamma)$ is defined as $\bigcup_{\varphi \in \Gamma} \mathsf{Prop}(\varphi)$. Similarly, $\mathsf{Agt}(\varphi)$ is defined as the set of agents appearing in φ and $\mathsf{Agt}(\Gamma)$ as $\bigcup_{\varphi \in \Gamma} \mathsf{Agt}(\varphi)$

The following is a key lemma for Craig Interpolation Theorem.

Lemma 5.8. *Let* $\mathbf{X}$ *be any of* $\mathbf{IK}$*,* $\mathbf{IKT}$*,* $\mathbf{IKD}$*,* $\mathbf{IK}4$*,* $\mathbf{IK}4\mathbf{D}$*, and* $\mathbf{IS}4$*. Suppose* $\vdash_{\mathsf{G}(\mathbf{X})} \Gamma \Rightarrow \Delta$*. Then, for any partition* $\langle \Gamma_1; \Gamma_2 \rangle$ *for the sequent* $\Gamma \Rightarrow \Delta$*, there exists a formula* φ *called "interpolant", satisfying the following:*

1. $\vdash_{\mathsf{G}(\mathbf{X})} \Gamma_1 \Rightarrow \varphi$ *and* $\vdash_{\mathsf{G}(\mathbf{X})} \varphi, \Gamma_2 \Rightarrow \Delta$*.*
2. $\mathsf{Prop}(\varphi) \subseteq \mathsf{Prop}(\Gamma_1) \cap \mathsf{Prop}(\Gamma_2, \Delta)$*.*
3. $\mathsf{Agt}(\varphi) \subseteq \mathsf{Agt}(\Gamma_1) \cap \mathsf{Agt}(\Gamma_2, \Delta)$*.*

Proof. We prove the case of $\mathbf{IK}$ by induction on the structure of a derivation for $\Gamma \Rightarrow \Delta$. Fix the derivation and name it $\mathcal{D}$. By Theorem 5.3, we can assume that $\mathcal{D}$ is cut-free. We treat only the case of (D) below (for

other cases, the reader is referred to [21]). Suppose $\mathcal{D}$ is of the form

$$\dfrac{\dfrac{\mathcal{E}}{\varphi_1, \ldots, \varphi_n \Rightarrow \psi} \quad (\bigcup_{i=1}^{n} G_i \subseteq G)}{D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow D_G\psi}\ (D).$$

A partition of $D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow D_G\psi$ is of the following form:

$$\langle D_{G_1}\varphi_1, \ldots, D_{G_k}\varphi_k; D_{G_{k+1}}\varphi_{k+1}, \ldots, D_{G_n}\varphi_n\rangle.$$

The induction hypothesis on $\mathcal{E}$ for a partition $\langle \varphi_1, \ldots, \varphi_k; \varphi_{k+1}, \ldots, \varphi_n\rangle$ is used. That is, we have derivations for $\varphi_1, \ldots, \varphi_k \Rightarrow \chi$ and $\chi, \varphi_{k+1}, \ldots, \varphi_n \Rightarrow \psi$ for some formula χ. If $k > 0$, we can choose $D_{\bigcup_{i=1}^{k} G_i}\chi$ as a required interpolant, because we have following derivations:

$$\dfrac{\dfrac{\text{I.H.}}{\varphi_1, \ldots, \varphi_k \Rightarrow \chi} \quad (\bigcup_{i=1}^{k} G_i \subseteq \bigcup_{i=1}^{k} G_i)}{D_{G_1}\varphi_1, \ldots, D_{G_k}\varphi_k \Rightarrow D_{\bigcup_{i=1}^{k} G_i}\chi}\ (D)$$

$$\dfrac{\dfrac{\text{I.H.}}{\chi, \varphi_{k+1}, \ldots, \varphi_n \Rightarrow \psi} \quad (\bigcup_{i=1}^{k} G_i \cup \bigcup_{i=k+1}^{n} G_i = \bigcup_{i=1}^{n} G_i \subseteq G)}{D_{\bigcup_{i=1}^{k} G_i}\chi, D_{G_{k+1}}\varphi_{k+1}, \ldots, D_{G_n}\varphi_n \Rightarrow D_G\psi}\ (D).$$

Furthermore, the interpolant enjoys the condition 2 and 3 as induction hypothesis and simple calculation show. If $k = 0$, we can choose χ as an interpolant, since we have the following derivations:

$$\dfrac{\dfrac{\mathcal{E}}{\varphi_1, \ldots, \varphi_n \Rightarrow \psi} \quad (\bigcup_{i=1}^{n} G_i \subseteq G)}{D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow D_G\psi}\ (D)$$
$$\dfrac{\text{I.H.}}{\Rightarrow \chi} \qquad \dfrac{}{\chi, D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow D_G\psi}\ (w\Rightarrow) \qquad \square$$

Theorem 5.9 (Craig Interpolation Theorem)**.** *Let* $\mathbf{X}$ *be any of* $\mathbf{IK}$*,* $\mathbf{IKT}$*,* $\mathbf{IKD}$*,* $\mathbf{IK4}$*,* $\mathbf{IK4D}$*, and* $\mathbf{IS4}$*. Given that* $\vdash_{\mathsf{G}(\mathbf{X})} \varphi \Rightarrow \psi$*, there exists a formula* χ *satisfying the following conditions:*

1. $\vdash_{\mathsf{G}(\mathbf{X})} \varphi \Rightarrow \chi$ *and* $\vdash_{\mathsf{G}(\mathbf{X})} \chi \Rightarrow \psi$*.*
2. $\mathsf{Prop}(\chi) \subseteq \mathsf{Prop}(\varphi) \cap \mathsf{Prop}(\psi)$*.*
3. $\mathsf{Agt}(\chi) \subseteq \mathsf{Agt}(\varphi) \cap \mathsf{Agt}(\psi)$*.*

We note that not only the condition for propositional variables but also the condition for agents can be satisfied.

Proof. When we set $\Gamma := \varphi$ and $\Delta := \psi$, and take a partition $\langle \Gamma; \varnothing\rangle$, Lemma 5.8 proves Craig Interpolation Theorem. □

Further, decidability of the logics we investigate also follows from the cut-elimination theorem (Theorem 5.3). To show decidability, we introduce a notion of "(1-)reduced sequent".

Definition 5.10. A sequent $\Gamma \Rightarrow \Delta$ is called *reduced* if every formula occurs at most three times in Γ. A sequent $\Gamma \Rightarrow \Delta$ is called *1-reduced* if every formula occurs at most once in Γ.

Definition 5.11. For any sequent $\Gamma \Rightarrow \Delta$, a sequent $\Gamma^* \Rightarrow \Delta$ is a *1-reduced contraction* of $\Gamma \Rightarrow \Delta$ if $\Gamma^* \Rightarrow \Delta$ can be derived from $\Gamma \Rightarrow \Delta$ by applying $(c\Rightarrow)$ to $\Gamma \Rightarrow \Delta$ and is 1-reduced. Clearly, a 1-reduced contraction is determined uniquely.

Proposition 5.12. $\vdash_{\mathsf{G}(\mathbf{X})} \Gamma \Rightarrow \Delta$ *if and only if* $\vdash_{\mathsf{G}(\mathbf{X})} \Gamma^* \Rightarrow \Delta$*.*

Proof. By definition of the 1-reduced contraction, the left-to-right is obvious. The right-to-left is also easily shown by applying $(w\Rightarrow)$ to $\Gamma^* \Rightarrow \Delta$. □

Lemma 5.13. *Suppose that* $\vdash_{\mathsf{G}(\mathbf{X})} \Gamma \Rightarrow \Delta$*. Then, there exists a derivation of* $\Gamma^* \Rightarrow \Delta$ *such that the derivation is cut-free and has only reduced sequents.*

Proof. Thanks to Theorem 5.3, we can take a cut-free derivation of $\Gamma \Rightarrow \Delta$. We name it $\mathcal{D}$. We show by induction on the height of $\mathcal{D}$. We treat only the case where the last rule application of $\mathcal{D}$ is (D). That is, suppose $\mathcal{D}$ is of the form

$$\dfrac{\dfrac{\mathcal{D}'}{\varphi_1, \ldots, \varphi_n \Rightarrow \psi} \quad (\bigcup_{i=1}^{n} G_i \subseteq G)}{D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n \Rightarrow D_G\psi}\ (D).$$

By induction hypothesis, we have a derivation $\mathcal{E}'$ of $(\varphi_1, \ldots, \varphi_n)^* \Rightarrow \psi$ such that $\mathcal{E}'$ is cut-free and has only reduced sequents. Applying the rule (D) to $\mathcal{E}'$, we obtain the desired derivation of $(D_{G_1}\varphi_1, \ldots, D_{G_n}\varphi_n)^* \Rightarrow D_G\psi$. □

Remark 5.14. We admit three occurrences of the same formula in a reduced sequent, because if we only allow at most two occurrences, induction fails in the case of $(\rightarrow\Rightarrow)$ in the proof of this lemma.

Theorem 5.15 (Decidability)**.** *Let* $\mathbf{X}$ *be any of* $\mathbf{IK}$*,* $\mathbf{IKT}$*,* $\mathbf{IKD}$*,* $\mathbf{IK4}$*,* $\mathbf{IK4D}$*, and* $\mathbf{IS4}$*. A logic* $\mathbf{X}$ *is decidable, that is, there is an algorithm checking whether each sequent* $\Gamma \Rightarrow \Delta$ *has a derivation in* $\mathsf{G}(\mathbf{X})$ *or not.*

Proof. We describe a rough sketch of the proof, based on [21, p. 228]. By Proposition 5.12, it suffices to check whether $\Gamma^* \Rightarrow \Delta$ has a derivation. In what follows, by "tree (of $\Sigma \Rightarrow \Theta$)", we mean a tree of sequents (ending with $\Sigma \Rightarrow \Theta$), whose leaves are axioms, or sequents, to which no rule can be applied. Without any restriction, there are infinitely many trees of $\Gamma^* \Rightarrow \Delta$. Therefore, in order to execute a brute-force search, we impose three restrictions on the trees. In general, if a derivation exists, Lemma 5.13 allows us to find a derivation such that (i) it is cut-free and (ii) it has only reduced sequents. By Corollary 5.5, it has subformula property. Therefore, there are finitely many reduced sequents that can be a part of the derivation. Moreover, we can safely assume that (iii) for each path in the derivation from the root sequent to an initial sequent, each sequent in the path occurs exactly once, because, if there are multiple occurrences of the same sequent, we can always eliminate the redundant occurrences by grafting the subderivation for the uppermost occurrence onto the lowermost occurrence. From the above argument, if we impose the conditions (i) to (iii) on the trees of $\Gamma^* \Rightarrow \Delta$, the number of trees becomes finite and we can construct an algorithm enumerating all of them which also checks whether each tree is a derivation or not. If the algorithm does not find any derivation, we can conclude that $\Gamma^* \Rightarrow \Delta$ has no derivation. □

6 Concluding Remark

We conclude this paper with four possible directions for further research. The first direction is to simplify our semantic completeness argument via a similar method given in [30] for classical epistemic logic with distributed knowledge. One of the merits of the method is that the notion of pseudo- (or pre-) model is not necessary. The second direction is to add $\mathbf{S5}$-type axioms to our intuitionistic epistemic logic with distributed knowledge. Since Ono [20] showed that there are at least four distinct $\mathbf{S5}$-type axioms over the intuitionistic modal logic $\mathbf{S4}$, it would be interesting to study the corresponding $\mathbf{S5}$-type axioms in our setting. The third direction is to expand our syntax with the common knowledge operator (cf. [29]). This amounts to investigating the intuitionistic counterpart of [30]. The final direction is to consider dynamic expansions of our syntax. In order to formalize changes of agents' distributed knowledge, for example, we may add public announcement operators [22, 4] or resolution operators [1].

Acknowledgments

The work of both author was partially supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (C) Grant Number JP19K12113 and JSPS Core-to-Core Program (A. Advanced Research Networks). The first author was also partially supported by JSPS KAKENHI Grant-in-Aid for JSPS Fellows Grant Number JP21J10573. The second author was also partially supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (B) Grant Number JP17H02258.

References

1. **Ågotnes, T., Wáng, Y. N. (2017).** Resolving distributed knowledge. Artificial Intelligence, Vol. 252, pp. 1–21.
2. **Ågotnes, T., Wáng, Y. N. (2020).** Group belief. **Dastani, M., Dong, H., van der Torre, L.**, editors, Logic and Argumentation - Third International Conference, CLAR 2020, Hangzhou, China, April 6-9, 2020, Proceedings, volume 12061 of Lecture Notes in Computer Science, Springer, pp. 3–21.
3. **Artëmov, S. N., Protopopescu, T. (2016).** Intuitionistic epistemic logic. The Review of Symbolic Logic, Vol. 9, pp. 266–298.
4. **Balbiani, P., Galmiche, D. (2016).** About intuitionistic public announcement logic. volume 11 of Advances in Modal logic. College Publications, pp. 97–116.
5. **Fagin, R., Halpern, J. Y., Vardi, M. Y. (1996).** What can machines know? on the properties of knowledge in distributed systems. Journal of the ACM, Vol. 39, pp. 328–376.
6. **Fagin, R., Halpern, J. Y., Vardi, M. Y., Moses, Y. (1995).** Reasoning About Knowledge. MIT Press, Cambridge, MA, USA.
7. **Fitch, F. B. (1963).** A logical analysis of some value concepts. The Journal of Symbolic Logic, Vol. 28, No. 2, pp. 135–142.
8. **Gentzen, G. (1935).** Untersuchungen über das logische Schließen. I. Mathematische Zeitschrift, Vol. 39, No. 1, pp. 176–210.
9. **Gentzen, G. (1935).** Untersuchungen über das logische Schließen. II. Mathematische Zeitschrift, Vol. 39, No. 1, pp. 405–431.
10. **Gerbrandy, J. (1999).** Bisimulations on Planet Kripke. Ph.D. thesis, University of Amsterdam.

11. **Giedra, H. (2010).** Cut free sequent calculus for logic $S5_n(ED)$. Lietuvos matematikos rinkinys, Vol. 51, pp. 336–341.
12. **Hakli, R., Negri, S. (2008).** Proof theory for distributed knowledge. **Sadri, F., Satoh, K.**, editors, Computational Logic in Multi-Agent Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 100–116.
13. **Herlihy, M., Shavit, N. (1999).** The topological structure of asynchronous computability. J. ACM, Vol. 46, No. 6, pp. 858–923.
14. **Hirai, Y. (2010).** An intuitionistic epistemic logic for sequential consistency on shared memory. International Conference on Logic for Programming Artificial Intelligence and Reasoning, Springer, pp. 272–289.
15. **Jäger, G., Marti, M. (2016).** A canonical model construction for intuitionistic distributed knowledge. volume 11 of Advances in Modal logic. College Publications, pp. 420–434.
16. **Maehara, S. (1961).** Craig no interpolation theorem (On Craig's interpolation theorem). Sugaku, Vol. 12, No. 4, pp. 235–237.
17. **Marti, M. (2017).** Contributions to Intuitionistic Epistemic Logic. Ph.D. thesis, Universität Bern.
18. **Meyer, J.-J. C., van der Hoek, W. (1995).** Epistemic Logic for AI and Computer Science. Cambridge University Press, New York, NY, USA.
19. **Murai, R., Sano, K. (2020).** Craig interpolation of epistemic logics with distributed knowledge. **Herzig, A., Kontinen, J.**, editors, Foundations of Information and Knowledge Systems - 11th International Symposium, FoIKS 2020, Dortmund, Germany, February 17-21, 2020, Proceedings, volume 12012 of Lecture Notes in Computer Science, Springer, pp. 211–221.
20. **Ono, H. (1977).** On some intuitionistic modal logics. Publications of the Research Institute for Mathematical Sciences, Vol. 13, No. 3, pp. 687–722.
21. **Ono, H. (1998).** Proof-theoretic methods in nonclassical logic –an introduction. **Takahashi, M., Okada, M., Dezani-Ciancaglini, M.**, editors, Theories of Types and Proofs, volume 2 of MSJ Memoirs, The Mathematical Society of Japan, Tokyo, Japan, pp. 207–254.
22. **Plaza, J. (2007).** Logics of public communications. Synthese, Vol. 158, No. 2, pp. 165–179.
23. **Pliuskevicius, R., Pliuskeviciene, A. (2008).** Termination of derivations in a fragment of transitive distributed knowledge logic. Informatica, Lith. Acad. Sci., Vol. 19, pp. 597–616.
24. **Proietti, C. (2012).** Intuitionistic epistemic logic, Kripke models and Fitch's paradox. Journal of Philosophical Logic, Vol. 41, No. 5, pp. 877–900.
25. **Roelofsen, F. (2007).** Distributed knowledge. Journal of Applied Non-Classical Logics, Vol. 17, No. 2, pp. 255–273.
26. **Saks, M., Zaharoglou, F. (2000).** Wait-free k-set agreement is impossible: The topology of public knowledge. SIAM Journal on Computing, Vol. 29, No. 5, pp. 1449–1483.
27. **Suzuki, N.-Y. (2013).** Semantics for intuitionistic epistemic logics of shallow depths for game theory. Economic Theory, Vol. 53, No. 1, pp. 85–110.
28. **van der Hoek, W., van Linder, B., Meyer, J.-J. (1999).** Group knowledge is not always distributed (neither is it always implicit). Mathematical Social Sciences, Vol. 38, No. 2, pp. 215–240.
29. **van Ditmarsch, H., van der Hoek, W., Kooi, B. P. (2007).** Dynamic Epistemic Logic, volume 337 of Synthese Library. Springer Science & Business Media.
30. **Wáng, Y. N., Ågotnes, T. (2020).** Simpler completeness proofs for modal logics with intersection. **Martins, M. A., Sedlár, I.**, editors, Dynamic Logic. New Trends and Applications, Springer International Publishing, Cham, pp. 259–276.
31. **Williamson, T. (1992).** On intuitionistic modal epistemic logic. Journal of Philosophical Logic, Vol. 21, pp. 63–89.

Article received on 20/10/2020; accepted on 24/02/2021.
Corresponding author is Ryo Murai.

ISSN 2007-9737

Measuring the Quality of Low-Resourced Statistical Parametric Speech Synthesis Trained with Noise-Degraded Data Supported by the University of Costa Rica

Marvin Coto-Jiménez

University of Costa Rica,
Costa Rica

marvin.coto@ucr.ac.cr

Abstract. After the successful implementation of speech synthesis in several languages, the study of robustness became an important topic so as to increase the possibility of building voices from non-standard sources, e.g. historical recordings, children's speech, and data freely available on the Internet. In this work, a measure of the influence of noise in the source speech of the statistical parametric speech synthesis system based on HMM is performed, for a case of a low-resourced database. For this purpose, three types of additive noise were considered at five signal-to-noise ratio levels to affect the source speech data. Using objective measures to assess the perceptual quality of the results and the propagation of the noise through all the processes of building speech synthesis, the results show a severe drop in the quality of artificial speech, even for the cases of lower levels of noise. Such degradation seems to be independent of the noise type, and is at lower proportion to the noise level. This results are of importance for any practical implementation of speech synthesis from degraded data in similar conditions, and shows that applying denoising processes became mandatory in order to keep the possibility of building intelligible voices.

Keywords. Noise, robustness, speech synthesis.

1 Introduction

The purpose of speech synthesis can be established as the production of artificial speech from a given text input using computers. The resulting speech should be perceived with intelligibility and naturalness, in order to apply the results in the desired application. This process of speech synthesis (also referred to as text-to-speech) has a long history, from early mechanic systems to our days, where complex techniques and the release of dedicated software have extended the speech synthesis possibilities to many languages and applications.

The evolution of modern techniques can be traced back to the early 1970s [1], where the waveform generation was made using low-dimensional information, such as formants. And it has evolved to perform direct manipulations of waveforms (e.g. concatenative and unit selection approaches) or high dimensional parameters and deep learning-based models.

The statistical models of speech synthesis, mainly based on Hidden Markov Models (HMM), were popularized among researchers of the field after the first publications of the technique [2, 3], particularly after the release of the HTS software [4]. HMMs were previously successfully applied to speech recognition, and many of the ideas and parameters applied for that task were translated to the speech synthesis field.

With the HTS software, many papers were published on the implementation of statistical parametric speech synthesis in several languages around the world. The case of Spanish was also reported by a reduced number of researchers [5, 6, 7].

The advantages of statistical parametric speech synthesis based on HMM were reported in terms of its flexibility and capacity for producing intelligible

voices with low-training data [8]. The main disadvantages were the buzzy, muffled sound often reported.

With the increased performance and success of deep learning in several fields during the last decade, speech synthesis also benefits from the possibilities of the complex modeling and effective training algorithms of deep neural networks. The first ideas on the implementation of deep learning in speech synthesis were published in [9].

In previous years, many proposals have been made to apply different types of neural networks, such as Restrictive Boltzmann Machines, Deep Belief Networks, Bidirectional Long Short-term Memory Neural Networks, and Convolutional Neural Networks [10]. In some recent reports, the combination of both statistical parametric modeling combined with deep learning was also published [11, 12].

Typically, the deep learning-based approaches report a higher quality of results but require a large amount of training data. There are many situations where the availability of such resources is not possible to achieve. For example, in building speech from historical recordings, children's speech and low resourced languages [13, 14].

For these cases, HMM-based statistical parametric speech synthesis remains the main possibility to produce intelligible artificial voices. In many of such cases, the quality of the recordings was also a shortcoming for the quality of the results.

The usual framework in the building of synthetic voices was considered in the vast majority of cases: the recording of datasets in highly-controlled environments, which has typically done in professional studios with high-quality equipment. According to to [15], given the advances in speech synthesis techniques, the research community can consider building quality voices from data collected in less controlled environments. These new conditions represent several challenges for the process, for example, non-consistent recording conditions, unbalanced phonetic material, and noisy data. It is still not clear how robust speech systems are under such unfavorable conditions [16].

The problem of producing artificial speech has been addressed by some authors with particular interests in techniques that take advantage of a large corpus of clean data, such as speaker-adaptation in HMM-based speech synthesis. Using such corpus new voices can be built by incorporating information from the corpus in the smaller datasets.

For example, in [17], the authors proved that naturalness is not significantly affected by the presence of noise in the smaller dataset. The unfavorable conditions can be presented in found data, i.e. data freely available on the web. Such data has significant variation in terms of speaking style and channel characteristics [18].

In this paper, an experimental study on the influence of noisy recordings in the results of statistical parametric speech synthesis is performed, for the case of a small database in Spanish. The purpose of the study is to numerically report and compare the influence of several types and levels of noise in the speech data required to produce artificial speech.

The influence of the noise provide information to anticipate the quality of artificial speech that can be produced from recordings with unfavorable conditions. Such information is relevant for the evaluation of low-quality sources of speech resources in building speech synthesis.

The rest of this paper is organized as follows: Section 2 presents the theoretical background of speech synthesis and the effects of noise. Section 3 presents the experimental setup of the proposal. Section 4 presents the results. Finally, the Conclusions are presented in Section 5.

2 Statistical Parametric Speech Synthesis

The Statistical Parametric Speech Synthesis based on HMMs models the speech production process using the source-filter theory of voice production [1]. This model comprises the voicing information using fundamental frequency (or the logarithm of this measure) and the spectral envelope, commonly represented by mel-frequency cepstral coefficients (MFCC). The

speech waveforms are reconstructed from sequences of such parameters, and additional information about dynamic features (e.g. rate of change in form of delta and delta delta features [19]).

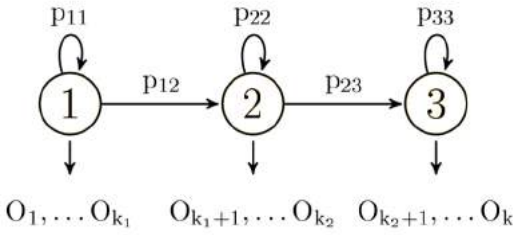


Fig. 1. Left-to-right Hidden Markov Model with three states

First, the HMMs are trained, using a similar approach to that utilized in speech recognition: adjusting the parameters of the HMM model (Figure 1) using information extracted from a speech database. Each HMM can be expressed as:

$$\lambda = (\pi, \mathbf{a}, \mathbf{b}), \tag{1}$$

where π is the probability of initial-state, $\mathbf{a}$ and $\mathbf{b}$ the state-transition and output probability distributions, assumed as multivariate Gaussian distributions (with a mixture of continuous and 0-dimensional distributions).

In statistical parametric speech synthesis based on HMM, the set of models depends not only on the number of phonemes of the particular language, but on the context-dependency of the phonemes (phonetic and prosody contexts) as well. For this reason, a large number of models are trained to represent the temporal, spectral and pitch characteristics of every sound and its context. For example, a model for the $< a >$ phoneme at the beginning of a phrase, followed by consonant, and a model for the $< a >$ phoneme at the beginning of a phrase, followed by a vowel, etc.

The training of each HMM can be expressed as:

$$\lambda_{max} = \arg\max_{\lambda} p(\mathbf{O}|\lambda, W), \tag{2}$$

where $\mathbf{O}$ is the set of speech parameters and W the phoneme labels. A detailed description of the HMM and the procedures involved in the speech synthesis can be found in [1, 20].

For this work, it is of particular importance to state that the quality of the speech synthesis relies on the proper adjustment of the parameters of λ_{max} in Equation 2. And this adjustment depends on the quality of the features $\mathbf{O}$ extracted from the dataset, and its consistency according to the phoneme labels (linguistic specification) W.

Several factors can affect the outcomes of the process: The amount of information in the database (few information implies less $\mathbf{O}$ to estimate the parameters of the HMMs) and the quality of this information. If the information is corrupted by noise, or the recordings have large variations among phonemes (typically, this can occur in very expressive or emotional speech), the ability of the HMMs to reproduce the parameters of the speech for a natural sounding voice with high intelligibility may be affected. The nature of such noise and its level can also be a relevant factor for the results. In this work, an experimental validation of such assumptions is proposed and measured.

3 Experimental Setup

3.1 Database

For this work, we selected the set of words and sentences of [21], developed at the Center for Language and Speech Technologies and Applications of the Polytechnic University of Catalonia. The 184 utterances were recorded by a professional native Spanish speaker actor in a professional studio, where the recording conditions were controlled completely. The database includes affirmative and interrogative sentences, fifteen paragraphs, digits and isolated words.

3.2 Experiments

To determine how noise affects the building of synthetic voices with such small database, several voices were produced using the HTS system, each one after affecting the speech source with noise. The complete database was degraded with additive noise of three types: two artificial-generated noise (White, Noise) and

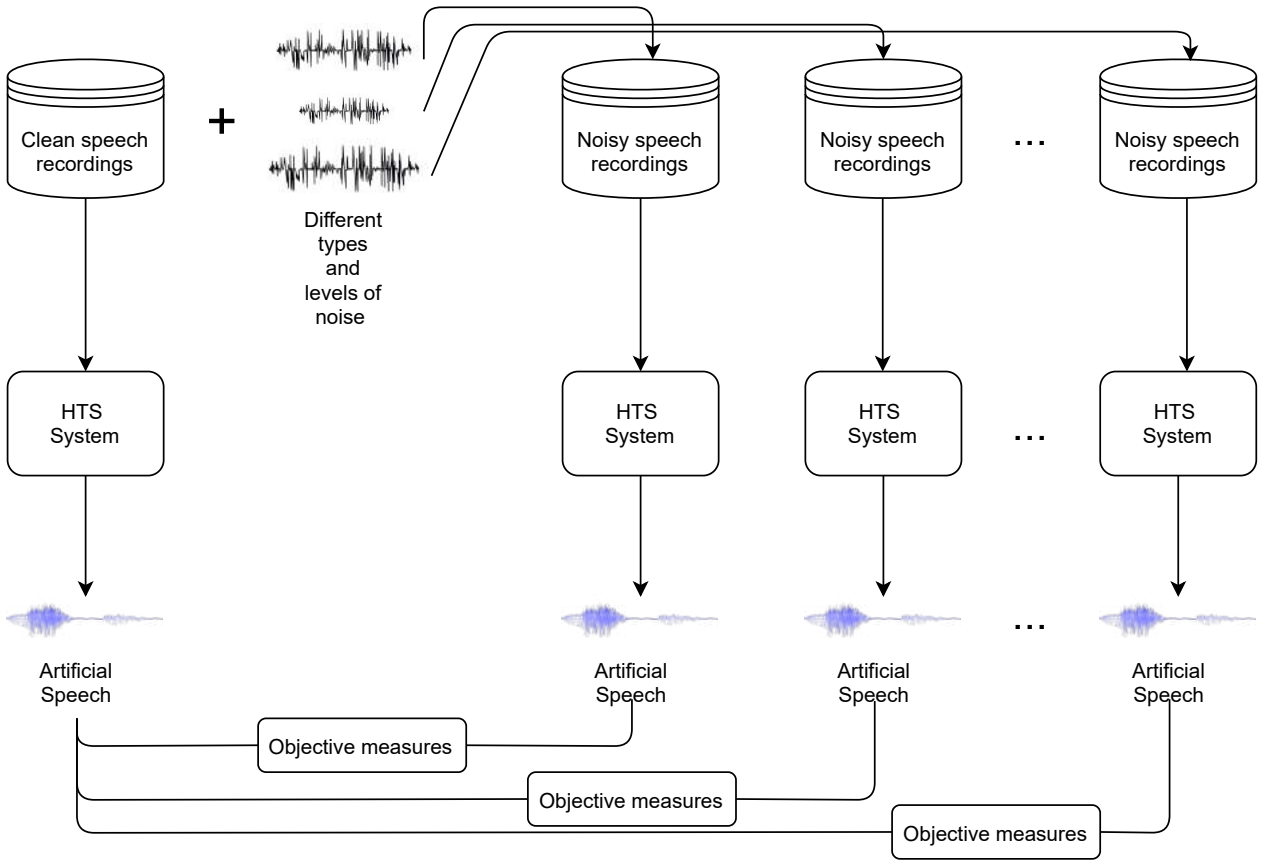


Fig. 2. Diagram of the experimental procedure

one natural noise (Babble). Five levels of Signal-to-noise (SNR) ratio were considered, to cover a range of conditions and comparatively assess the effect on the results.

The whole set of voices to compare can be listed as:

— HTS Clean: The produced with the clean database, without any noise added.

— White Noise added at five SNR levels: SNR 5, SNR 7.5, SNR 10, SNR 12.5, SNR 15.

— Pink Noise added at five SNR levels: SNR 5, SNR 7.5, SNR 10, SNR 12.5, SNR 15.

— Babble Noise added at five SNR levels: SNR 5, SNR 7.5, SNR 10, SNR 12.5, SNR 15.

The evaluation metrics proposed in the following section were used to compare the level of degradation on the artificial voice in comparison with the base system (HTS clean). A diagram of the complete process is presented in Figure 2.

3.3 Evaluation

To determine the quality of each case of synthetic voice, two objective measures were applyed. These measures have been reported in speech synthesis reports as reliable in measuring the quality of synthesized voices:

— Segmental SNR (SegSNR): This measure calculate the average of SNR at frame level,

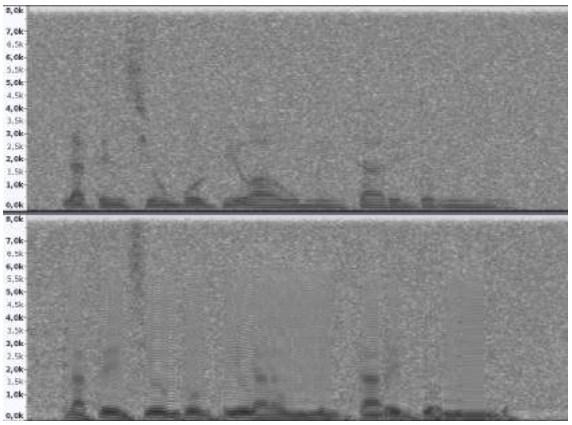


Fig. 3. Spectrograms of an utterance with White noise at SNR5 (above) and the same utterance synthesized from a database degraded with the same type and level of noise (below)

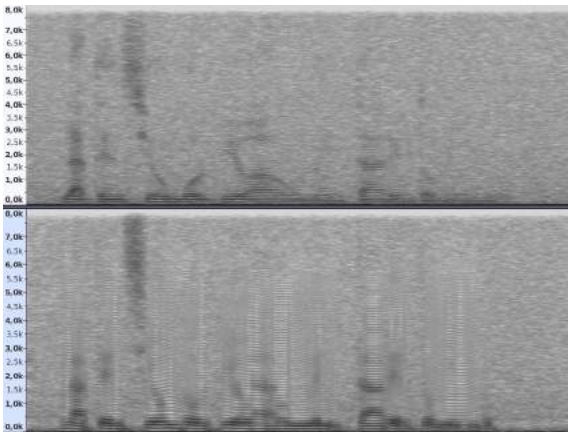


Fig. 4. Spectrograms of an utterance with Pink noise at SNR10 (above) and the same utterance synthesized from a database degraded with the same type and level of noise (below)

according to the equation:

$$\text{SegSNR} = \frac{10}{N}\sum_{i=1}^{N}\log\left[\frac{\sum_{j=0}^{L-1} s^2(i,j)}{\sum_{j=0}^{L-1}\left(s(i,j)-x(i,j)\right)^2}\right], \tag{3}$$

where $x(i)$ is the original sample and s_i the i^{th} synthetic speech sample. N is the total number of samples of the utterance and L is the frame length.

— PESQ: This is a measure intended to predict the subjective perception of speech, in ITU-T recommendation P.862.ITU. The results are reported in the interval $[0.5, 4.5]$. A PESQ value of 4.5 means an exact reconstruction of the speech. PESQ is computed following the equation:

$$\text{PESQ} = a_0 + a_1 D_{ind} + a_2 A_{ind}. \tag{4}$$

The coefficients a_k are chosen to optimize PESQ measure in signal distortions and overall quality.

Additionally, we propose the visualization of spectrograms as a mean to represent the noise and its effect on the spectrum of the speech signals.

4 Results

This section presents the evaluation metrics on the different experiments and its analysis in terms of how the presence of noise affects the building of synthetic voices. For example, in the spectrograms of Figure 3, the silence segments at the beginning and the end of the noisy speech (with SNR 5), and the synthesized version of the same utterance preserves similar patterns of the noise. On the other hand, in the speech segments, the spectrogram presents noticeably blurred bands of frequencies.

A similar observation can be made for the case of Pink noise at SNR 10, as presented in Figure 4. The particular pattern in the form of bands of frequencies can be explained for the process of adjusting the trajectories of parameters in the HMMs. The noisy information became part of the information adjusted in the models, and in the process of generating parameters, the characteristics of flat trajectories also affected the noise.

Unfortunately, such characteristic during the speech segments in the spectrograms represent considerable decrease in the objective measures of the synthesized voice. For example, Figure 5 shows how the noisy condition of the data severely affects the perceptual quality of the

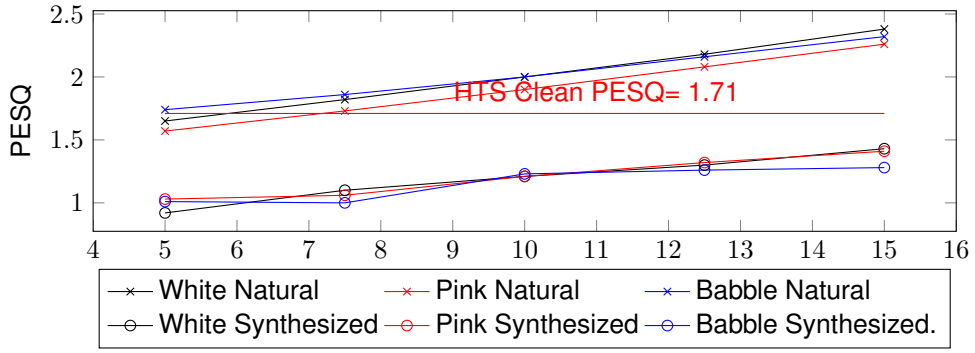


Fig. 5. PESQ results for the noise-degraded speech and the artificial version produced from the same speech

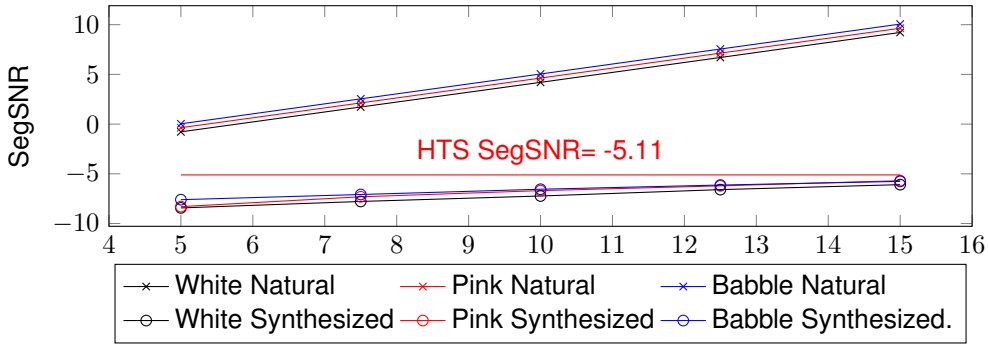


Fig. 6. SegSNR results for the noise-degraded speech and the artificial version produced from the same speech

synthesized speech at all SNR levels. At SNR5 of White Noise, the resulting synthesized speech is closed to the lowest value of PESQ. All artificial voices produced under noisy conditions have considerably lower PESQ values than the base system: the HTS voice.

There are no significant differences between the three types of noise analyzed in this work. The Babble noise seems to affect the results more than the artificial voices, which is expected due to the speech nature of such noise (consisting of a crowd talking in the background).

Considering SNR levels below SNR 5 is a common practice in the study of robust speech recognition. But with these results, it seems that below this level, the synthesized speech for a low resource database cannot be considered for any practical application.

The results of the measure SegSNR are presented in Figure 6. Like the previous measure, there is a significant drop in the quality of synthetic voices at all SNR levels, and very similar among the noise types. All the cases present values below the base system (HTS Clean voice, with SegSNR=-5.11) as expected, but there is a decrease in the slope of the lines in the synthesized speech that can be considered an unexpected result of this study. Such behavior in the SegSNR trends at all SNR levels can be explained by the averaging process performed during the training of the HMMs.

All the results presented have similar trends in the dropping of the quality of synthetic voices in the presence of noise; thereby, preserving the slope of the degraded speech for the case of PESQ. It is important to remark that the results were obtained from a Spanish speech database that can be considered low-resourced. The robustness of the HTS system under such conditions can be considered very low in contrast to the experiences

reported in the references that took advantage of adaption systems or the complement of clean speech from other speakers during the process of generating the artificial speech.

5 Conclusions

In this work, an experimental study on the quality of synthetic speech built from a Spanish noisy database was performed. The amount of data available for the experiments can be considered low-resourced in contrast to larger speech databases available in other languages.

The obtained results show how the presence of noise in the recordings severely affects the synthetic voices produced, regardless of the type of noise and the SNR. In particular, the perceptual quality measured using PESQ shows how the resulting voices have lower quality than the voices produced from clean speech. The type of noise seems to make no difference in the quality of the synthetic speech.

The results are relevant to the building of synthetic voices where data cannot be collected in controlled environments, from historic recordings, data freely available on the Internet, or recordings performed during videoconferencing.

In addition, the results help to establish the importance of building a clean larger speech corpus for endangered languages, children's speech, and many other potential applications of speech synthesis in new languages or languages where such resources have not been produced.

For future work, several relevant questions can be addressed for experimental validation, in terms of the robustness of speech synthesis systems under partially noise-corrupted data, and a broader range of noise types and levels. Applying denoising algorithms before the building of the voices is an important opportunity to preserve the possibility of generating synthetic voices from noise-degraded data.

References

1. **Tokuda, K., Nankaku, Y., Toda, T., Zen, H., Yamagishi, J., Oura, K. (2013).** Speech synthesis based on hidden Markov models. Proceedings of the IEEE, pp. 1234–1252.
2. **Masuko, T., Tokuda, K., Kobayashi, T., Imai, S. (1996).** Speech synthesis using HMMs with dynamic features. IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, Vol. 1.
3. **Tokuda, K., Kobayashi, T., Imai, S. (1995).** Speech parameter generation from HMM using dynamic features.International Conference on Acoustics, Speech, and Signal Processing. Vol. 1.
4. **Zen, H., Nose, T., Yamagishi, J. (2007).** The HMM-based speech synthesis system (HTS) version 2.0. SSW.
5. **Gonzalvo, X., Sanz, I.I., Socoró-Carrié, J.C., Alías F. (2007).** HMM-based Spanish speech synthesis using CBR as F0 estimator. ITRW on NOLISP, pp. 788–793.
6. **Gonzalvo, X., Taylor, P., Monzo, C., Sanz, I.I. (2009).** High quality emotional HMM-based synthesis in Spanish. International Conference on Nonlinear Speech Processing, Springer. DOI:10.1007/978-3-642-11509-7_4.
7. **Franco, C.A., Herrera, A., Escalante B. (2017).** Speech synthesis in Mexican Spanish using voice parameterization. IIISCI, 15(4), pp. 72–75.
8. **Ekpenyong, M., Urua, E.A., Watts, O., King, S., Yamagishi, J. (2014).** Statistical parametric speech synthesis for Ibibio. Speech Communication, Vol. 56, pp. 243–251. DOI: 10.1016/j.specom.2013.02.003.
9. **Ze, H., Senior, A., Schuster, M. (2013).** Statistical parametric speech synthesis using deep neural networks. IEEE International Conference on Acoustics, Speech and Signal Processing. DOI: 10.1109/ICASSP.2013.6639215.

10. **Ning, Y., He, S., Wu, Z., Xing, Ch. (2019).** A review of deep learning based speech synthesis. Applied Sciences, Vol. 9, No. 19, pp. 4050. DOI: 10.3390/app9194050.

11. **Hu, Y.J., Ling, Z.H. (2016).** DBN-based spectral feature representation for statistical parametric speech synthesis. IEEE Signal Processing Letters, Vol. 23, No. 3, pp. 321–325. DOI: 10.1109/LSP.2016.2516032.

12. **Hu, Y.J., Ling, Z.H. (2018).** Extracting spectral features using deep autoencoders with binary distributed hidden units for statistical parametric speech synthesis. IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 26, No. 4, pp. 713–724. DOI: 10.1109/TASLP.2018.2791804.

13. **Suraj-Pandurang, P., Laxman-Lahudkar, S. (2019).** Hidden-Markov-model based statistical parametric speech synthesis for Marathi with optimal number of hidden states. International Journal of Speech Technology, Vol. 22, No. 1, pp. 93–98.

14. **Sefara, T.J., Mokgonyane, T.B., Manamela, M.J., Modipa, T.I. (2019).** HMM-based speech synthesis system incorporated with language identification for low-resourced languages. International Conference on Advances in Big Data, Computing and Data Communication Systems (ICABCD). DOI: 10.1109/ICABCD.2019.8851055.

15. **Junichi, Y., Ling, Z., King, S. (2008).** Robustness of HMM-based speech synthesis.

16. **Valentini-Botinhao, C., Wang, X., Takaki, S., Yamagishi, J. (2016).** Speech Enhancement for a Noise-Robust Text-to-Speech Synthesis System Using Deep Recurrent Neural Networks. Interspeech.

17. **Karhila, R., Remes, U., Kurimo, M. (2013).** Noise in HMM-based speech synthesis adaptation: Analysis, evaluation methods and experiments. IEEE Journal of Selected Topics in Signal Processing, Vol. 8, No. 2, pp. 285–295.

18. **Baljekar, P. (2018).** Speech synthesis from found data. PhD thesis, Carnegie Mellon University.

19. **Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., Kitamura, T. (2000).** Speech parameter generation algorithms for HMM-based speech synthesis. IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 3. DOI: 10.1109/ICASSP.2000.861820.

20. **Toda, T., Tokuda, K. (2007).** A speech parameter generation algorithm considering global variance for HMM-based speech synthesis. IEICE Transaction on Information and Systems, Vol. 90, No. 5, pp. 816–824. DOI: 10.1093/ietisy/e90-d.5.816.

21. **Maegaard, B., Choukri, K., Calzolari, N., Odijk, J. (2005).** Elra – European language resources association - background, recent developments and future perspectives. Language Resources and Evaluation, Vol. 39, No. 1, pp. 9–23. DOI: 10.1007/s10579-005-2692-5.

Article received on 09/10/2020; accepted on 16/02/2021. Corresponding author is Marvin Coto-Jiménez.

ISSN 2007-9737

Searching and Updating Research Materials for Renewing Curricula of Academic Disciplines Using Example of Logistics

Pavel P. Makagonov[1], Sergey A. Maruev[1], Varvara D. Makagonova[2], Liliana Chanona-Hernandez[3]

[1] Russian Presidential Academy of National Economy and Public Administration (RANEPA), Russia

[2] Independent researcher, Moscow, Russia

[3] Instituto Politécnico Nacional, ESIMEZ, Mexico

mpp2003@inbox.ru, maruev@ranepa.ru, var-mak@mail.ru, lchanona@gmail.com

Abstract. The research covers the preparation and use of academic publications' materials for the analysis of development dynamics of a narrow subject area's ontology by searching for new concepts (keywords). This technology should be applied annually to monitor the dynamics of the appearance of new keywords that are not included in the course materials of a higher education institution on a relevant academic discipline. In this research, logistics is chosen as a narrow subject area by the example of which the use of the proposed methodology for the analysis of development dynamics of an academic discipline's ontology is demonstrated. A more narrowly focused goal associated with this methodology is to search for new concepts in logistics when formulating the theme of upcoming research conducted by a young specialist. Identification of rare keywords that are not presented in course materials of higher education institutions on a relevant subject is a poorly formalized research problem the solution to which does not guarantee success. It is feasible to generate annual life cycle curves for the identified concepts even if they do not change gradually over time because their behaviors can be important for planning future research.

Keywords. Development dynamics of ontology; narrow subject area; curricula of academic disciplines.

1 Introduction

The purpose of the research is to prepare and use academic publications' materials for the analysis of development dynamics of a narrow subject area's ontology by searching for new concepts (keywords) to update course materials of a higher education institution on a relevant academic discipline. In this research, logistics is chosen as a narrow subject area by the example of which the use of the proposed methodology for the analysis of development dynamics of an academic discipline's ontology is demonstrated.

Instead of ontology, the prototype will account for the materials provided in the curriculum of a specific academic discipline (CAD).

This document obligatorily contains concepts (keywords of the studied discipline) and connections between them (as a rule, these are "whole-part" connections). CAD is the most important document guiding students through the subject of the studied discipline and its updating is of great significance for achieving the necessary learning goals.

As an academic discipline, logistics is the basis (core) of several specialties of Bachelor's and Master's Degree programs in higher education institutions. A more narrowly focused goal is to search for new concepts in logistics for upcoming research conducted by a young specialist. However, the initial variant of the purpose of the research will be studied here. Proposed stages of the research:

1. Estimation of the life cycle of logistics as an academic discipline based on the dynamics of academic publications. The dynamics will be analyzed through multiple search queries in the Scientific Electronic Library *e-library* and a set of annotations with keywords and references.
2. Formation and analysis of a "words-texts" matrix to identify new logistics concepts and keywords that are not presented in course materials and estimate the upward or downward dynamics of researchers' interest in the identified concepts.
3. Automated analysis of the "words-texts" matrix to cluster words and texts and search for prototypes of new keywords that can be included in curricula of academic disciplines with the help of the following programs:
   - FrequencyOfWords - the program is designed to generate a "words-texts" matrix from a set of texts. This matrix is suitable for identifying connections in texts' dictionaries and their filtration.
   - CorrTable - the program analyzes the "words-texts" matrix generated on the basis of a set of texts belonging to a narrow subject area. Pairs of words that can be potentially included in the list of bigrams and key concepts belonging to the narrow subject area are singled out from the matrix using automated exhaustive search. Such words are characterized by a high value of the ratio of the standard deviation to the texts' average compared to the original sample. Moreover, their use in a pair has a high correlation in two texts.
   - BigramsInText - the program contains a list of pairs of words and text(s) where they were used. The analyzed text is divided into "phrases" without punctuation marks (except for hyphens and quotation marks). This allows finding all phrases where pairs of words form bigrams suitable for identifying two-word key concepts that belong to a certain subject area in an explicative context.
   - HapaxToBigrams - the program contains a list of words used only in one text as well as these texts. Its results coincide with the results obtained using the program BigramsInText.

This set of programs is called ONT-01 because it is designed to collect concepts based on which a prototype of the topic's ontology reflecting the set (collection, corpus) of analyzed texts can be constructed. Logistics is of interest as an academic discipline. This is why the ontological basics of this category that are accepted in education courses of higher education institutions are of great importance.

General principles of ontology-building are not included in the research. It would be sufficient to consider recent education courses as a representative basis for the hierarchical network of concepts used in logistics courses. These concepts can be found in courses' tables of contents and textbooks' indices. The completeness of the concepts' system is associated with the completeness and specificity of courses that depend on the specialization of a higher education institution for which a particular textbook or study guide was written.

This is why the difference in courses' ontologies can be explained by the specialization of a particular course. The problem that this research aims to solve is to estimate the degree of novelty of course materials against the background of recent academic publications. This will help to avoid or at least mitigate the deterioration of course materials. Consequently, a university graduate will have sound theoretical training to work with research and production subjects related to what he or she has studied in the last two years.

Comparison of existing courses allows evaluating the scope of a studied discipline and the narrowness of a course's specialization. However, it may be that a course is divided into several disciplines. To evaluate the completeness of a

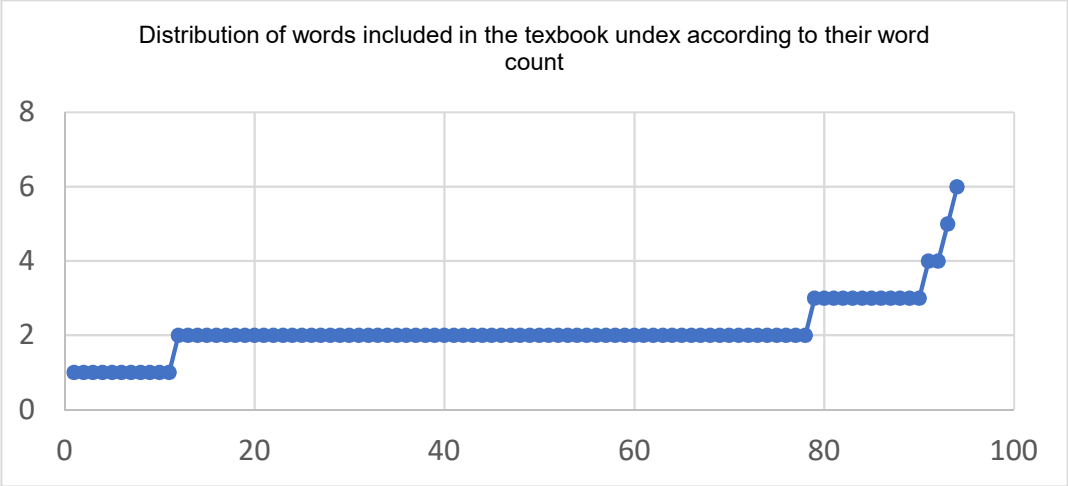


Fig. 1. Illustration of information provided in the text

course from the standpoint of its novelty, it is necessary to identify the appearance of new concepts and notions in the academic literature that relate to the development of the subject covered in the course.

Existing methods of building initial ontologies of education courses as well as methods suitable for these purposes [15-20] cannot solve the aforementioned problem. The proposed methodology does not imply the work of ontology-building specialists and focuses only on identifying new concepts and keywords that can be used to evaluate the feasibility of including them in a constantly updated curriculum of a particular academic discipline. In this case, it is necessary to identify terms that can potentially turn into keywords at a relatively high abstraction level but have not yet become a part of the concepts' network studied in the course.

In contrast to searching for basic keywords belonging to the subject area of logistics, in this case, the search is aimed only at new, recently appeared keywords that have not been included in curricula of academic disciplines related to a particular subject area for this or another reason. These new keywords should possess the following characteristics to distinguish them from those that are already accepted in education courses:

1. They should not be presented in dictionaries (lists), compiled on the basis of textbooks or study guides on logistics for higher education institutions. Thus, keywords mentioned in textbooks should already be included in ontologies of higher education institutions' courses so they need to be filtered out from the texts of academic publications on logistics.
2. They should be rarely used. In other words, they should be presented in a limited number of recent publications.
3. New keywords should be found neither at the top of the hierarchical structure of logistics ontology nor at the highest abstraction level of this subject. This level of abstraction is typical of concepts expressed by one or a few words that are often characterized by a supersystemic nature.
4. The most preferable approach here is to search for two-word key concepts, that is, bigrams.

Let us clarify the latter characteristic by the following example.

The index of the textbook "Logistics" by V. N. Zhigalova contains one hundred terms. 11 terms consist of one word, 67 - of two words, 12 - of three words, and 4 - of four and more words. The word count of four other terms is questionable. These are freight traffic, make-or-buy problem (MOB), Kanban system (in Japanese, Kanban means a "signboard" or a "billboard"), and Optimized Production Technology system (OPT). However, two-word terms constitute about 70% of the

sample. Terms consisting of one word (i.e., agent, broker, dealer, distributor, supplies, reputation, incoterms, commissioner, tender, terminal, and transport) are characterized by a supersystemic nature with regard to logistics. Terms consisting of more than one word are at a lower abstraction level in the logistics ontology (e.g., deterministic calculation method, link in a logistics system, total cost concept, supply chain in logistics, transportation invoice, direct logistics channels, min-max system, stochastic calculation method, consignment note, customer service level, expert calculation method, echelon logistics channels, fixed order quantity system, distribution level of a logistics flow, fixed interval reorder system, and fixed replenishment frequency system).

Based on this example, it is feasible to search only for two-word key concepts. The necessary reference materials include:

- A conceptual dictionary of recent education courses on logistics (published up to 2018).
- A conceptual dictionary of academic publications on logistics over the last 5 years.

It is necessary to identify the rarest and most recent concepts from the latter dictionary that are not included in the former one.

Identification of rare terms and estimation of their potential to become keywords can be carried out by reading a plethora of recent publications and manual selection performed by experts in a particular subject area.

This research, however, considers a situation when this problem needs to be solved by a young logistics specialist who is familiar with the introductory course provided in his or her higher education institution.

A program-algorithm complex used to solve the aforementioned problem will not be examined here in detail. However, the key points of its rationale for the analysis of logistics courses as well as obtained results will be provided below.

The proposed methodology was presented by P.P. Makagonov during the international conference session on June 20, 2020 [1].

2 Model of the Logistics Life Cycle

As was previously demonstrated by Ruiz Figueroa and Makagonov [3], the life cycle of a narrow subject area should have its own empirical model that is described by some parameter *S*, the model of which constitutes a logistic curve *S(t)* denoting the cumulative number of publications for the time *t*:

$$S(t) = \frac{1}{1 + \exp\left(-k\left(t - T_{ip}\right)\right)}. \quad (1)$$

Here, T_{ip} denotes the abscissa of the inflection point, *k* – the tangent of the slope of the curve in the inflection point.

A rising branch of a cubic parabola between the left minimum and the right maximum is more preferable as a model than a logistic curve. This model is easier to generate by calculating the polynomial coefficients with the help of methods provided in Excel for searching for the trend based on a set of points of the initial distribution.

Academic publications best reflect the success of a scientific paradigm's development. Moreover, academic publications on a narrow subject area are best suited for content analysis due to better self-organization, the higher linguistic discipline of researchers, and the stricter selection of articles before publication.

Development dynamics of a scientific paradigm can be represented by the cumulative frequency of academic publications and academic conferences' reports on topics related to the core of the paradigm. A representative sample of a scientific paradigm's manifestations from a temporal perspective usually resembles a logistic curve. In the case with a "young" and immature paradigm, the sample is well approximated by a J-shaped curve that can potentially develop into an S-shaped curve. This J-shaped curve normally has a starting point that corresponds to the time of appearance of the first academic publication marking the beginning of a new scientific paradigm's life cycle.

If the sample is formed on the basis of data that do not contain the initial stage of the paradigm's development, then the curve may have the form of an upper branch of a logistic function, the outgo to the horizontal asymptote of which is poorly manifested.

Table 1. The total number of publications: 15,773

Year	Number of publications	Cumulative number of publications
2016	3,461	3,461
2017	3,572	7,033
2018	3,657	10,690
2019	3,838	14,528
2020	1,245	15,773

Let us consider the simplest appropriate mathematical representations of the aforementioned heuristic J-shaped and S-shaped models. In a simplified manner, complete J-shaped and S-shaped models can be represented by a segment of the curve of a cubic parabola, between the left minimum and the right maximum of the following function:

$$S(t) = a_3 \times t^3 + a_2 \times t^2 - a_1 \times t + a_0. \qquad (2)$$

Logistics as a science has a long development period. This is why the number of recent publications constitutes from 3500 to 3800 texts and is characterized by a positive dynamic.

A search query in e-library by subject “Logistics” for the period from 2016 to 2020 gave the results in Table 1.

The query was made in the middle of 2020 so that year was not taken into account while calculating the coefficients of the approximating cubic parabola for the cumulative number of publications. The approximation in the form of a cubic parabola is of the following form:

$$y = 16x3 - 773.5x2 + 16026x – 120467, \quad R^2 = 1. \qquad (3)$$

Here, *y* denotes the cumulative number of publications starting from 2016, *x* - time in years (excluding 2000), *R* - the coefficient of determination.

A positive value of the coefficient given the third degree means that despite a significant number of texts published annually, the life cycle of logistics as a subject area in the scientific and technical literature is still at an early developmental stage. If the coefficient is positive under a higher degree (a3=16), the curve of the forecasted number of publications goes into infinity. This means that the subject of logistics will not become obsolete in the next 20-30 years and specialists in this field will be in demand for a long time. However, the subject may significantly change in the details over time.

The number of publications within the considered period is so large that the number of analyzed texts needs to be limited to evaluate all scientific interests. Let us take 50 publications per year under the condition that the search query implies a decrease in texts’ relevance so that the first 50 publications are the most relevant. Such a limited analysis will not allow identifying numerous underrepresented scientific endeavors. However, the main new topics will be included in the analysis.

It should be clarified that the research does not aim to build a generalized ontology of the aforementioned subject. Notions and concepts are presented in sufficient detail in the review “Logistic paradigms: Functional, resource, innovative” (link: https://mybiblioteka.su/tom3/3-6190.html).
However, it is important to compare and contrast notions in the curriculum of an academic discipline with notions appearing in recent academic publications for the former to remain up-to-date and meet the standards of contemporary science.

It should be also mentioned that a simple comparison of notions, concepts, and keywords from a particular article with the academic discipline’s curriculum can be very useful in evaluating the limitedness of the curriculum compared to the contents of logistic paradigms and concepts from a temporal perspective. The purpose of this research is to help experts in logistics to realize the need to adjust or supplement the contents of the academic discipline’s curriculum promptly. This adjustment or supplement can relate only to that part of the

subject's ontology that is associated with topics of a high abstraction level thus slightly expanding the subject according to the latest industrial trends. Studying the dynamics of the recent linguistic scientific dictionary in comparison with the dictionaries of logistics courses can contribute to the achievement of the aforementioned goals.

3 Results of Application of the Program Complex to a Set of Texts on Logistics

Let us consider the results of the application of the program complex by the example of a comparative analysis of the logistics subject area accompanied by the corresponding course materials published before 2018 and the search for new key concepts in recent academic publications. The first stage implies searching for the words that can potentially become new key concepts. This process is associated with the following filtration procedures:

1. Compilation of a list of words included in 7 course materials on logistics published before 2018.
2. Compilation of a "texts-words" matrix for a set of 250 academic texts on logistics published in the period from 2016 to 2020.
3. Exclusion of words presented in the dictionary that is compiled based on course materials on logistics from the matrix.
4. Exclusion of frequently encountered words presented in a large number of academic texts from the matrix.
5. Selection of rare words from those remaining in the matrix and generation of bigrams, that is, two-word phrases that can potentially become new key concepts in logistics.

These pairs of words are well correlated within particular academic texts on logistics. It should be noted that to limit the scope of the problem during the search for the correlated pairs of words, only those words that are used in one text from the analyzed corpus are singled out from the entire list. Such words are further analyzed separately.

The obtained list of potential two-word key concepts is subject to search for simple phrases containing both candidate words from academic texts. In this case, a simple phrase means a phrase without punctuation marks (except for dashes and quotation marks) containing two words that can potentially become a two-word key concept. Here, the automated part of the procedure aimed at identifying a two-word key concept is over and manual selection of phrases containing possible concepts begins. This selection is necessary to exclude widespread terms from the list.

Let us have a look at the results of the material collecting procedure. In this case, the proposed method is not the best option because the query includes only one word (logistics) without any specification of the subject. Let us demonstrate how the consistent application of ONT-01 programs reduces the search sample and increases the quality of the remaining material.

There are groups of bigrams different in their characteristics in relation to the time of their appearance and the contents of the dictionary of the curriculum of the Logistics academic discipline that was compiled based on the text published in 2018.

For example, 13 out of 50 abstracts of academic articles published in 2020 have no overlaps with the dictionary of the curriculum. There are also bigrams and one-word concepts that appeared only in 2018 or 2019 and thus are not included in the curriculum.

A relatively short list of new logistic terms was obtained after the exclusion of words used in annotations that coincided with textbooks' dictionaries. Let us examine the main phrases consisting of keywords for the period from 2016 to 2020.

1. Airships, cruise and cargo submarines (especially for the Arctic region), string transport, private spaceships allowing flights to the Moon and Mars constitute advanced and innovative means of transportation;
2. Development of the Russian logistics infrastructure in the Arctic region as a factor of global competition;
3. Boat pilot;
4. Economic Order Quantity (EOQ) model;
5. The last mile;
6. Urban sprawl;

Table 2. For the sample consisting of 50 texts per year

Year	Words found	Hapaxes found
2016	5172	3718
2017	5610	2116
2018	5103	1877
2019	5351	2190
2020	6133	2633
The curriculum of the logistics academic discipline	709	162

7. Unmanned aerial vehicle (UAV);
8. Unmanned technical;
9. Unmanned civilian aircraft systems;
10. Building Information Model (BIM) or Modeling - an information model (or modeling) of buildings and constructions including any infrastructural object, for example, utility networks (water, gas, electricity, sewer, communication), highways, railroads, bridges, ports, tunnels, etc.;
11. Geoinformation and satellite;
12. Cross-border dynamics;
13. Discounters;
14. Additional loading of drones;
15. Green logistics;
16. Cross-border online stores;
17. Intralogistics - logistics within four walls;
18. Corruption risks;
19. Criminal threats;
20. Criminal fraud;
21. Logistic bush as a set of branches;
22. Logistics mix. Logistics mission;
23. Pilot boat;
24. Network healthcare companies;
25. Logistics mission mix;
26. Pilot boats' modernization;
27. Multi-agent systems;
28. Ineffective use of WMS;
29. Omnichannel retail;
30. The last mile;
31. SAP programming environment;
32. Uberization;
33. Supply chains with regard to logistic bush;
34. Digital transformation of the last mile in logistics;
35. The fourth industrial revolution;
36. The Silk Road.

Economic coordination of the EAEU and the Silk Road project.

The following topics and concepts should be emphasized:

- Drones and unmanned aerial vehicles as well as "the last mile" in the "delivery chain";
- Digitalization (in a temporal perspective). The curriculum contains no information about this phenomenon;
- International logistics in comparison with the dictionary and curriculum text;
- Examples of bigrams that are not included in the curriculum but are worth including in an updated version of the document.

4 Overview of Results of the Analysis of the Query "Logistics" in e-Library

As the result of the conducted analysis, about 40 new key logistics concepts consisting of two-five

words were singled out from about 18500 words used in 250 academic publications that were filtered with the help of 15500 words used in textbooks and course materials on logistics. The list of new concepts contains the following important development areas:

- The Arctic region and the Northern Sea Route;
- Specification of port logistics;
- The last mile problem and unmanned aerial vehicles;
- Information logistics software - specification;
- Partial specification of the already known key concepts of a higher abstraction level.

The latter areas can be too detailed to include them in the hierarchical structure of logistics ontology that is suitable for the learning process. Thus, the software can become obsolete before it is included in the courses. The other development areas are promising enough to be included in the curricula.

In the ancient world, logistics was defined as the art of army supply and control over its movement. The military experience was later used in the civilian economy and transformed into a new subject area related to material flows management both in distribution and production. A similar situation is currently with such concepts as the last mile and unmanned aerial vehicles (even researchers' affiliation and places of publication indicate this [11, 12]). However, rapid economic changes caused by the pandemic can transform the last mile problem into the development of land transportation methods in retail.

The semantics of a simple phrase paired with a two-word concept included in this phrase can point to the fact that this two-word concept is not a fixed collocation. However, together with other words from this phrase, a bigram can help to identify a collocation (it does not have to consist of two words) that has better chances to become a concept or a keyword. Results of the final stage of manual selection remain a hypothesis until they are tested by a subject area expert. If this cannot be done promptly, an online query is a possible way out. This query should consist of a bigram and a single phrase or a part of the phrase containing this bigram.

Let us give an example. A pair of words "logistics mix" was identified in the following phrase: "it is proposed to define the logistics mission of industrial enterprises not as a logistics mix." The contents of course materials on logistics were checked for the use of this pair of words and it was not found. The same result was obtained for the pair of words "logistics mission." A query in the search engine Yandex was made consisting of words from the simple phrase that contained the original pair of words. The query had the following form: "logistics, mission, industrial, enterprises, logistics, mix." The obtained search results constituted at least 60 links where the concept of "logistics mission" was mentioned. The query for the pair "logistics mix" was less successful.

However, one of the most popular research results contains the following information: "Logistics mix (the "7R rule"), i.e., getting the right product, in the right quantity, in the right condition, at the right place, at the right time, to the right customer, at the right price" ([21], p. 465).

Less popular research results include such concepts as "logistics mission, marketing, and logistics mix" and "7R." For example, the article "Logistics mission and environment of a company" contains the following phrase: "In this regard, logistics mission is often interpreted by foreign specialists as the 7R rule or logistics mix (similar to marketing mix)[1]". Moreover, the phrase "Marketing and logistics mix, examples of interaction between sciences" is presented in an article on the economy and economic theory[2]. The aforementioned terms are not new and the authors of education courses should decide whether to include them in the logistics curriculum or not. At the curriculum level, the course on logistics does not contain the term "the last mile." However, it is mentioned in two texts published in 2016.

5 Conclusion

The proposed technology of searching for words that can potentially become keywords proves its

[1] lektsii.org/3-86757.html

[2] otherreferats.allbest.ru/economy/d0022214.html

efficiency. This technology should be applied annually to monitor the dynamics of the appearance of new keywords that are not included in the course materials. The following observations are worth noting:

- The concentration of a large number of bigrams in a small number of texts. For example, the fragment "Airships, cruise and cargo submarines (especially for the Arctic region), string transport, private spaceships allowing flights to the Moon and Mars constitute advanced and innovative means of transportation" [13] is a quote from the text published in 2015 [14] where the word "logistics" is not used.
- The concentration of bigrams in such parts of texts, which are rich in logistics keywords.

The main disadvantages are:

- Inclusion of articles that are written not in Russian but use the Cyrillic alphabet in the sample;
- Multiple repetitions of one phrase in the section "Context" in the reference list.

Identification of rare keywords that are not presented in textbooks of higher education institutions on a relevant subject is a poorly formalized research problem the solution to which does not guarantee success due to at least three reasons:

- The object of search may not be presented in the analyzed material;
- Recognition of a previously unused fixed collocation as a new key concept is subjective;
- Some steps in searching for bigrams are associated with the loss of mainly irrelevant information.

Lists of stop-words should be altered together with a change of the analyzed subject area. However, keeping the lists of excluded words in a particular year allows saving some time for manual processing at the stage of the preparation of data for the next analysis step.

If it is necessary to find 100 main innovations that should be included in the course, it cannot be done based on a sample. To do this, every innovation should be examined separately. Moreover, in some cases, the model (even if it is generated on the basis of a sample) should be applied to the entire set of data.

When it comes to the practical application of the model, the option with samples is not suitable. Thus, it is feasible to generate annual life cycle curves for the identified concepts even if they do not change gradually over time because their behaviors can be important for planning future research. Comparing different concepts' behaviors is of particular interest.

References

1. **Makagonov, P.P. (2020).** Identification of new key bigrams for constructing a prototype of digital economy and cybersecurity ontology. Report at the International Conference Session Public administration and development of Russia: Global threats and structural changes. Section: Change management: Challenges of Digital Civilization [in Russian].
2. **Balandina, G.V., Ponomarev, Y.Y., Sinelnikov-Murylev, S.G. (2020).** Customs administration in Russia: What modern procedures should look like. Economic Policy, Vol. 15, No. 1, pp. 108–135.
3. **Ruiz-Figueroa, A., Makagonov, P. (2007).** Hardware and software development models based on the study of parallel computing. Interciencia, Vol. 32, No. 3, pp. 160–166.
4. **Gadzhinsky, A.M. (2007).** Logistics: Textbook for higher education students of specialty Economy, 15th ed., Moscow [in Russian].
5. **Baranovsky, S.I., Shishlo, S.V. (2014).** Logistics: Texts of lectures for students of specialty 1-25 01 07 "Economy and management in industry" of intramural and extramural forms of study. Belarusian State Technological University, Minsk [in Russian].
6. **Zhigalova, V.N. (2013).** Logistics: Textbook. Tomsk State University of Control Systems and Radioelectronics, Tomsk. Publisher: El Content, pp. 165 [in Russian].

7. **Konotopsky, V.Y. (2014).** Logistics: Textbook. Tomsk, 2014, pp. 139, 2 UDC 336 К 64 [in Russian].
8. **Shash, N.N., Azimov, K.A., Shepeleva, A.Y. (2010).** Logistics: Compendium of lectures. Publisher: Yurait, pp. 205. ISBN 978-5-9916-0592-2, UDC 33, LBC 65.40я73 Ш32 М [in Russian].
9. **Voronkov, A.N. (2013).** Logistics: Basic principles of operating activity: Textbook. Nizhny Novgorod State University of Architecture and Civil Engineering. Nizhny Novgorod, pp. 168. LBC 65.291.592 [in Russian].
10. **Chudakov, A.D. (2001).** Logistics: Textbook. Moscow. Publisher: RDL, pp. 480 [in Russian].
11. **Kurbanov, T., Starchenko, D., Zaikin, A. (2020).** Drones in logistics: Experience of leading foreign and domestic companies, prospects, and application problems. Logistics, A.V. Khrulev Military Academy for Logistics and Volsk Military Institute of Logistics. Moscow. Publisher: Market Guide Agency. Vol. 2, No. 159, pp. 26–29 [in Russian].
12. **Dmitriev, A.V. (2019).** Logistics: Current development trends. Proceedings of the XVIII International research and practical conference. Saint-Petersburg. Publisher: Admiral Makarov State University of Maritime and Inland Shipping, pp. 154–161. UDC 658.7 [in Russian].
13. **Vladimirov, S.A. (2016**). On major development directions of global transportation system and logistics. Transport Information Bulletin, Mytishchi. Publisher: Individual Entrepreneur Davydov, G.E. Vol. 1, No. 247, pp. 13–19 [in Russian].
14. **Vladimirov, S.A. (2016).** On major development directions of global transportation system and logistics. Transport Messenger, No. 2, pp. 2–8. Moscow. Publisher: Transport Messenger editorial office [in Russian].
15. Logistic paradigms: Functional, resource, innovative. https://mybiblioteka.su/tom3/3-6190.html.
16. **Zagorulko, Y.A., Borovikova, O.I. (2007).** Ontology-building technology for scientific portals. Vestnik NSU, Series: Information technology, Vol. 5, No. 5, pp. 42–52 [in Russian].
17. **Volegzhanina, I.S. (2019).** Establishment and development of engineer professional competence in context of digital transformation of industry (by the example of universities of transport). Pedagogical Journal, Vol. 9, No. 3A, pp. 189–198 [in Russian].
18. **Adolf, V.A., Volegzhanina, I.S. (2019).** Concept of establishment and development of professional competence of industrial staff in research and educational complex. Pedagogical Journal, Vol. 9, No. 1A, pp. 346–355. DOI: 10.34670/AR.2019.44.1.064.
19. **Leshcheva, I.A., Leshchev, D.V. (2014).** Analysis of dynamics of changes in an academic field by methods of ontological engineering. Open Semantic Technologies for Designing Intelligent Systems, Belarusian State University of Informatics and Radioelectronics, Minsk. ISSN: 2415-7740, UDC 001.53.No. 4, pp. 483–486 [in Russian].
20. **Novikov, A.Y., Golikov, I.Y., Zakharov, K.N., Satin, B.B. (2018).**On necessity to develop scenario-based ontology for modeling dynamics of changes in subjects areas. Scientific Thought, 2018, vol. 5, no. 3 (29). Cherepovets Military Command College of Radioelectronics. State Classifier of Scientific and Technical Information 28.23.35, UDC 004.891 [in Russian].
21. **Moiseeva, N.K. (2008).** Economic basis of logistics: Textbook. Moscow. Publisher: INFRA-M, pp. 528 [in Russian].

Article received on 01/07/2021; accepted on 27/11/2021.
Corresponding author is Pavel P. Makagonov.

ISSN 2007-9737

Evolutionary Instance Selection Based on Preservation of the Data Probability Density Function

Samuel Omar Tovias-Alanis, Wilfrido Gómez-Flores, Gregorio Toscano-Pulido

Instituto Politécnico Nacional,
Centro de Investigación y de Estudios Avanzados,
Mexico

{samuel.tovias, wgomez, gtoscano}@cinvestav.mx

Abstract. The generation of massive amounts of data has motivated the use of machine learning models to perform predictive analysis. However, the computational complexity of these algorithms depends mainly on the number of training samples. Thus, training predictive models with high generalization performance within a reasonable computing time is a challenging problem. Instance selection (IS) can be applied to remove unnecessary points based on a specific criterion to reduce the training time of predictive models. This paper introduces an evolutionary IS algorithm that employs a novel fitness function to maximize the similarity of the probability density function (PDF) between the original dataset and the selected subset, and to minimize the number of samples chosen. This method is compared against six other IS algorithms using four performance measures relating to the accuracy, reduction rate, PDF preservation, and efficiency (which combines the first three indices using a geometric mean). Experiments with 40 datasets show that the proposed approach outperforms its counterparts. The selected instances are also used to train seven classifiers, in order to evaluate the generalization and reusability of this approach. Finally, the accuracy results show that the proposed approach is competitive with other methods and that the selected instances have adequate capabilities for reuse in different classifiers.

Keywords. Instance selection, probability density function, evolutionary algorithm.

1 Introduction

Nowadays, ubiquitous computing and the Internet of Things are generating massive amount of multivariate data. As a result, researchers in many scientific and engineering fields have applied machine learning (ML) techniques to take advantage of this information for data modeling and decision making [5].

In supervised learning, ML algorithms are used to create prediction models based on a set of labeled training data. A dataset is typically represented by a matrix $X \in \mathbb{R}^{n \times d}$ composed of n instances and d predictor variables. Furthermore, the ith instance is represented by the vector $\mathbf{x}_i = [x_{i1}, \ldots, x_{id}]$, associated with an actual class label $y_i \in \Omega = \{\omega_1, \ldots, \omega_c\}$, where c is the total number of classes. Thus, training of the model implies the supervised learning of a mapping function $g : \mathbb{R}^{n \times d} \rightarrow \hat{y}$, where $\hat{y} \in \Omega$ is a predicted class label [11].

The computational complexity of ML algorithms depends mainly on the number of training instances and prediction variables. Thus, for extensive datasets, building a classification model within a reasonable computing time is a challenging problem. In addition, the model evaluation stage and hyperparameter tuning increase the training time [3]. Instance selection (IS) algorithms remove unnecessary patterns based on some elimination criterion. The aim is to select a representative subset (denoted by X_S) from the original dataset (denoted by X_O) to reduce the training time when building prediction models.

IS algorithms can be divided into wrapper and filter methods; the former uses a classification rate from a supervised learning algorithm, whereas the latter uses statistical information from data [13].

In addition, metaheuristic-based IS techniques generally use evolutionary algorithms (EAs), in which each member of the population represents a subset of selected instances. EAs encode the individuals as a fixed-length binary vector of size n_O, corresponding to the number of instances in X_O. In this encoding scheme, a value of 1 means that the corresponding instance is selected, whereas a value of 0 indicates the opposite. Optimization is commonly performed using a wrapper scheme that maximizes both the classification accuracy and the reduction rate [9].

According to Reeves and Bush [17], the training set should be an accurate representation of the actual probability distribution over the input space. However, selecting instances using an EA-based wrapper scheme may lead to solutions that fulfill the classifier's criteria but cannot preserve the original probability distribution. For instance, if a support vector machine (SVM) is used, the selected instances may be biased towards the local distribution of the support vectors. We can therefore state that the selection of instances should be made only once, and the resulting data subset can then be used to train different classifiers without loss of generalization, thus avoiding the need to repeat the selection process for each type of classifier.

This work presents an evolutionary IS method based on a filter approach, in which the fitness function incorporates both preservation of the probability density function (PDF) and a reduction rate. The underlying concept is to preserve the original data distribution by maximizing the similarity between the X_O and X_S PDFs, while reducing the number of instances in X_S. Both objectives are combined through the use of a weighted sum, and a global optimization scheme based on a genetic algorithm (GA) with binary encoding is used to carry out instance selection.

2 Related Work

2.1 Classical IS Methods

Classical IS methods generally reduce the number of instances using the nearest neighbor rule. These approaches can be divided into condensation, edition, and hybrid methods. The condensed nearest neighbor (CNN) method retains points closer to the decision boundaries, while internal points are removed, since they do not affect the decision boundaries [14]. The edited nearest neighbor (ENN) approach preserves internal samples while removing points closer to the decision boundaries with class labels that are different from their neighboring points (i.e., noisy points) [20]. The decremental reduction optimization procedure (DROP3) is a hybrid method that removes border points by first applying ENN to filter noisy instances, and then removes internal instances far from the decision boundaries [21]. Finally, the iterative case filtering algorithm (ICF) produces data clusters based on reachable and coverage sets, where points with a reachable set size greater than the coverage set size are removed [4].

2.2 Evolutionary IS Methods

An IS based on an EA is generally classed as a wrapper scheme. In this context, Kuncheva [15] proposed a GA with binary encoding, where the fitness function measured the error rate of the k-nearest neighbors (kNN) classifier.

In another work, Cano et al. [6] analyzed the performance of four binary-based representation EAs. The objective function adopted in this case was a weighted sum of the classification error and the reduction rate, with the same relative importance. Likewise, Garcia et al. [12] proposed a memetic algorithm that combined the heuristic approach of population-based algorithms with local search methods. The fitness function in this approach maximized both the accuracy and the reduction rate.

Aldana et al. [2] introduced a method based on an eclectic GA (EGA). This approach adopted a binary string encode in which two positive integers represented the number of randomly sampled instances. The EGAs objective function evaluated the reduction rate, and two constraints were considered: the error between the original and selected sets, and the proportion of elements between the quantiles of both sets.

Rosales-Perez et al. [18] used a multiobjective EA to solve the IS problem. Their solution encoded the reduction technique and the hyperparameters of an SVM, and the two objective functions were the classification rate and the reduction rate.

3 Kernel Density Estimation

Kernel density estimation (KDE) is a non-parametric method for estimating the PDF of a random variable, and can handle an arbitrary distribution without requiring any assumptions about the form of its underlying density [11].

Let $x_1, x_2, \ldots, x_n$ be independent and identically distributed samples, taken randomly from a distribution with unknown density $p(x)$. KDE in a region $\mathcal{R}$ centered at $\hat{x}$ is given by:

$$\hat{p}_h(\hat{x}) = \frac{1}{nh} \sum_{i=1}^{n} \phi_{\mathcal{N}} \left(\frac{\|\hat{x} - x_i\|_2}{h} \right), \tag{1}$$

where $\| \cdot \|_2$ denotes the Euclidean distance, $\phi_{\mathcal{N}}(\cdot)$ is the Gaussian kernel function with zero mean and unit variance, expressed as:

$$\phi_{\mathcal{N}}(u) = \frac{1}{2\pi^{(1/2)}} \exp\left(-\frac{u^2}{2} \right), \tag{2}$$

and $h > 0$ is a smoothing parameter, also known as the bandwidth. This parameter must be fine-tuned, since it has a strong influence on the result of the density estimation. When $h \to 0$, the shape of the estimated PDF is noisy and may include spurious peaks; conversely, if $h \to \infty$, the shape of the estimated PDF is over-smoothed.

The optimality criterion that is typically applied to select h is the expected L_2-risk function, also known as the mean integrated squared error (MISE). In this work, we use the direct plug-in rule (DPI), which is a method for automatically selecting a near-optimal h value by minimizing the MISE quality estimates (ψ). The following steps are used to calculate h based on the DPI rule [19]:

1. Estimate ψ_8 using an estimator of dispersion $\hat{\sigma}$, such as the median absolute deviation:

$$\hat{\psi}_8^{\hat{\sigma}} = \frac{105}{32\pi^{1/2}\hat{\sigma}(x)^9}, \tag{3}$$

2. Estimate ψ_6 using the estimator $\hat{\psi}_6(g_1)$, where:

$$g_1 = \left(\frac{11.9683}{\hat{\psi}_8^{\hat{\sigma}} n} \right)^{1/9}, \tag{4}$$

3. Estimate ψ_4 using the estimator $\hat{\psi}_4(g_2)$, where:

$$g_2 = \left(-\frac{2.3937}{\hat{\psi}_6(g_1) n} \right)^{1/7}, \tag{5}$$

4. The value of the bandwidth h is then calculated as:

$$h = \left(\frac{0.2821}{\hat{\psi}_4(g_2) n} \right)^{1/5}. \tag{6}$$

In Steps 2 and 3, the estimator $\hat{\psi}_r(g)$ is:

$$\hat{\psi}_r(g) = \frac{g^{(-r-1)}}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} \phi_{\mathcal{N}}^{(r)} \left(\frac{x_i - x_j}{g} \right), \tag{7}$$

where $\phi_{\mathcal{N}}^{(r)}$ is the rth derivative of the kernel $\phi_{\mathcal{N}}$.

4 Proposed Approach

Finding the optimal subset of instances in the IS task implies exploring a search space of size $2^{n_O} - 1$. This number reflects the possible subsets X_S, with cardinality $n_S = 1, \ldots, n_O - 1$, chosen from X_O with n_O instances. Although the search space is finite, it grows exponentially, making an exhaustive exploration intractable, and IS is therefore generally addressed as an optimization problem, using metaheuristics to find a sub-optimal solution within a reasonable computing time. In this following, we describe the proposed evolutionary IS method based on a PDF preservation approach.

4.1 Fitness Function

We propose a novel fitness function for use in an evolutionary IS algorithm. It has two components: (i) maximizing the similarity between the X_O and X_S PDFs; and (ii) minimizing the number of instances in X_S.

This first component uses the Hellinger distance to measure the distributional divergence. The Hellinger distance between two densities p and q is defined as [8]:

$$\mathcal{H}(p,q) = \left(\frac{1}{2}\int\left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 dx\right)^{1/2}. \quad (8)$$

and satisfies the property $0 \leq \mathcal{H}(p,q) \leq 1$, where a value closer to zero indicates a higher similarity between the densities p and q. Hence, maximizing the similarity between p and q implies minimizing the Hellinger distance.

It is worth noting that p and q are univariate density functions. In contrast, the dataset X_O usually contains instances in $\mathbb{R}^d$. To handle multivariate data, the Hellinger distance is calculated for each predictor variable for each class, to form the following matrix:

$$H = \begin{pmatrix} \mathcal{H}_{1,1} & \cdots & \mathcal{H}_{1,d} \\ \vdots & \ddots & \vdots \\ \mathcal{H}_{c,1} & \cdots & \mathcal{H}_{c,d} \end{pmatrix}, \quad (9)$$

where $\mathcal{H}_{i,j} \equiv \mathcal{H}(p_{i,j}, q_{i,j})$ is the Hellinger distance between the original, $p_{i,j}$, and approximated, $q_{i,j}$, densities of the jth variable in the ith class. The PDF of a predictor variable is obtained using KDE, as described in Section 3. It is worth mentioning that the DPI rule is calculated on the jth dimension of the ith class of X_O (for $i = 1, \ldots, c$ and $j = 1, \ldots, d$), and the resulting bandwidth $h_{i,j}$ is used to estimate the densities $p_{i,j}$ and $q_{i,j}$.

The second objective is addressed by measuring the fraction of selected instances as:

$$s_f = \frac{n_S}{n_O}, \quad (10)$$

where n_S and n_O denote the number of instances in X_S and X_O, respectively. This objective varies in the range $[0, 1]$, and a value close to zero indicates an adequate reduction rate.

Finally, the fitness function is used to combine the two criteria in (9) and (10) through a weighted sum, as follows:

$$f = \frac{w}{c \cdot d}\sum_{i=1}^{c}\sum_{j=1}^{d}\mathcal{H}_{i,j} + (1-w)\cdot s_f, \quad (11)$$

where $w \in (0, 1)$ is a weight coefficient that expresses the importance of each objective. This weighted fitness function varies in the range $[0, 1]$, where an aptitude value close to zero indicates that X_S achieves a high reduction rate and preserves the probability distribution of X_O. We evaluated the impact of w by varying its value in the range $[0.50, 0.95]$ with steps of 0.05. Algorithm 1 presents a pseudocode for the evaluation of the fitness function.

4.2 Evolutionary IS

In this work, we use a GA to minimize the objective function in (11). Each individual in the population is a fixed-length binary vector of size n_O, where a value of 1 means that the corresponding instance in X_O is selected, and 0 indicates the opposite.

Solutions are randomly initialized with a discrete uniform distribution in the range $[0, 1]$. The parent selection strategy uses a two-way tournament approach. Next, a two-point crossover is performed in the recombination step to exchange the selected parents' genetic information to generate new offspring. Then, based on the mutation probability factor, the mutation operator performs a bit flip in random positions of each offspring vector. An elitist strategy is also applied to ensure that the quality of the solution does not decrease over the generations. Finally, the GA returns the best individual in the last generation.

In this case, a population of 100 individuals was evolved over 2000 generations. The crossover and mutation probabilities were set to 0.9 and $1/n_O$, respectively.

5 Experimental Setup

5.1 Datasets

The datasets used in the experiments were obtained from the KEEL repository [1] and the UCI Machine Learning Database [10]. Table 1 summarizes the characteristics of 40 small datasets (with no more than $5,456$ instances). In order to test the performance of the proposed method with large datasets, two medium-sized datasets were considered: Magic Gamma Telescope (MGT) with

Algorithm 1 Fitness function evaluation

Input: Individual $\mathbf{q} \in \{0,1\}^{n_O}$, original dataset normalized in the range $[-1,1]$: $\hat{X}_O \in \mathbb{R}^{n_O \times d}$, set of density estimates of $\hat{X}_O$: $\{p_{1,1},\ldots,p_{c,d}\}$, set of bandwidths: $\{h_{1,1},\ldots,h_{c,d}\}$, $\mathcal{R} = 100$ equidistant points in the range $[-1.5,1.5]$: $\hat{\mathbf{x}} = [\hat{x}_1,\ldots,\hat{x}_{\mathcal{R}}]$ and the weight of the fitness function: w
Output: Fitness value: f

Decode $\mathbf{q}$ to obtain the selected subset with n_S instances from $\hat{X}_O$: X_S
Get the number of classes of $\hat{X}_O$: c
Get the number of classes of X_S: c_S
if $c = c_S$ **then**
  Initialize the cumulative sum of the values of H (9): $\Sigma_{\mathcal{H}} = 0$
  for $i = 1$ to c **do**
    for $j = 1$ to d **do**
      Get the values of the jth variable from the ith class of X_S: $\mathbf{x}_{i,j} = [x_1,\ldots,x_n]$
      Compute KDE (1) for each point in $\hat{\mathbf{x}}$ using $\mathbf{x}_{i,j}$ and $h_{i,j}$: $q_{i,j}$
      Compute the Hellinger distance (8): $\mathcal{H}_{i,j} \equiv \mathcal{H}(p_{i,j}.q_{i,j})$
      Update the cumulative sum of the elements of H: $\Sigma_{\mathcal{H}} = \Sigma_{\mathcal{H}} + \mathcal{H}_{i,j}$
    end for
  end for
  Compute the average of H: $\mu_{\mathcal{H}} = \Sigma_{\mathcal{H}}/(c \cdot d)$
  Compute the second objective: $s_f = n_S/n_O$
  Compute the fitness function (11): $f = w \cdot \mu_{\mathcal{H}} + (1-w) \cdot s_f$
else
  Penalize solution if one or more classes are eliminated: $f = 1$
end if
return f

$n = 19,020$, $d = 10$, and $c = 2$, and Letter Recognition (LT) with $n = 20,000$, $d = 16$, and $c = 26$.

5.2 Instance Selection Methods

The proposed approach, denoted as F_w (i.e., the proposed filter method with a specific weight value w), was compared against six other IS methods.

The first alternative approach was an evolutionary IS algorithm based on a wrapper scheme. This method used the same GA as the proposed approach but applied a fitness function that is commonly adopted in the literature, in which a weighted sum is used to combine the classification accuracy and the reduction rate with the same relative importance [17, 6, 12]. When evaluating the fitness function, the selected subset X_S, given by a potential solution, is used to train the classification model, whereas the validation set is obtained as $X_V = X_O - X_S$, and is used for measuring the classification accuracy. Two classifiers are considered, SVM and kNN, and the two variants of this approach are denoted as W_{SVM} and W_{kNN} (i.e., the wrapper method with SVM and kNN, respectively).

In addition, the soft margin parameter (C) and the bandwidth of the Gaussian kernel (γ) used for the W_{SVM} algorithm were found using the grid search method in the ranges $C = [2^{-5}, 2^{-3}, \ldots, 2^{15}]$ and $\gamma = [2^{-15}, 2^{-13}, \ldots, 2^{3}]$, with 5-fold cross-validation [7]. Appendix 7 lists the C and γ hyperparameters found.

The four remaining IS methods are CNN, ENN, DROP3, and ICF, as depicted in Section 2.1. The number of nearest-neighbors for W_{kNN} and the classical methods are set to $k = 3$.

5.3 Performance Assessment

The performance of the IS methods was measured using four indices:

Table 1. Characteristics of the datasets: n is the number of instances, d is the dimensionality, and c is the number of classes

ID	Dataset	n	d	c	ID	Dataset	n	d	c
1	Appendicitis	106	7	2	21	Ionosphere	351	33	2
2	Australian	690	14	2	22	Iris	150	4	3
3	Balance	625	4	3	23	Led7digit	500	7	10
4	Banana	5,300	2	2	24	Mammographic	830	5	2
5	Bands	365	19	2	25	Monk-2	432	6	2
6	Breast	277	9	2	26	LIBRAS	360	90	15
7	Bupa	345	6	2	27	New Thyroid	215	5	3
8	Car	1,728	6	4	28	Pima	768	8	2
9	Cleveland	297	13	5	29	Saheart	462	9	2
10	Contraceptive	1,473	9	3	30	Sonar	208	60	2
11	Crx	653	15	2	31	Spectfheart	267	44	2
12	Dermatology	358	34	6	32	Tae	151	5	3
13	Flare	1,066	11	8	33	Tic-Tac-Toe	958	9	2
14	German	1,000	20	2	34	Vehicle	846	18	4
15	Glass	214	9	6	35	Vowel	990	13	11
16	Haberman	306	3	2	36	Wall Following	5,456	2	4
17	Hayes-Roth	160	4	3	37	WDBC	569	30	2
18	Heart	270	13	2	38	Wine	178	13	3
19	Hepatitis	80	19	2	39	Wisconsin	683	9	2
20	Housevotes	232	16	2	40	Yeast	1,484	8	10

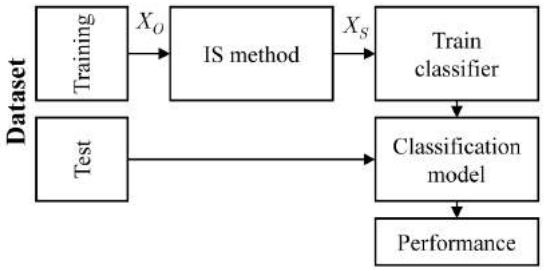


Fig. 1. Evaluation framework for IS methods

— Accuracy (ACC): A classifier was trained using the selected instances, and the accuracy (success rate) was measured on an independent test set.

— Reduction rate (RR): The fraction of removed instances was calculated as $1 - s_f$, where s_f is given by (10).

— Hellinger distance complement (HDC): The similarity between the densities X_O and X_S was calculated using the mean Hellinger distance (HD) for the matrix in (9). From a maximization perspective, HDC $= 1 -$ HD.

— Efficiency (E): The geometric mean $\sqrt[3]{\text{ACC} \times \text{RR} \times \text{HDC}}$ was used to calculate the tradeoff between accuracy, reduction rate, and PDF preservation.

A t-times k-fold cross-validation method (where $t = 10$ and $k = 5$) was used to split the small datasets into training and test sets. This resampling process reduced the influence of randomness introduced by data splitting [22]. To divide the medium-sized datasets, a 10-fold cross-validation technique was applied. The procedure illustrated in Fig. 1 was then performed on each fold.

The reusability of the selected instances is related to the ability to train different classifiers without losing generalization. In this sense, in wrapper techniques, the subset X_S could fulfill the classifier's criteria to increase the accuracy, but loses similarity with the X_O distribution when the number of instances is reduced; thus, X_S may be useless for training other types of classifiers. To measure the reusability of X_S, we use two types of accuracy:

— Type 1, which measures the classification performance on the test set using only the classifier within the wrapper method.

— Type 2, which measures the classification performance on the test set using different classifiers that are not used by the wrapper method.

Seven classifiers were considered when measuring the classification accuracy: classification and regression tree (CART), linear discriminant analysis (LDA), quadric discriminant analysis (QDA), naïve Bayes classifier (NB), radial basis function network (RBFN), SVM, and kNN. Furthermore, $\max(3, \sqrt{n})$ hidden nodes were used for the RBFN architecture (where n is the number of the training instances), the number of nearest neighbors in kNN was set to k=3, and the hyperparameters of the SVM classifier were tuned for each subset using the grid search method depicted in Section 5.2 [11, 3].

The non-parametric Kruskal-Wallis test, followed by a Bonferroni correction ($\alpha = 0.05$), was performed for multiple comparisons to determine the statistical significance between the proposed

approach and the six IS methods in terms of the four indices listed above.

Additionally, the McNemar test ($\alpha = 0.05$) was used to statistically assess the accuracy of two classification models trained with X_O and X_S against the actual labels. It detects whether the difference between the misclassification rates is statistically significant. The null hypothesis establishes that the two predicted class labels, $\hat{y}_1$ and $\hat{y}_2$, have equal accuracy when predicting the actual class labels, y.

The testing platform used a computer with four cores at 3.5 GHz (Intel i7 4770k) and 32 GB of RAM. All the algorithms were developed in MATLAB 2018b [16], and the source codes are available upon request to the authors.

6 Results

6.1 Instance Selection Performance on Small Datasets

Fig. 2 shows the average performance results on the 40 small datasets, for all of the classifiers specified in Section 5.3. For the proposed method, the F_w performance changed according to w, as expected. As w increases, the values of ACC and HDC also increase, while the values of E and RR decrease. However, the first five values of w (i.e., 0.50 to 0.70) produced the same efficiency (E=0.81), which was the highest for all of the IS methods. Furthermore, F_{65} and F_{70} yielded a value of ACC=0.70 and a fairly similar PDF preservation (HDC=0.93). F_{65} gave the best tradeoff, as it achieved a higher reduction rate (RR=0.85).

Moreover, the F_w variants gave better PDF preservation than the wrapper and classical IS algorithms. Notably, only ENN achieved the same Hellinger distance complement as F_{50} (HDC=0.90), the F_w variant with the lowest PDF preservation.

Of the classical IS methods, ENN obtained the best accuracy (ACC=0.76) and PDF preservation (HDC=0.90), although it had the worst efficiency of all of the methods (E=0.51) due to the low reduction rate (RR=0.24). CNN obtained the second-best value of accuracy (ACC=0.71) and PDF preservation (HDC=0.89) of the classical techniques, although it outperformed ENN in terms of efficiency (E=0.71) and reduction rate (RR=0.59). The hybrid methods yielded better reduction rates and efficiency but lower accuracies and PDF preservation than CNN and ENN. For instance, DROP3 had a better reduction rate (RR=0.83) and efficiency (E=0.74) than ICF, but the lowest accuracy (ACC=0.67) and PDF preservation (HDC=0.76) of all the methods. In contrast, ICF achieved a better accuracy (ACC=0.70) and PDF preservation (HDC=0.81), but a lower reduction rate (RR=0.72) and efficiency (E=0.73) than DROP3.

The wrapper methods achieved better efficiency and reduction rate than the classical techniques. Specifically, W_{kNN} obtained the highest efficiency (E=0.76), and W_{SVM} the best reduction rate (RR=0.87). However, in terms of the accuracy index, the wrapper methods did not outperform the classical ones. For example, W_{kNN} had the highest classification performance of the wrapper methods, but this was the same as for DROP3 (ACC=0.67), which was the worst of the classical algorithms in terms of accuracy. Regarding PDF preservation, the wrapper methods were again surpassed by CNN and ENN; however, W_{kNN} slightly outperformed ICF, while W_{SVM} had the second-worst HDC index, only outperforming DROP3.

Fig. 3 shows a detailed comparison of the results from F_{65} against all the IS methods. Tables 2 and 3 show the resulting p-value of the Kruskal-Wallis test for multiple comparisons of each performance index.

Regarding accuracy, F_{65} attained the third-best result, and was only outperformed by CNN and ENN; still, according to Table 2, there are no statistical differences from IS methods. Although F_{65} obtained the second-best value in terms of reduction rate, there was no significant difference from W_{SVM}, which achieved the highest performance for this index.

These methods also showed statistical differences from CNN, ENN, and ICF. Despite the high reduction rate of F_{65}, it achieved the highest PDF preservation and showed statistical differences from W_{kNN}, W_{SVM}, DROP3, and ICF on the HDC index.

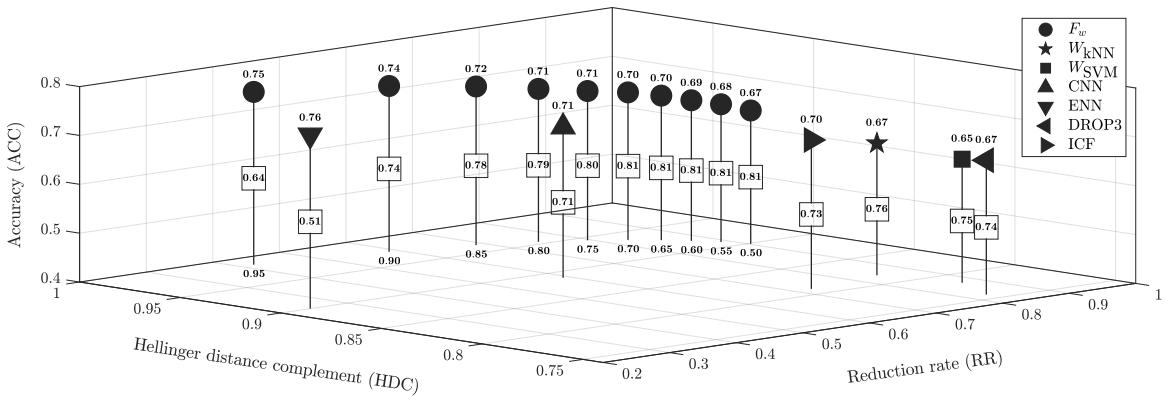


Fig. 2. Average performance results. They were calculated over the seven classifiers on the 40 datasets. Above each marker is shown the accuracy (ACC). The squared label shows the efficiency (E). The weight values (w) for the proposed approach F_w are shown at the bottom

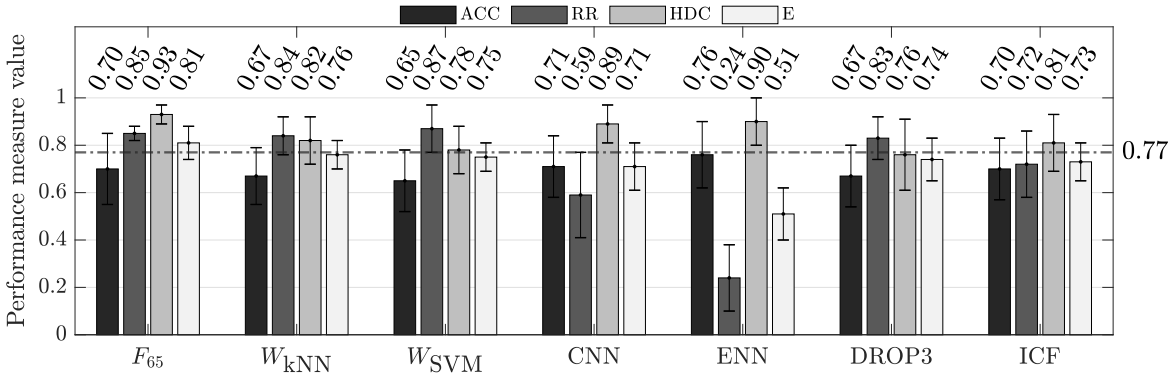


Fig. 3. Average performance results over the seven classifiers on the 40 datasets. Above each bar, the value of the corresponding measure. The dashed line marks the average accuracy obtained by the seven classifiers trained on the original datasets

For the efficiency, F_{65} attained the highest value, and the results in Table 3 show that this method gave statistical differences from all of the IS methods except W_{kNN}.

These results suggest that for a specific value of w, the proposed approach selects a subset of instances that preserve the probability distribution of the original dataset with a high reduction rate. The selected subset is also useful for training distinct classification models with good generalization performance.

6.2 Instance Selection Performance on Medium-Size Datasets

The medium-sized datasets described in Section 5.1 were used to test the performance of the F_w method, with $w = 0.50$ to give both objectives in the fitness function the same relative importance. The accuracy was calculated as the average over the results from the seven classifiers described in Section 5.3.

Table 2. Bonferroni correction results. The upper triangular matrix shows the p-values for ACC, and the lower triangular matrix shows the p-values for RR. In bold, $p < 0.05$

	F_{65}	W_{kNN}	W_{SVM}	CNN	ENN	DROP3	ICF
F_{65}	-	1.000	1.000	1.000	1.000	1.000	1.000
W_{kNN}	1.000	-	1.000	1.000	0.158	1.000	1.000
W_{SVM}	1.000	1.000	-	1.000	**0.022**	1.000	1.000
CNN	**0.000**	**0.000**	**0.000**	-	1.000	1.000	1.000
ENN	**0.000**	**0.000**	**0.000**	**0.010**	-	0.355	1.000
DROP3	1.000	1.000	1.000	**0.000**	**0.000**	-	1.000
ICF	**0.007**	**0.016**	**0.000**	1.000	**0.000**	0.086	-

Table 3. Bonferroni correction results. The upper triangular matrix shows the p-values for E, and the lower triangular matrix shows the p-values for HDC. In bold, $p < 0.05$

	F_{65}	W_{kNN}	W_{SVM}	CNN	ENN	DROP3	ICF
F_{65}	-	0.297	**0.057**	**0.000**	**0.000**	**0.023**	**0.005**
W_{kNN}	**0.000**	-	1.000	0.437	**0.000**	1.000	1.000
W_{SVM}	**0.000**	1.000	-	1.000	**0.000**	1.000	1.000
CNN	1.000	**0.005**	**0.000**	-	**0.000**	1.000	1.000
ENN	1.000	**0.000**	**0.000**	1.000	-	**0.000**	**0.000**
DROP3	**0.000**	1.000	1.000	**0.000**	**0.000**	-	1.000
ICF	**0.000**	1.000	1.000	**0.012**	**0.000**	1.000	-

On the MGT dataset, the accuracy of the subsets was slightly better (ACC=0.81) than that obtained on the original dataset (ACC=0.80). The F_{50} method attained a regular efficiency (E=0.74) due to the poor reduction rate (RR=0.52) and a high PDF preservation (HDC=0.98).

Regarding the LT dataset, the accuracy on the original set (ACC=0.81) was slightly higher that attained on the subsets (ACC=0.79).

However, similarly to the MGT dataset, the efficiency was regular (E=0.73) due to the low reduction rate (RR=0.52) and the high PDF preservation (HDC=0.96).

6.3 Reusability of Selected Instances

The reusability results are shown in Fig. 4, where the Type 1 and 2 accuracies are displayed as pairs of box plots. The upper graphic relates to W_{kNN}, in which the Type 1 accuracy was measured only using the kNN classification, while the Type 2 accuracy was measured using the remaining six classifiers.

Likewise, the lower graphic considers W_{SVM}, in which the Type 1 accuracy was measured only using SVM classification, while the Type 2 accuracy was measured using the remaining six classifiers.

For the proposed approach F_w, the accuracy improved with the values of the weights, i.e., the higher the weight, the better the accuracy. For each weight value, the median values for the two types of accuracy were quite similar. This suggests that the proposed method can give similar Type 1 and 2 accuracies, independently of the classifier, due to the criterion used to preserve the PDF of the original data.

The classical methods gave slightly better performance for Type 1 than Type 2, and ENN attained the highest accuracy due to its lower reduction rate. On the other hand, W_{kNN} and W_{SVM} obtained consistently better accuracy for Type 1 than Type 2. These results confirm that X_S is biased towards the classifier characteristics in the wrapper method, limiting the reusability of the selected data for training other classifiers.

Table 4. $\log_2 C$ and $\log_2 \gamma$ are the logarithms of the soft margin parameter and the Gaussian kernel's bandwidth, respectively, corresponding to the SVM classifier used in the W_{SVM} method

ID	Dataset	$\log_2 C$	$\log_2 \gamma$	ID	Dataset	$\log_2 C$	$\log_2 \gamma$
1	Appendicitis	11	−5	21	Ionosphere	5	−1
2	Australian	1	−15	22	Iris	5	−7
3	Balance	13	−7	23	Led7digit	15	−15
4	Banana	3	−1	24	Mammographic	13	−11
5	Bands	1	−9	25	Monk-2	1	−1
6	Breast	1	−3	26	LIBRAS	5	−1
7	Bupa	9	−15	27	New Thyroid	9	−15
8	Car	3	−1	28	Pima	1	−15
9	Cleveland	9	−15	29	Saheart	11	−15
10	Contraceptive	11	−9	30	Sonar	3	−1
11	Crx	1	−15	31	Spectfheart	3	−15
12	Dermatology	13	−15	32	Tae	11	−13
13	Flare	1	−3	33	Tic-Tac-Toe	11	−7
14	German	−5	−7	34	Vehicle	9	−15
15	Glass	5	−5	35	Vowel	7	−3
16	Haberman	13	−13	36	Wall Following	15	1
17	Hayes-Roth	5	−5	37	WDBC	5	−15
18	Heart	9	−15	38	Wine	11	−15
19	Hepatitis	13	−15	39	Wisconsin	3	−13
20	Housevotes	3	−7	40	Yeast	5	1

Finally, Fig. 5 shows the number of datasets for which there was no rejection of the null hypothesis

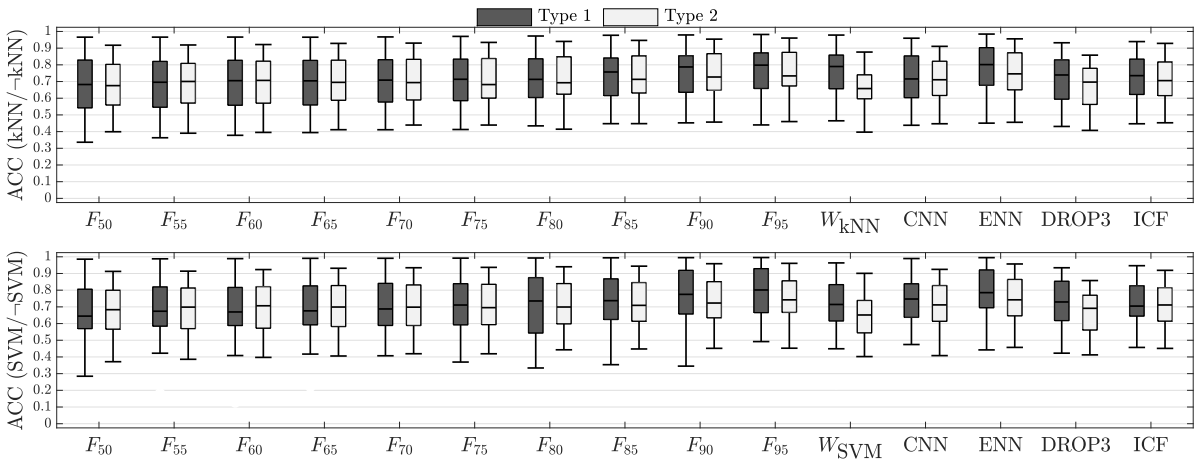


Fig. 4. Two types of accuracy results on the 40 datasets. Type 1: accuracy considering a single classifier, i.e., kNN (upper) and SVM (lower). Type 2: accuracy of six classifiers disregarding kNN (upper) and SVM (lower)

in the McNemar test for each classifier. The higher the count, the more similar the classification accuracy between models trained with X_O and X_S.

In the proposed approach F_w, the higher the values of the weights, the lower the number of rejections. Thus, F_{95} obtained the highest number of selected subsets that did not show a statistical difference from X_O for all of the supervised learning algorithms. Of the wrapper methods, W_{kNN} had a higher count of no rejections than W_{SVM}.

For the kNN classifier, W_{kNN} attained a significantly higher count than any other classifier, which was as expected since the selection criterion involved maximizing the accuracy of that specific classifier. W_{SVM} attained inferior results, even for the SVM classifier, where the count was not notably higher for the different classifiers, unlike the behavior of W_{kNN} for the kNN classifier.

Of the classical IS methods, ENN achieved a higher count in most cases, but gave a similar count to CNN for the kNN classifier, and was slightly outperformed by the same condensation method for RBFN. In terms of the number of counts for each classifier, the NB attained the highest value, and the QDA obtained the lowest counts for most IS methods.

These results reveal that the proposed method produces subsets of instances that can be reused to train different classifiers with a similar classification performance to that achieved from training with the original dataset.

6.4 Case Study on the Banana Dataset

Fig. 6 shows a case study carried out on the Banana dataset to compare the performance indices obtained by X_O and X_S. In terms of accuracy, F_{65} achieved a competitive result (ACC=0.74) for X_O and obtained better performance than its counterparts except for ENN. For the reduction rate, F_{65} removed more than 85% of the samples (RR=0.86) and surpassed W_{SVM}, CNN, and ENN. It also yielded a competitive reduction rate with regard to W_{kNN}, DROP3, and ICF.

For PDF preservation, F_{65} gave the highest performance (HDC=0.98), and the data points in the feature space show that the selected subset follows the distribution shape of the original dataset, despite the high reduction rate. In contrast, the selected subset generated by ICF had holes and clumps, and produced the worst PDF preservation of all the methods compared here (HDC=0.79).

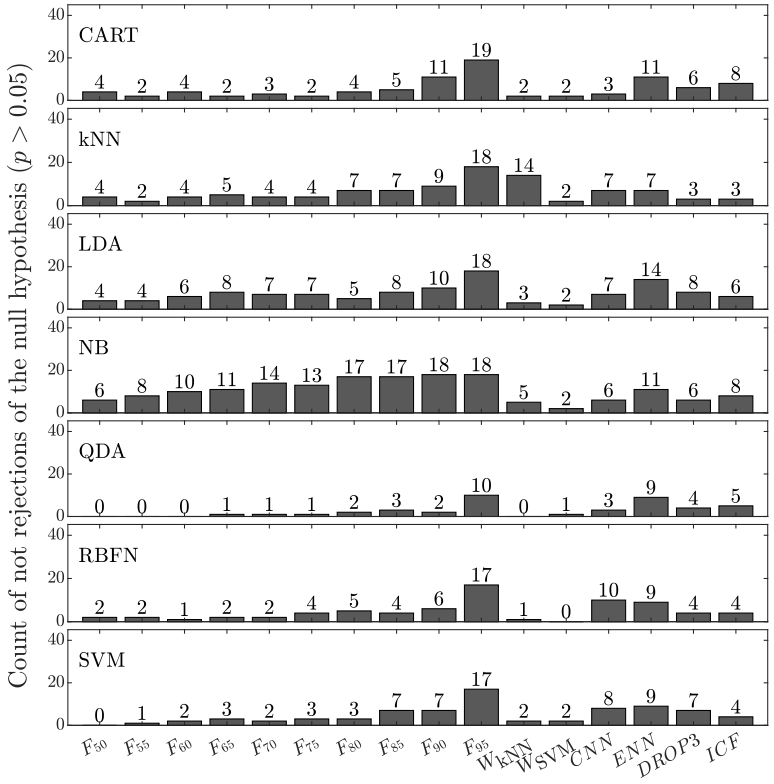


Fig. 5. Counts of no rejections of the null hypothesis of the McNemar test

Finally, the F_{65} method gave the best efficiency (E=0.85) due to the high accuracy, reduction rate, and highest PDF preservation. ENN had the worst efficiency (E=0.44) as it gave the lowest value of the reduction rate (RR=0.12), despite its high accuracy (ACC=0.77) and PDF preservation (HDC=0.96).

Fig. 7 shows a comparison between the X_O and X_S PDFs for each class and dimension of the Banana dataset. The three IS methods yielded the highest HDC values in this case study, namely F_{65}, W_{SVM}, and ENN.

The results show that the proposed approach produced a selected subset that correctly matched the probability distribution of X_O. In contrast, W_{SVM} and ENN gave a set of instances with slightly different distributions regarding X_O, even when these methods obtained a lower reduction rate than F_{65}.

7 Hyperparameters for the W_{SVM} Method

Table 4 shows the hyperparameters used by the W_{SVM} method.

8 Conclusions

This paper has presented an evolutionary IS method called F_w, which maximizes the PDF similarity between the original dataset and the selected subset and minimizes the number of selected instances. In this method, we consider the

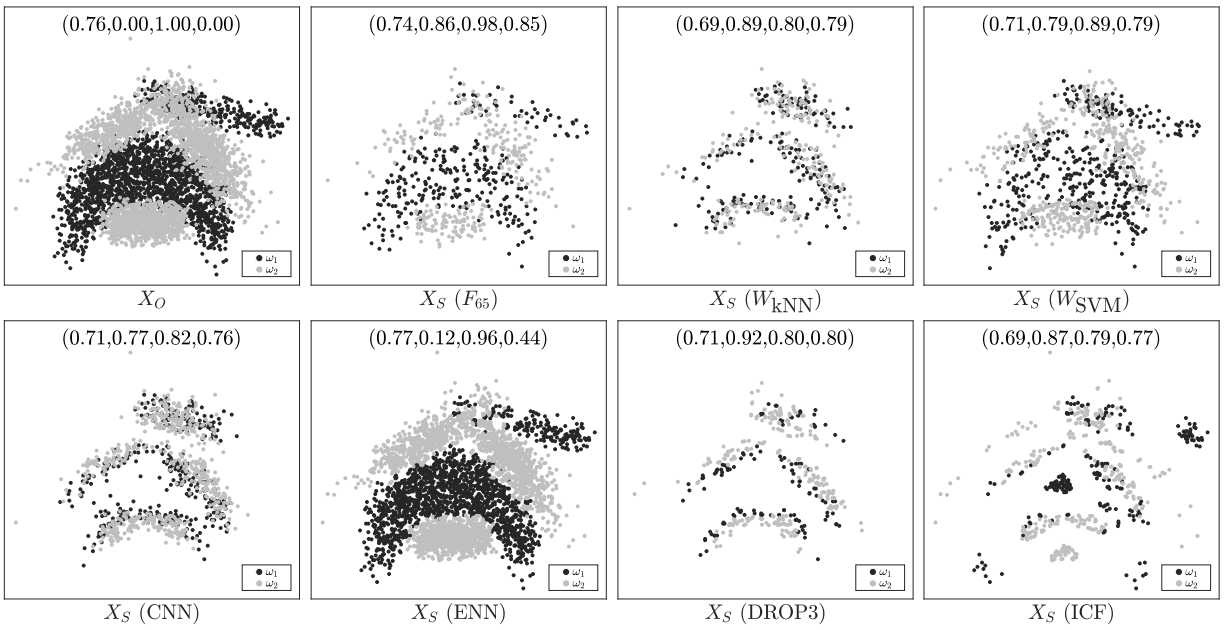


Fig. 6. Instance selection results on the Banana dataset. The average (ACC, RR, HDC, E) measures are shown in parenthesis inside each plot

IS task as an optimization problem, and aim to find subsets of instances that appropriately represent the original samples in the feature space.

We used the Hellinger distance to compare the similarity between two PDFs, since this is a measure of distributional divergence. Thus, we introduced a fitness function that calculates a weighted sum of the Hellinger distance between the original and selected subsets and the reduction rate of the selected subset. To examine the influence of the weight values on the performance, 10 different values were evaluated in the range $w = [0.50, 0.95]$ with steps of 0.05.

The results revealed that these two objectives conflict, i.e., the higher the weight value, the better the PDF preservation but the lower the reduction rate.

Unlike the EA wrapper methods in the literature, the proposed technique does not use a classifier to bias the search process towards samples that maximize the classification accuracy, but instead uses a PDF preservation approach as a novel heuristic. We evaluated the reusability of the obtained subsets to train different classifiers, comparing them against six IS methods: two EA wrappers (W_{kNN} and W_{SVM}) and four classic IS techniques (CNN, ENN, DROP3, and ICF). We also used four performance indices (the average accuracy, reduction rate, PDF preservation, and efficiency) to evaluate and compare the IS methods.

The results revealed that depending on the weight value, the proposed approach was able to outperform the alternative methods.

For instance, F_{95} attained the highest accuracy, while F_{50} obtained the best reduction rate. However, for $0.50 \leq w \leq 0.70$, F_w obtained the higher efficiency, and for all $w > 0.50$, the proposed approach outperformed all the EA wrappers and classical techniques in terms of PDF preservation.

It is worth mentioning that the results in Fig. 4 show that the proposed approach yields better

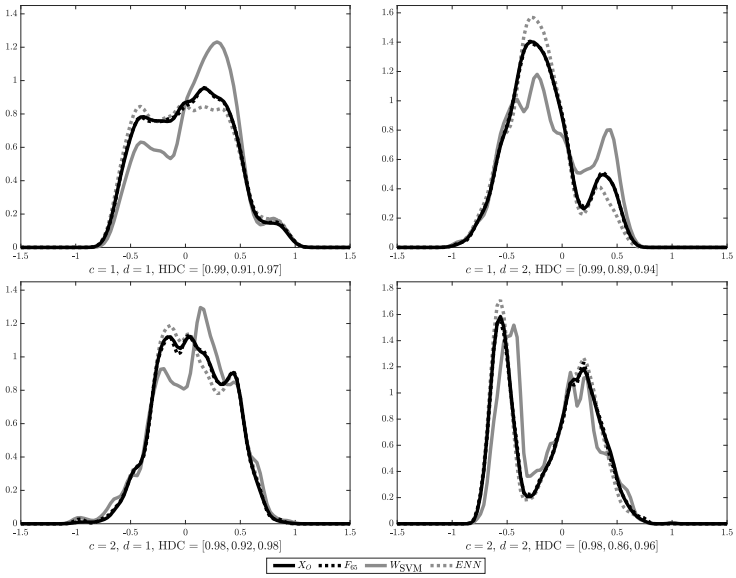


Fig. 7. KDE results by class and predictor variable on the Banana dataset for X_O and X_S subsets obtained with F_{65}, W_{SVM}, and ENN, respectively

generalization of the classification performance of different supervised learning algorithms than the EA wrappers; that is, F_w produces selected instances with good reusability capabilities in terms of training different classifiers.

The results of a McNemar test showed that the F_{95} method gave more subsets that did not have statistical differences from the original datasets than any alternative algorithm; this was due to the weight value in the fitness function ($w = 0.95$), which produced a high PDF preservation but poor performance in terms of the reduction rate.

The F_{50} method attained a regular efficiency on the medium-sized datasets due to its high PDF preservation and low reduction rate. Given the numbers of instances in these larger datasets, the proposed method may require more generations to explore the vast search space, which grows exponentially due to the original patterns.

It will therefore be necessary to investigate new representation schemes for evolutionary IS algorithms that do not explicitly encode all the original dataset instances and reduce the search space size for the IS problem.

Future work could focus on using multiobjective optimization to maximize the PDF preservation and minimize the reduction rate, so that non-dominated Pareto front solutions are obtained to make decisions and choices from among the different possible selected subsets.

Acknowledgments

Samuel Omar Tovias-Alanis thanks the National Council of Science and Technology (CONACyT, Mexico) for the doctoral scholarship.

References

1. **Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S. (2011).** Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. J. Mult-Valued Log. S., Vol. 17, pp. 255–287.
2. **Aldana-Bobadilla, E., Lopez-Arevalo, I., Molina-Villegas, A. (2017).** A novel data reduction method based on information theory and the eclectic genetic algorithm. Intell. Data Anal., Vol. 21, No. 4, pp. 803–826.

3. **Bishop, C. M. (2006).** Pattern Recognition and Machine Learning. Springer-Verlag, Berlin, Heidelberg.
4. **Brighton, H., Mellish, C. (2002).** Advances in instance selection for instance-based learning algorithms. Data Min. Knowl. Discov., Vol. 6, pp. 153–172.
5. **Cady, F. (2017).** The Data Science Handbook. Wiley.
6. **Cano, J. R., Herrera, F., Lozano, M. (2003).** Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study. IEEE Trans. Evol. Comput., Vol. 7, No. 6, pp. 561–575.
7. **Chang, C.-C., Lin, C.-J. (2011).** LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, Vol. 2, pp. 27:1–27:27.
8. **Cutler, A., Cordero, O. I. (1996).** Minimum Hellinger distance estimation for finite mixture models. J. Am. Stat. Assoc., Vol. 91, No. 436, pp. 1716–1723.
9. **Derrac, J., García, S., Herrera, F. (2010).** A survey on evolutionary instance selection and generation. Int. J. Appl. Metaheuristic Comput., Vol. 1, No. 1, pp. 60–92.
10. **Dua, D., Graff, C. (2017).** UCI machine learning repository.
11. **Duda, R. O., Hart, P. E., Stork, D. G. (2000).** Pattern Classification (2nd Edition). Wiley-Interscience, USA.
12. **García, S., Cano, J. R., Herrera, F. (2008).** A memetic algorithm for evolutionary prototype selection: A scaling up approach. Pattern Recogn., Vol. 41, No. 8, pp. 2693–2709.
13. **Garcia, S., Derrac, J., Cano, J., Herrera, F. (2012).** Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE Trans. Pattern Anal. Mach. Intell., Vol. 34, No. 3, pp. 417–435.
14. **Hart, P. (1968).** The condensed nearest neighbor rule. IEEE Trans. Inf. Theory, Vol. 14, No. 3, pp. 515–516.
15. **Kuncheva, L. I. (1995).** Editing for the k-nearest neighbors rule by a genetic algorithm. Pattern Recognit. Lett., Vol. 16, No. 8, pp. 809–814.
16. **MATLAB (2018).** 9.5.0.944444 (R2018b). The MathWorks Inc., Natick, Massachusetts.
17. **Reeves, C. R., Bush, D. R. (2001).** Using Genetic Algorithms for Training Data Selection in RBF Networks. Springer US, Boston, MA, pp. 339–356.
18. **Rosales-Perez, A., Garcia, S., Gonzalez, J. A., Coello, C. A. C., Herrera, F. (2017).** An evolutionary multiobjective model and instance selection for support vector machines with pareto-based ensembles. IEEE Trans. Evol. Comput., Vol. 21, No. 6, pp. 863–877.
19. **Wand, M. P. (1995).** Kernel smoothing. Chapman & Hall, London New York.
20. **Wilson, D. L. (1972).** Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. Syst. Man Cybern. Syst., Vol. SMC-2, No. 3, pp. 408–421.
21. **Wilson, D. R., Martinez, T. R. (2000).** Reduction techniques for instance-based learning algorithms. Mach. Learn., Vol. 38, No. 3, pp. 257–286.
22. **Zhou, Z.-H. (2012).** Ensemble methods: foundations and algorithms. CRC Press.

Article received on 30/09/2021; accepted on 16/12/2021.
Corresponding author is Samuel Omar Tovias-Alanis.

ISSN 2007-9737

Effect of Temporal Patterns on Task Cohesion in Global Software Development Teams

Alberto Castro-Hernández[1], Verónica Pérez-Rosas[2], Kathleen Swigger[3]

[1] Wayne State University Detroit,
Computer Science and Engineering,
United State

[2] University of Michigan,
Computer Science and Engineering,
United State

[3] University of North Texas,
Computer Science and Engineering,
United State

hg3246@wayne.edu, vrncapr@umich.edu, kathy@cse.unt.edu

Abstract. This work focuses on the analysis of temporal measures and their ability to predict *task cohesion* within global software development projects. Messages were collected from three software development projects that involved students from two different countries. The similarities and quantities of these interactions were computed and analyzed at individual and group levels. We proposed *pacing similarity*, *pacing rate* and *synchrony*, a set of temporal metrics measuring frequency and rhythm of team member's interactions. Results showed a statistically significant negative correlation between *pacing rate* and *task cohesion*, which suggests that frequent communications increases the cohesion between team members. The study also found a positive correlation between *synchrony* and *task cohesion*, which indicates the importance of establishing a communication rhythm within members a team. In addition, the temporal models at individual and group-levels were found to be good predictors of *task cohesion*, which indicates the existence of a strong effect of frequent and rhythmic communications on cohesion related to the task.

Keywords. Virtual groups, cross-culture communication, teamwork, task cohesion.

1 Introduction

Group cohesion is an important factor that affects collaboration behaviors among members of global software development teams. Previous research found that using communication technology often causes delays in the development of the group cohesion construct in virtual teams. As a result, global teams tend to have much lower group cohesion levels than co-located groups [10]. An important reason to examine group cohesion levels is that it seems to affect how a team deals with different obstacles during a project development. In addition, the relation between group cohesion and other constructs (e.g. trust) has been shown to have a significant relationship to team performance [6].

There have been a number of approaches used to measure group cohesiveness; ranging from paper-based individual surveys to the more automatic process of following a team's communication trails [4]. Such methods have shown the significance of examining factors that affect cohesion at the team process level. Despite they usefulness, an important drawback is that they

fail to capture temporal aspects of group processes and how they relate to task cohesion. A failure to utilize temporal information naturally reduces the power of the analysis, which may, in turn, limit the validity of a study's conclusions.

In an effort to gain a better understanding of group's temporal factors and their effect on task cohesion, we introduce three temporal measurements to predict a group's overall task cohesion levels. In particular, we focus on the analysis of features capturing temporal factors such as *pacing rate*, *pacing similarity*, and *synchrony* within medium-sized global software development teams and show how they can be used to predict cohesion levels within a group.

The goal of this paper, therefore, is to examine temporal features of global virtual teams and determine whether these measures relate to *task cohesion*. The measures are also used to create a learning model to predicts *task cohesion*. The main hypothesis of this paper is that the communication activities within distributed teams are cyclical in nature and oscillate between discussions and individual work.

2 Methodology

We began the study by asking students from institutions in the US and Mexico to work in teams on mid-sized software development projects. Once the projects were completed, we extracted temporal patterns from a team members' interactions. We then determined the relationship between temporal variables and *task cohesion*. A more detailed description of the teams' composition and the assigned projects now follows.

2.1 Teams

The data used in this study was obtained from three student global software development projects that occurred between 2014 and 2015. Students who participated in these projects were enrolled in Computer Science courses at the University of North Texas and Universidad Politécnica de Altamira –located in the United States of America and Mexico, respectively.

While each of the three projects addressed aspects related to the software development process, the actual assignments often varied in terms of team size (4-8 members), and specific task.

Two of the projects consisted of redesigning a non-profit website, including the redesign of the home page, the events page, and the contribution page sections as well as implementing a database that could support the various operations that were needed to maintain the different pages.

A third project consisted of creating a learning website about an optimization algorithm. The various elements of the website included a section where users could read information about the algorithm, as well as a section where users could test the algorithm. The length of the project also varied between 6-7 weeks. The software development methodology was defined by each team.

2.2 Software

Students who participated in the three projects were asked to communicate with one another using Redmine, a project management web application. This application platform included several collaborative tools such as chat, forums, wikis, and document sharing. Moreover, the application software was enhanced so that it recorded and time stamped all interactions among group members and transferred this information to a centralized database.

2.3 Measures

In order to analyze interactions among team members, we developed three temporal measures (*Pacing similarity*, *Pacing rate*, and *Synchrony*), which we believed to be good predictors of a team's task cohesion levels within a global software development context.

These measures were calculated at the individual-level as described below. Additionally, we calculated these measures at the group-level by averaging their values i.e. *group pacing rate*, and *group synchrony*. We also measured *task cohesion* at the individual and group level.

2.3.1 Pacing Rate and Pacing Similarity

The *pacing rate* (pr) was defined as the average number of seconds between messages from participants in the same team. *Pacing similarity* was obtained by averaging the similarity between the *pacing rate* of team participant i and the *pacing rate* of each of the other team members (see Equation 1):

$$pacing\_similarity = 1 - \frac{|pr_i - pr_j|}{pr_i + pr_j}. \qquad (1)$$

2.3.2 Synchrony

This temporal measure captures the synchrony of messages sent between two team participants; where the messages from each participant are defined as a time series, and then their frequencies are compared to one another, one at the time [7].

Figure 1 (top) shows an example of the obtained time series and the number of messages sent by a team consisting of 4 participants.

We estimated the spectral density from each data series by creating a periodogram, a diagram of frequencies by amplitude, using the Fast Fourier transform.

Figure 1 (bottom) shows the resulting periodogram for the time series in our previous example. From this information, we calculated the coherence between two time series, which was the correlation between paired members of each frequency.

As a result we got a coherence series for each combination of pairs within a team ($c(2, n)$), see Figure 2. This calculation results in a vector of coherence values (each value representing coherence in a specific frequency) between two subjects.

To provide a unique temporal value between two individuals, we took the highest coherence score. Finally, for each participant in a team, we obtained the highest coherence scores of a participant as compared to the rest of the team members and averaged those scores.

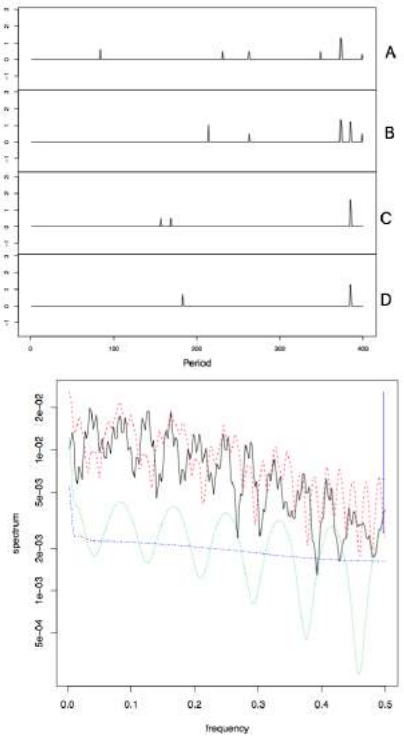


Fig. 1. Frequency graph (top) and periodogram (bottom) of 4 team members

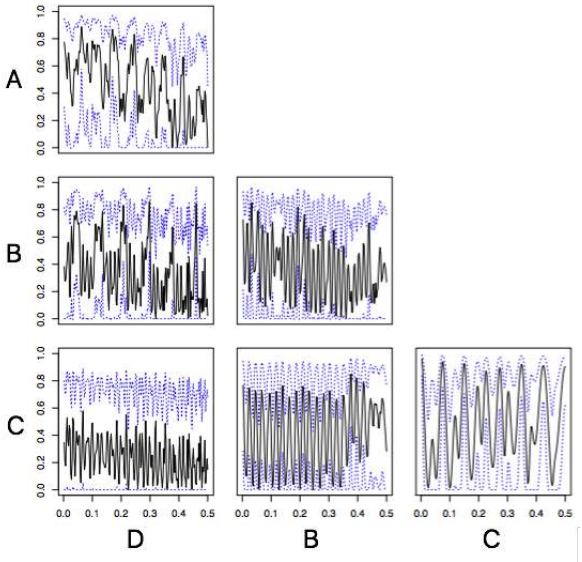


Fig. 2. Coherence graphs for each pair team members

2.3.3 Task Cohesion

Task cohesion is a construct that measures the degree to which team members are working together. To calculate the value of this construct, we used an individual survey approach because it seemed more appropriate for a distributed virtual team context [8]. The survey, intended to capture a participant perception of their team, included questions from the *task cohesion* section of the Group Environment Questionnaire (GEQ) [3]. A group's cohesion measure was then obtained by averaging each team's individual responses to the survey.

2.4 Linear Model

We conducted a set of experiments to assess the performance of temporal features as predictors of task cohesion. The data, collected from the three global projects, was then used to determine which set of variables were most successful at predicting individual task cohesion. We examined the relationship between *task cohesion* and the three temporal measures of *synchrony*, *pacing similarity*, and *pacing rate*. We also conducted comparative experiments between *pacing similarity* and *pacing rate* to choose the best pacing representation for the linear model.

3 Results

3.1 Sample Characteristics

A total of 5,583 messages were transmitted during the three projects. A total of 167, out of a possible 180, task cohesion surveys were collected. Since we had 27 missing surveys, we decided to include only messages sent by students who had completed the questionnaire; thus, we had a total of 5,446 messages in the final dataset.

.

Table 1. Task cohesion values by culture *$p < 0.05$

Country	Mean	India	US
India	8.01		
US	6.21	1.8057*	
Mexico	6.48	1.5310*	0.2746

Table 2. Temporal metric statistics

Project	Instances used	Avg. Pacing Similarity	Avg. Pacing Rate
A	94.52%	0.56	4d 02h 52m
B	82.97%	0.54	4d 01h 09m
C	88.33%	0.53	6d 03h 38m

3.2 The Culture Effect

Due to the multicultural composition of our teams, we obtained cohesion surveys from people who were born in eleven different countries. However, the majority of the completed surveys came from students born in India, the US, and Mexico (i.e $n > 20$), while only a few surveys came from students born in other countries (i.e. $n < 4$). As a result, we reduced the data set even further by only keeping responses from students born in India, the US, and Mexico; thus, ending up with a final count of 4,849 messages sent by 153 individuals. We then compared the task cohesion mean values between countries and found that students from India tended to have higher Task cohesion perceptions than either US or Mexican students (see Table 1). Considering these findings, we used the culture factor as a control variable in our analyses.

3.3 Pacing at Project-Level

In addition, analyses conducted on the pacing metric measured at project level excluded data points from students who posted only one message during their interactions as the pacing similarity measure cannot be calculated for single messages. The final percentage of messages used for each project is shown in Table 2.

Results obtained in the remaining subjects suggest that pacing similarity has a similar value across all three projects; however, a comparison of the pacing rate shows that the communication rhythms within the three projects were dissimilar. More specifically, the rate of student replies was much slower in the last project i.e., Project C.

3.4 Temporal Measures and Task Cohesion

Using the final dataset, we calculated the correlation between *pacing similarity* and *task cohesion* and between *pacing rate* and *task cohesion* at the individual-level. Table 3 shows that *pacing similarity* has no effect on task cohesion. On the other hand, *pacing rate* has a weak, but statistically significant effect on *task cohesion*. This correlation, albeit weak, suggests that frequent communication tends to lead to an increase of individual's perception of Task cohesion.

The lack of a relation between *pacing similarity* and *task cohesion* suggests that individuals prefer frequent, although erratic, communication (*pacing rate*) with team members, over a more uniform, but less frequent, rate of communication.

Moreover, the *pacing rate* metric consists of 2 components: 1) Duration of communication (time between first and last communications), and 2) Number of communications. As a result, we analyzed the relation of these two components to *task cohesion*. Table 4 shows that both components have a statistically significant relation to *task cohesion*. The relation of the number and duration of communications to *task cohesion* may represent the communication engagement by each participant. Therefore, participants who are engaged in the project will perceive a greater team cohesion than those who are not.

Results for the *Synchrony* metric, presented in Table 5, suggest that *synchrony* is a very good predictor of *task cohesion*. Hence, this metric seems to capture the synchrony of collaborations among team members, regardless of the frequency of when those communications occur (e.g. communications every 12 hours, versus communications every 24 hours, etc.).

We also created an additional model to predict *task cohesion* (coh) that use the *Pacing rate* (pr) and *synchrony* (sy) (while controlled by *team size* (ts) and *culture* (cu). The top row in Table 6 shows that the temporal model is a good predictor of the variability of the *task cohesion* variable (r=0.358). These results suggest that temporal-based measures at the individual-level are helpful for understanding the perceptions of *task cohesion*.

Finally, we also evaluated the predictive power of temporal metrics measures at the group-level and built a model using *team size* (ts), *group pacing rate* (gpr), *group synchrony* (gsy) and *group task cohesion* (gcoh). Results shown in the last row of Table 6 indicate that group-level temporal measures are also good predictors of *task cohesion*.

Table 3. Pacing correlation with Task Cohesion *$p < 0.1$

Project	Correlation
Pacing Similarity	0.049
Pacing Rate	-0.129*

Table 4. Relation of components of Pacing rate to Task cohesion *$p < 0.1$, **$p < 0.05$

Component	Task cohesion
Duration of communication	0.137*
Number of communications	0.161**

4 Conclusion

The major goal of this study was to determine which temporal features can predict *task cohesion* for individuals and teams who were engaged in global software development projects. The motivation for this work was the expectation that information about the cohesion levels among individuals and teams in global software development projects could lead to better interactions among team members, and ultimately better group performance. Although many researchers have explored relationships between certain collaboration variables [11, 2, 9], few have examined the effects of temporal factors of pacing and synchrony.

Table 5. Effect of coherence similarity to *task Cohesion* $^{**}p < 0.05$

Measure	Correlation
Coherence Similarity	0.170**

Table 6. Linear temporal models to predict *task cohesion* $^{***}p < 0.01$

Model	R
$a_0 + a_1 \cdot ts + a_2 \cdot cu + a_3 \cdot pr + a_4 \cdot sy = coh$	0.358**
$b_0 + b_1 \cdot ts + b_2 \cdot gpr + b_3 \cdot gsy = gcoh$	0.733**

For example, previous research has indicated that linguistic similarity, information exchange, and message content can help determine cohesion levels within groups [5, 12, 1]. However, we felt that just-in-time and abrupt communications require new methodologies to measure temporal phenomena that are more complex than simple linear patterns. For example, current research has not captured the effects of communication cycles spiraling up, down or intensifying.

Thus, this work herein proposes a methodology for describing the temporal narratives of distributed team processes over time and determining whether they could be used to predict Task cohesion. The three measures that were developed for this study were *pacing similarity*, *pacing rate*, and *synchrony*. Each of these factors was examined at both the individual and group levels.

Results showed a statistically significant negative correlation between the *pacing rate* and *task cohesion*, which suggests that frequent and perhaps sporadic rhythmic communications about different social and work themes increases the cohesion among team members. On the other hand, *pacing similarity* was not found to be related to *task cohesion*, which suggests that a minimum (although not equal) participation is necessary for individual's to perceive that their group is cohesive.

Thus, *pacing similarity* is not relevant to Task cohesion. We also found a positive correlation between *synchrony* and *task cohesion*, which suggests the importance of establishing a rhythm within a team. Finally, the temporal models constructed at individual and group-levels were found to be good predictors of *task cohesion*, which indicates the existence of a strong effect of frequent and rhythmic communications on cohesion related to the task.

Knowledge obtained from this study should provide insight into current empirical research on global virtual teams by defining the different temporal patterns that occur in these projects and how this can affect a team's perception of their cohesion levels.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0705638. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

References

1. **Brooks, I., Swigger, K. (2012).** Using sentiment analysis to measure the effects of leaders in global software development. Collaboration Technologies and Systems (CTS), 2012 International Conference on, pp. 517–524.
2. **Brooks, I., Swigger, K. (2013).** Leadership's effect on overall temporal patterns of global virtual teams. Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on, IEEE, pp. 371–379.
3. **Carless, S. A., Paola, C. D. (2000).** The Measurement of Cohesion in Work Teams. Small Group Research, Vol. 31, No. 1, pp. 71–88.
4. **Castro-Hernandez, A., Swigger, K., Ponce-Flores, M. P. (2014).** Effects of cohesion-based feedback on the collaborations in global software development teams. Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on, IEEE, pp. 74–83.

5. **Gonzales, A. L., Hancock, J. T., Pennebaker, J. W. (2010).** Language Style Matching as a Predictor of Social Dynamics in Small Groups. Communication Research, Vol. 37, No. 1, pp. 3–19.
6. **Goodman, P., Ravlin, E., Schminke, M. (1987).** Understanding Groups in Organizations. Tepper School of Business.
7. **Gottman, J. M. (1981).** Time-series analysis: A comprehensive introduction for social scientists, volume 400. Cambridge University Press.
8. **Hardin, A. M., Fuller, M. A., Valacich, J. S. (2006).** Measuring Group Efficacy in Virtual Teams New Questions in an Old Debate. Small Group Research, Vol. 37, No. 1, pp. 65–85.
9. **Niederhoffer, K. G., Pennebaker, J. W. (2002).** Linguistic style matching in social interaction. Journal of Language and Social Psychology, Vol. 21, No. 4, pp. 337–360.
10. **Powell, A., Piccoli, G., Ives, B. (2004).** Virtual teams: a review of current literature and directions for future research. ACM Sigmis Database, Vol. 35, No. 1, pp. 6–36.
11. **Swigger, K., Nur Aplaslan, F., Lopez, V., Brazile, R., Dafoulas, G., Serce, F. C. (2009).** Structural Factors That Affect Global Software Development Learning Team Performance. Proceedings of the Special Interest Group on Management Information System's 47th Annual Conference on Computer Personnel Research, SIGMIS CPR '09, ACM, New York, NY, USA, pp. 187–196.
12. **Tausczik, Y. R. (2012).** Changing group dynamics through computerized language feedback.

Article received on 20/06/2021; accepted on 20/09/2021.
Corresponding author is Alberto Castro-Hernández.

ISSN 2007-9737

Emotional Similarity Word Embedding Model for Sentiment Analysis

Kazuyuki Matsumoto[1], Takumi Matsunaga[2], Minoru Yoshida[1], Kenji Kita[1]

[1] Tokushima University,
Graduate School of Technology, Industrial and Social Sciences,
Japan

[2] Tokushima University,
Graduate Schools of Science and Technology
for Innovation Division of Science and Technology,
Japan

{Matumoto, mino, kita} @is.tokushima-u.ac.jp, c612135014@tokushima-u.ac.jp

Abstract. We propose a method for constructing a dictionary of emotional expressions, which is an indispensable language resource for sentiment analysis in the Japanese. Furthermore, we propose a method for constructing a language model that reproduces emotional similarity between words, which to date has yet not been considered in conventional dictionaries and language models. In the proposed method, we pre-trained sentiment labels for the distributed representations of words. An intermediate feature vector was obtained from the pre-trained model. By learning an additional semantic label on this feature vector, we can construct an emotional semantic language model that embeds both emotion and semantics. To confirm the effectiveness of the proposed method, we conducted a simple experiment to retrieve similar emotional words using the constructed model. The results of this experiment showed that the proposed method can retrieve similar emotional words with higher accuracy than the conventional word-embedding model.

Keywords. Emotion recognition, emotional similarity, neural networks.

1 Introduction

In recent years, the distributed representation of words and sentences has been frequently used as an artificial intelligence technique to analyze data on social networking sites. Distributed representation has made it easier to calculate relevance and similarity, and to use them as features in machine learning by quantifying the features of words and sentences in the form of vectors, which were handled symbolically. However, one problem with a distributed representation of words and sentences is that it can handle semantic information, but does not deal with emotional information effectively. For example, suppose that there are two types of expressions that express emotions in a certain situation: positive and negative words. These expressions are often used in similar contexts, even if the emotions are the opposite. Many distributed expressions are based on a large corpus and are intended to extract semantic information from context, etc. If they are used as is, they are considered incapable of expressing emotions correctly, as aforementioned.

In emotion recognition, these problems do not have a significant impact because supervised machine learning is performed using distributed expressions as features. However, when generating paraphrased sentences, based on word variants, it is necessary to have a mechanism to suppress the replacement of words with those that are semantically similar but are of the opposite meaning. In addition, emotions are often analyzed not only based on polarity, such as positive/negative, but also based on basic emotions expressed in a circle of emotions, as proposed by psychologist Plutchik [1]. Ptaszynski et al. [2] conducted an emotion analysis based on

a dictionary of emotional expressions. In addition, Sano [3] created a systematic dictionary of words expressing emotions in the form of a dictionary of appraisal expressions.

The aforementioned generalized dictionary of emotional expressions, is useful not only for the emotional analysis of linguistic information, but also for facilitating communication between people. Emotional Quotient (EQ) is a measure of the intelligence required to express one's own emotions appropriately and to understand the emotions of others [4].

However, generalized dictionaries do not include unknown expressions, such as new words and popular phrases, and they have problems corresponding to the changes in language usage over time.

In this study, we focus on the strengths of unsupervised language distributed representation learning: the ability to specialize a model to a specific domain by training based on a corpus, and the ability to update the training data easily. Specifically, we convert a generalized sentiment dictionary into a numerical vector using distributed representations and pre-trained linguistic distributed representations that are specific to emotions.

The distributed representation obtained by this method may lose semantic information because it is specific to emotions. Therefore, a model based on a semantic dictionary, such as a thesaurus, is used to acquire distributed representations that contain semantic information.

This method aims to facilitate the construction of emotionally distributed representations, specialized for sentiment analysis of language as well as semantic information. To evaluate the effectiveness of the constructed model, we compared it with the conventional model of language distributed representation.

2 Related Works

The WordNet-Affect [5] and the Japanese Evaluative Polarity Dictionary (Kobayashi et al.) [6] are examples of the linguistic systematization of words expressing emotions. WordNet-Affect has a thesaurus of words expressing emotions; however, there is no official Japanese version.

Although some of them are translated from English to Japanese, there are many expressions that are not suitable for direct translation.

Therefore, an emotional thesaurus, specialized for the Japanese language, is needed.

The Dictionary of Emotional Expressions (Akira Nakamura) [7] is a collection of emotional expressions, in written text, from 806 works by 197 modern and contemporary Japanese authors. The dictionary defines ten types of emotion (joy, anger, sorrow, fear, shame, like, hate, excitement, relief, surprise) and compound emotions.

This dictionary contains a relatively comprehensive summary of emotional expressions used within the Japanese language.

The dictionary of appraisal expressions constructed by Sano [3], is a classification of expressions that describe values. It is unique in that it defines perspectives other than the criteria of emotion polarity, that is, positive and negative, for evaluation classification and emotion analysis. However, the dictionary does not clearly define the types of emotions; therefore, it is necessary to associate emotion classes with attributes to employ conventional emotion analysis methods.

Emo2Vec, proposed by Wang et al. [8], is based on two different models (i.e., local and global) for adding sentiment information to word vectors to analyze opinions from review sentences. This method is based on Plutchik's circle of emotions and uses multi-task learning to achieve higher expressive power than the existing emotion polarity. Their work improves on existing word and emotion embeddings adopted in experiments on the Chinese and English languages.

The difference between our method and their approach is that the emotion space is given as an 8-dimensional vector. Our study defines a 25-dimensional vector as the basic axis so that the types of emotions can be handled as flexibly as possible. To handle multiple emotions, the sigmoid function is used as the output layer to predict a 25-dimensional vector.

This allows ambiguity in phrases that correspond to multiple emotions. In addition, semantic features are pre-trained separately from word embeddings to enhance semantic expressiveness.

Table 1. Emotion class by Fischer

Large class	Sub Class	Code	Class name	Example emotions
A	Joy	A-1-1	Relief	looseness, peaceful, relief, solace, etc.
		A-1-2	Impression	ecstasy, delight, etc.
		A-1-3	Hope	optimistic, expectation, tympany, enthusiasm, etc.
		A-1-4	Proud	victory, boasting, adversarial quality, etc.
		A-1-5	Pleasure	contentment, feel good, briskness, etc.
		A-1-6	Excitement	ardency, alacrity, interest, gaiety, etc.
		A-1-7	Joy	ravishment, airiness, cheerfulness, etc.
	Love	A-2-1	Respect	adoration, envy, beautiful, etc.
		A-2-2	Passion	desire, intoxication, adhesion, etc.
		A-2-3	Like	love, attraction, charity, chummy, etc.
B	Surprise	B-1-1	Surprise	amazement, strangeness, etc.
C	Anger	C-1-1	Bitter	anguish, difficult, etc.
		C-1-2	Envy	jealousy, etc.
		C-1-3	Contempt	boke, sicken, etc.
		C-1-4	Rage	umbrage, fume, etc.
		C-1-5	Scandalize	frustration, shocked, etc.
		C-1-6	Displeasure	disconcertedness, accusation, etc.
D	Sorrow	D-1-1	Pity	commiseration, sympathy, empathy, etc.
		D-1-2	Alienation	isolation, loneliness, nostalgia, dejection, etc.
		D-1-3	Guilt	shame, regret, confession, abjection, etc.
		D-1-4	Disappointment	drug, fatigue, hit or miss, rejection, etc.
		D-1-5	Sorrow	despair, unluckness, dysphoria, etc.
		D-1-6	Cruel	smart, agonal, etc.
	Fear	D-2-1	Anxiety	nervousness, worry, suspense, heartache, fear, etc.
		D-2-2	Warning	consternation, abasement, fret, etc.
E	Neutral	E-1-1	Neutral	neutral

3 Learning of Emotional Embedding

3.1 Emotion Class

In this study, we classified several existing dictionaries, such as the Emotional Expressions Dictionary, the Appraisal Expressions Dictionary, and the Idiom Expressions Dictionary, according to the phylogeny of emotions proposed by Fischer [9], as shown in Table 1. Based on this classification, the distributed representation of each expression was learned to extract features specific to the emotion. As the expressions registered in the dictionary of emotional expressions do not include neutral expressions, 25 classes (excluding E) are actually used for emotional classification.

3.2 Embedding of Word / Phrase

Traditionally, word2vec [10], fastText [11], GloVe [12], and other methods based on CBOW or skip-gram have been used for word embedding. Recently, bidirectional encoder representation from transformers (BERT) [13] has been used; this approach enables unsupervised training by considering the position of word occurrences using trans-formers and attention mechanisms. In BERT, generic unsupervised task-solving models called

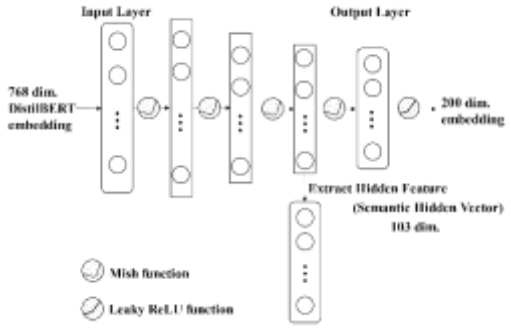


Fig. 1. Structure of neural networks extracting hidden semantic embedding

next sentence prediction and masked language models are pre-trained on a large unlabeled corpus.

Using the parameters obtained from the trained network, distributed representations of the language that can be applied to various tasks, are extracted. Based on the distributed representations, transfer learning or fine tuning was conducted for other tasks.

The disadvantage of BERT is that it takes a long time to train, and the trained model is large owing to the size of the network and the large number of parameters. For this reason, research has been conducted on reducing the network parameters of BERT as well as on models such as ALBERT [14] or DistilBERT (Distilled BERT) [15], which succeeded in reducing the size of the model without degrading the performance of BERT.

In this study, we target not only word units but also phrases consisting of multiple words, such as idiomatic expressions. Therefore, it is desirable to use a method that can obtain distributed expressions not only for words but also for phrases and sentences.

In our proposed method, it is necessary to convert the features obtained from words and phrases into other features that can express emotions and semantics. Therefore, we need to obtain the distributed representation to be inputted as flexibly and efficiently as possible.

Therefore, we decided to use DistilBERT, which is a lightweight model with a reduced number of parameters, as the initial embedding.

3.3 Sense Vector Based on Wikipedia Entity Vector

The biggest advantage of manually constructed semantic dictionaries is that they contain almost no noise, which may affect accuracy. In general, words in a manually constructed semantic dictionary belong to semantic categories, and these categories are defined by superordinate and subordinate categories.

For example, it is possible to determine the semantic distance and similarity between words, using electronic dictionaries such as WordNet [16]. However, as there are many ambiguities in language usage, it is practically impossible to construct a complete semantic dictionary that covers all the uses of a word in reality.

There have been several studies on the automatic creation of semantic dictionaries based on Wikipedia [17]. While their method can define the semantic concepts of a large number of words, they sometimes register incorrect information in the dictionary or show bias toward certain fields. However, we can extract conceptual information with high accuracy using the rich vocabulary of Wikipedia and the sophisticated information of the articles.

In this study, we consider training a model that uses the Japanese Wikipedia Entity Vector as its prediction target, which is a distributed representation of the vocabulary headings used in Wikipedia and other vocabulary used in the article, to obtain the middle-layer vector.

We used DistilBERT embeddings as inputs. Because the distributed representation vector to be predicted also contains negative values, we use Mish [18] as the activation function that can retain negative values. Figure 1 illustrates the structure of the semantic vector extraction network model. Using this model of semantic vectors together with the model of emotional vectors that will be described later, it is possible to consider both semantics and emotion. The hidden feature vector is called a semantic hidden vector (S-HV). The S-HV is a vector with 103 dimensions.

The formula for calculating Mish is shown in Equation (1), $\ln$ is the natural logarithm, and e^x is the exponential function:

$$f(x) = x \cdot \tanh\left(\ln\left(1 + e^x\right)\right). \quad (1)$$

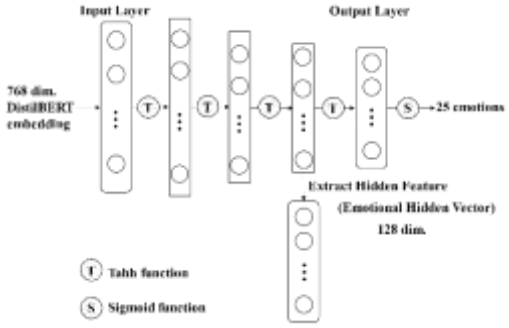


Fig. 2. Neural networks for learning with emotional embedding

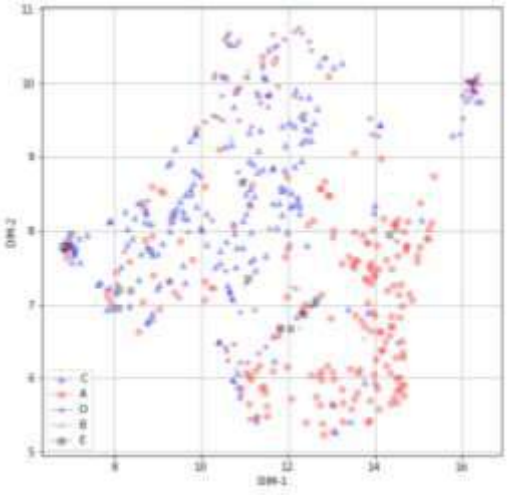


Fig. 3. Visualization of emotional embedding (using Neural Autoencoder and UMAP)

3.4 Learning Method of Emotional Embedding

To achieve embedding learning of word emotion, a model that predicts the emotion of words and phrases is required. We transformed words and phrases with emotion labels into distributed representations and then train a model to predict the emotion labels using the DistilBERT pre-training model described in Section 3.2.

The model was constructed based on a neural network. The architecture of this neural network has multiple hidden layers, and the hidden layer before the final output layer is designed to be a fully connected layer with more neurons than the number of dimensions of the emotion to be predicted.

Figure 2 shows the structure of the neural network used in this study. The hidden feature vector is called the emotional hidden vector (E-HV). The E-HV is a vector of 128 dimensions.

The results of emotional embedding, compressed using autoencoder based on neural networks, are converted to two dimensions by UMAP[19] and visualized in Fig. 3. From this figure, we can see that E (neutral) and B (surprise) are distributed in several clusters, and A (joy), C (Anger), and D (Sorrow) partially form their own clusters.

From this, it can be expected that B, which has a small number of cases, and E, which has few features, is relatively difficult to classify.

4 Experiment

4.1 Experimental Setup

We evaluated the validity of the emotional or semantic distributed representations of words, obtained by the proposed approach, using the following two methods:

Eval-1. We used emotion expressions with emotion labels that were not used for training as input, and calculated the similarity to the emotion expressions in the training data based on the emotion and semantic embedding of the words. From the results of this calculation, we predicted the emotion label based on the k-nearest neighbor method and obtained the correct answer rate. Experiments were conducted for the cases of k=10 and 20.

Eval-2. For a corpus of sentences with emotional labels, we obtained distributed representations of emotion and semantics using the trained models, and then trained sentiment prediction models using machine learning algorithms. The emotion classification model was evaluated using a cross-validation method.

The training and evaluation data for the dictionary used in Eval-1 are listed in Table 2. The

Table 2. Training data and evaluation data.

Train Total: 12,180 words							
A	A-1-1 495	A-1-2 140	A-1-3 182	A-1-4 851	A-1-5 462	A-1-6 240	A-1-7 818
	A-2-1 1,543	A-2-2 536	A-2-3 827				
B	B-1-1 784						
C	C-1-1 161	C-1-2 47	C-1-3 2,843	C-1-4 1,602	C-1-5 118	C-1-6 758	
D	D-1-1 56	D-1-2 544	D-1-3 124	D-1-4 445	D-1-5 677	D-1-6 282	
	D-2-1 590	D-2-2 292					
E	E-1-1 465						
Test Total: 693 words							
A	A-1-1 0	A-1-2 16	A-1-3 96	A-1-4 0	A-1-5 0	A-1-6 40	A-1-7 92
	A-2-1 136	A-2-2 0	A-2-3 68				
B	B-1-1 0						
C	C-1-1 41	C-1-2 0	C-1-3 0	C-1-4 16	C-1-5 0	C-1-6 96	
D	D-1-1 20	D-1-2 0	D-1-3 24	D-1-4 0	D-1-5 32	D-1-6 0	
	D-2-1 40	D-2-2 0					
E	E-1-1 0						

Eval-2 gradient boosting algorithm was used. LightGBM was used as the library.

For the evaluation corpus, we used Japanese sentences from the Japanese-English bilingual sentiment corpus (J-Corpus) [19, 20], and tweets and blogs with emotion tags (Web-Corpus). The tags assigned to J-Corpus were used after converting them into major categories A, B, C, D, and E.

The breakdown of the data is presented in Table 3. Table 4 shows the breakdown of words in the corpus by emotion type for both the Web-Corpus and J-Corpus, and Table 5 shows the number of words by part of speech.

We used the training data for emotion words (Train: 12,180 words) to count emotion words by emotion category. Some words, sentences, and phrases are given more than one emotion tag, because the interpretation may differ slightly from one dictionary to another.

The combinations of features to be compared are presented in Table 6. The "v" in the cells of the table indicates that the feature is used, and the "-" indicates that it is not used. To combine multiple features, each feature vector was connected horizontally.

4.2 Evaluation Method

In Eval-1, recall, precision, and F1-score were calculated and evaluated for each level of granularity in the hierarchy of emotion categories (1, 2, and 3 levels). The values of k were 10, 20, and 0.7, 0.5, and 0.3 were used for the similarity threshold.

Table 3. Evaluation corpora

Sentence Emotion Class (Large Class)	J-Corpus		Web-Corpus	
	Sentences	Words	Sentences	Words
A (Joy)	212	2,426	30,777	436,783
B (Surprise)	22	306	1,592	19,973
C (Anger)	238	2,770	15,018	252,922
D (Sorrow)	129	1,587	21,902	265,259
E (Neutral)	589	7,106	7,232	78,513
Total	1,190	14,195	76,521	1,053,450

Table 4. Number of emotion words for each emotion class

Sentence Emotion Class	J-Corpus Number of emotion words for each emotion class					Web-Corpus Number of emotion words for each emotion class				
	A	B	C	D	E	A	B	C	D	E
A	167	17	7	20	1	23,698	1,193	4,326	1,628	1,047
B	7	7	12	8	1	512	260	212	134	288
C	69	13	152	38	1	6,337	598	6,036	2,043	427
D	35	13	63	77	3	7,310	749	4,574	3,546	586
E	197	46	253	141	18	2,526	154	670	209	127
Total	475	96	487	284	24	40,383	2,954	15,818	7,560	2,475

Table 5. Number of words for each part of speech

Sentence Emotion Class	J-Corpus Number of words for each POS			Web-Corpus Number of words for each POS		
	Noun	Adjective	Verb	Noun	Adjective	Verb
A	785	79	281	149,617	15,969	55,504
B	92	3	56	6,826	497	2,606
C	874	70	340	82,468	6,222	35,874
D	495	36	203	85,605	8,974	37,565
E	2,169	151	1,052	28,646	1,482	11,353
Total	4,415	339	1,932	353,162	33,144	142,902

Table 6. Combination of features

Combination ID	Combination Type	E-HV	S-HV	DBERT
1	ehv	v	-	-
2	shv	-	v	-
3	dv	-	-	v
4	ehv+shv	v	v	-
5	ehv+shv+dv	v	v	v

In Eval-2, Recall, Precision, and F1-score were calculated for four major categories, A, B, C, and D, excluding neutral "E." 5-fold cross-validation was used to deal with class imbalance, and the Synthetic Minority Over-sampling Technique) [22], Edited Nearest Neighbor (ENN) [23], SMOTE-ENN [24], and SMOTE-Tomek Links [25] were used as resampling methods. For oversampling and undersampling, we used the class module in library imbalanced learning[1].

[1] https://imbalanced-learn.org/stable/

Table 7. Experimental Result of Eval-1

Category	Comb. Type	threshold	k	Accuracy
Large Class	ehv+shv+dv	0.3	10	0.595
			20	0.580
		0.5	10	0.595
			20	0.584
	ehv+shv	0.7	10	**0.600**
	ehv		20	0.590
Sub Class	ehv+shv+dv	0.3	10	0.392
	ehv		20	**0.411**
	ehv+shv+dv	0.5	10	0.392
	ehv		20	**0.411**
	ehv+shv	0.7	10	0.397
	ehv		20	0.405

Table 8. Precision, Recall, and F1-score for each emotion class (J-Corpus)

J-Corpus Result		A			B			C			D		
Resampling	**Comb.ID**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**
SMOTE	ehv	0.76	0.76	0.76	**0.38**	0.27	**0.32**	0.74	0.77	0.76	0.53	0.51	0.52
	shv	0.56	0.55	0.56	0.36	0.18	0.24	0.58	0.63	0.6	0.33	0.32	0.33
	dv	0.73	0.73	0.73	0.15	0.09	0.11	0.68	0.77	0.72	0.53	0.43	0.48
	ehv+shv	**0.77**	0.76	0.76	0.33	0.27	0.3	0.75	0.82	**0.79**	0.6	**0.53**	**0.56**
	ehv+shv+dv	0.76	0.76	0.76	0.24	0.23	0.23	0.74	0.81	0.77	0.6	0.5	0.55
ENN	ehv	0.69	0.8	0.74	0.16	0.23	0.19	0.68	0.82	0.74	0.6	0.19	0.28
	shv	0.41	0.19	0.26	0.08	0.41	0.14	0.43	0.72	0.54	0	0	0
	dv	0.53	0.52	0.53	0.12	0.27	0.16	0.51	0.7	0.59	0.54	0.05	0.1
	ehv+shv	0.76	0.82	**0.79**	0.22	0.27	0.24	0.67	**0.86**	0.76	0.67	0.22	0.33
	ehv+shv+dv	0.71	0.79	0.75	0.07	0.09	0.08	0.66	**0.86**	0.75	**0.71**	0.16	0.25
SMOTE-ENN	ehv	0.62	0.86	0.72	0.16	0.32	0.21	0.83	0.5	0.62	0.5	0.47	0.48
	shv	0.4	0.9	0.55	0.11	0.36	0.16	0.58	0.03	0.06	0.19	0.05	0.08
	dv	0.45	**0.92**	0.6	0.21	**0.55**	0.3	0.77	0.04	0.08	0.44	0.33	0.38
	ehv+shv	0.63	0.9	0.74	0.19	0.5	0.28	0.84	0.47	0.6	0.48	0.4	0.44
	ehv+shv+dv	0.61	0.9	0.72	0.24	0.5	**0.32**	**0.85**	0.46	0.6	0.49	0.42	0.45
SMOTE-Tomek Links	ehv	0.74	0.71	0.73	0.21	0.27	0.24	0.75	0.78	0.76	0.5	0.47	0.48
	shv	0.58	0.57	0.57	0.24	0.18	0.21	0.6	0.63	0.62	0.36	0.35	0.35
	dv	0.73	0.69	0.71	0.28	0.23	0.25	0.66	0.76	0.71	0.55	0.47	0.51
	ehv+shv	**0.77**	0.71	0.74	0.11	0.14	0.12	0.72	0.79	0.75	0.5	0.47	0.48
	ehv+shv+dv	0.76	0.76	0.76	0.26	0.27	0.27	0.73	0.82	0.77	0.63	0.5	**0.56**

5 Results and Discussion

5.1 Result of Eval-1

In the experiment of Eval-1, only the accuracy was calculated. Table 7 shows the top similarity thresholds, k values, and feature combinations for each class hierarchy (Large, Sub).

In the large class, the combination of emotional embedding and semantic embedding has the highest accuracy.

In the sub-class, the best accuracy is obtained when only emotional embedding is used. In the emotion classification of emotional expressions,

Table 9. Precision, Recall, and F1-score for each emotion class (Web-Corpus)

Web-Corpus Result		A			B			C			D		
Resampling	**Comb.ID**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**
SMOTE	ehv	0.72	0.69	0.7	0.37	0.23	0.28	0.51	0.57	0.54	0.61	0.61	0.61
	shv	0.66	0.71	0.68	0.58	0.18	0.28	0.5	0.44	0.47	0.58	0.59	0.58
	dv	0.71	0.75	**0.73**	**0.65**	0.23	**0.34**	**0.57**	0.53	0.55	0.64	0.64	**0.64**
	ehv+shv	0.7	0.72	0.71	0.58	0.19	0.29	0.53	0.53	0.53	0.61	0.62	0.61
	ehv+shv+dv	0.71	0.74	**0.73**	0.63	0.21	0.31	**0.57**	0.54	0.55	0.63	0.64	**0.64**
ENN	ehv	0.54	0.84	0.66	0.08	0.06	0.07	0.46	0.17	0.25	0.52	0.35	0.42
	shv	0.49	**0.88**	0.63	0.06	0.04	0.05	0.32	0.02	0.05	0.47	0.25	0.33
	dv	0.55	0.81	0.66	0.09	0.12	0.1	0.43	0.14	0.21	0.52	0.41	0.46
	ehv+shv	0.55	0.83	0.66	0.09	0.07	0.08	0.45	0.19	0.26	0.52	0.37	0.43
	ehv+shv+dv	0.56	0.83	0.67	0.09	0.1	0.09	0.45	0.19	0.26	0.54	0.39	0.45
SMOTE-ENN	**ehv**	0.6	0.76	0.67	0.1	**0.33**	0.15	0.44	0.45	0.45	0.65	0.29	0.41
	shv	0.51	**0.88**	0.64	0.13	0.14	0.14	0.41	0.23	0.3	0.65	0.18	0.28
	dv	0.57	0.87	0.69	0.31	0.21	0.25	0.5	0.37	0.43	0.7	0.31	0.43
	ehv+shv	0.56	0.84	0.68	0.16	0.18	0.17	0.46	0.38	0.42	0.68	0.27	0.39
	ehv+shv+dv	0.57	0.87	0.69	0.28	0.2	0.23	0.5	0.38	0.43	**0.71**	0.31	0.43
SMOTE-Tomek Links	**ehv**	**0.73**	0.66	0.69	0.22	0.27	0.24	0.5	**0.58**	0.54	0.6	0.62	0.61
	shv	0.66	0.7	0.68	0.45	0.19	0.26	0.49	0.44	0.47	0.58	0.59	0.58
	dv	0.71	0.74	**0.73**	0.57	0.23	0.32	**0.57**	0.54	**0.56**	0.64	**0.65**	**0.64**
	ehv+shv	0.71	0.71	0.71	0.49	0.22	0.3	0.54	0.54	0.54	0.61	0.63	0.62
	ehv+shv+dv	0.72	0.74	**0.73**	0.6	0.23	0.33	0.56	0.54	0.55	0.63	**0.65**	**0.64**

emotional embedding is effective, but semantic embedding is not so effective by itself; however, if it is combined with other features, it might be effective for expressions that cannot be classified properly by other features alone.

5.2 Result of Eval-2

Table 8 shows the values of Precision, Recall, and F1-score for each combination of features and the resampling method when J-Corpus is used. The results showed that the feature combination (Comb. ID=5) using SMOTE (with all three types of features) yielded the best results overall. In the case where only emotional embedding (ehv) is used as a feature (Comb. ID=1), emotion B (surprise) demonstrated relatively high scores.

When semantic embedding (shv) was added to emotional embedding (Comb. ID=4), the scores for all emotions, except for emotion B (surprise), were relatively high, and the overall accuracy was also improved. This suggests that emotional and semantic embeddings can complement each other.

Figure 4 shows a graph comparing the correct answer rate, the macro-average correct answer rate, and the weighted average correct answer rate. The feature combination (Comb ID=4) (ehv+shv) exhibited the best performance. These results indicate that two features of emotional embedding and semantic embedding are effective, and SMOTE is suitable as a resampling method.

Next, the results when the Web-Corpus was used are shown in Table 9 and Figure 5, as in the case of J-Corpus. When DistilBERT was used alone, the efficiency was the highest. This may be due to the fact that, unlike Web-Corpus, Web-Corpus has many colloquial expressions, and that emoticons other than emotional expressions are used frequently in tweets and blog posts.

6 Conclusion

We proposed a method to learn emotional and semantic embeddings based on a Japanese dictionary of emotional expressions and using a pre-trained model as the initial feature. Because the proposed method embeds both emotions and

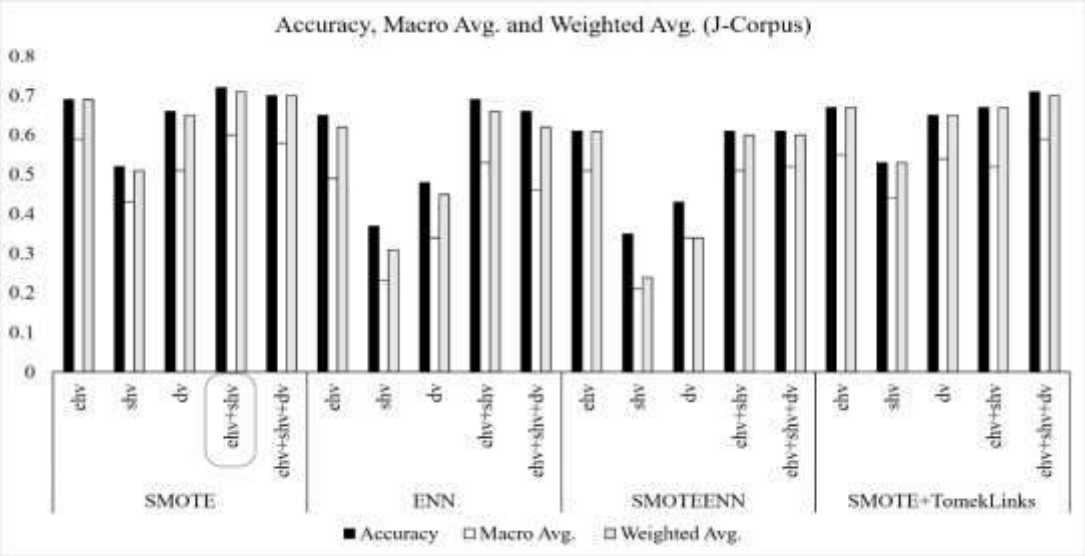


Fig. 4. Comparison of accuracy for each feature combination (J-Corpus)

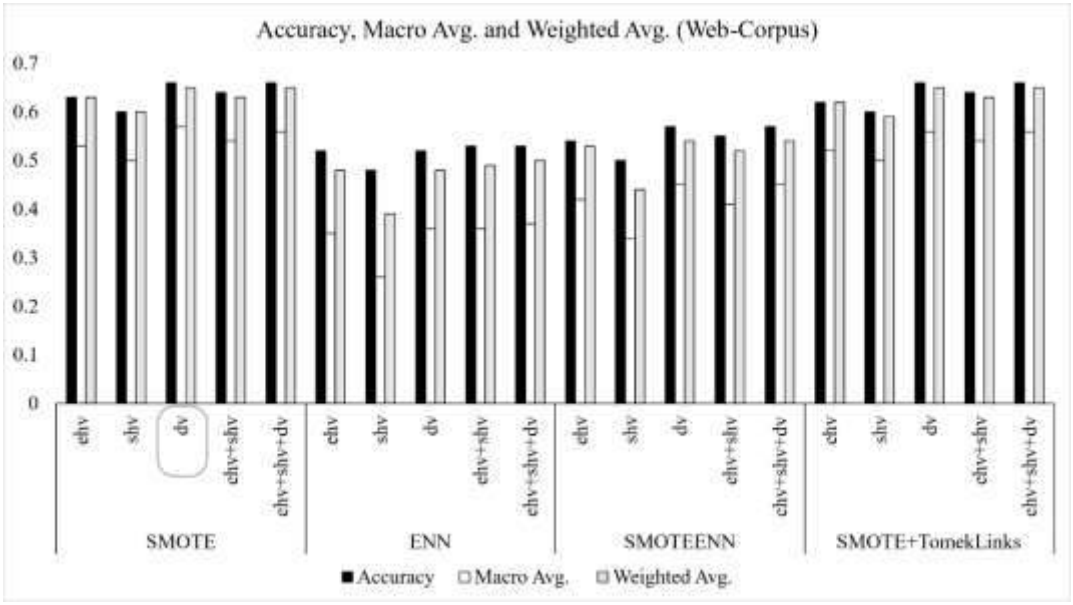


Fig. 5. Comparison of accuracy for each feature combination (Web-Corpus)

semantics, it can be said that it is more specialized for emotion analysis than existing language models.

To evaluate the validity of the proposed method, we conducted two experiments.

The first is a classification experiment on unknown emotional expressions based on the k-nearest neighbor method using words and phrases registered in the emotional expression dictionary.

In this experiment, using both emotional and semantic embedding, we observed a higher rate of correct answers than using only DistilBERT and demonstrated the effectiveness of the proposed method.

The other experiment was an emotion classification experiment on the corpus of utterances with the annotation of sentiment labels. We used a machine learning model based on the gradient boosting method and resampling methods, such as SMOTE, to deal with imbalances between classes, and then cross-validated the accuracy of the models.

In the experiments using the example sentence corpus, the proposed method of adding emotional embedding and semantic embedding showed better performance than using only DistilBERT's distributed representation. Meanwhile, in the experiment using the Web corpus, the performance was highest when only DistilBERT was used, indicating that it was not effective.

This may be owing to the fact that both emotion and semantic embedding are based on the data in the dictionary, and it may have been difficult to deal with the phrases unique to colloquial sentences used on the Web.

In the future, we would like to improve the accuracy by using a pre-training model that is fine-tuned based on a corpus containing a large number of colloquial sentences.

Acknowledgments

This work was supported by JSPS KAKENHI (Grant Number JP20K12027, JP21K12141).

References

1. **Plutchik, R. (1980).** A General Psychoevolutionary Theory of Emotion. Theories of Emotion, pp. 3–22.
2. **Ptaszynski, M., Dybala, P., Shi, W., Rzepka, R., Araki, K. (2009).** A System for Affect Analysis of Utterances in Japanese Supported with Web Mining. Journal of Japan Society for Fuzzy Theory and Intelligent Informatics, Vol. 21, No. 2, pp. 30–49.
3. **Sano, M. (2012)**. The classification of japanese evaluative expressions and the construction of a dictionary of attitudinal lexis: An interpretation from appraisal perspective, NINJAL Research Papers, pp. 53–83.
4. **Goleman, D. (2012).** Emotional intelligence. New York: Bantam Books.
5. **Strapparava, C., Valitutti, A. (2004).** WordNet affect: An affective extension of WordNet. Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04), pp. 1083–1086.
6. **Kobayashi, N., Inui, K., Matsumoto, Y., Tateishi, K. (2005).** Collecting evaluative expressions for opinion extraction. Journal of Natural Language Processing, Vol. 12, No. 3, pp. 203–222.
7. **Nakamura, A. (1993).** Kanjo hyogen jiten [Dictionary of Emotive Expression]. Tokyodo Publishing.
8. **Wang, S., Maoliniyazi, A., Wu, X., Meng, X. (2020).** Emo2Vec: Learning emotional embeddings via multi-emotion category. ACM Transactions on Internet Technology, Vol. 20, No. 2, pp. 11–17. DOI:10.1145/3372152.
9. **Fishcer, K.W. (1989).** A skill approach to emotional development: From basic- to subordinate-category emotions **Damon, W. (Ed.).** Child development today and tomorrow, pp. 107–136.
10. **Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. (2013).** Distributed Representations of Words and Phrases and their Compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13), Vol. 2, pp. 3111–3119.
11. **Joulin, A., Grave, E., Bojanowski, P., Mikolov, T. (2017).** Bag of Tricks for Efficient Text Classification. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Vol. 2, pp. 427–431.
12. **Pennington, J., Socher, R., Manning, C.D (2014).** GloVe: Global vectors for word representation. Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP´14), pp. 1532–1543.
13. **Devlin, J., Chang, M.W., Lee, K., Toutanova, K. (2018)**. BERT: Pre-training of deep bidirectional transformers for language understanding. http://arxiv.org/abs/1810.04805.
14. **Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. (2020).** ALBERT: A Lite BERT for self-supervised learning of language representations. Proceedings of The International Conference on Learning Representations (ICLR2020).
15. **Sanh, V., Debut, L., Chaumond, J., Wolf, T. (2019).** DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. Thirty-third Conference on Neural Information Processing Systems (NIPS).
16. **Miller, G.A. (1995).** WordNet: A lexical database for English. Communications of the ACM, Vol. 38, No. 11, pp. 39–41.

17. **Nakayama, K., Hara, T., Nishio, S. (2006).** Wikipedia Mining to Construct a Thesaurus. Journal of Information Processing Society of Japan, Vol. 47, No. 10, pp. 2917-2928.
18. **Diganta, M. (2019).** Mish: A self regularized non-monotonic activation function. arXiv preprint arXiv:1908.08681.
19. **McInnes, L., Healy, J., Saul, N., GroBberger, L. (2018).** UMAP: Uniform Manifold Approximation and Projection. The Journal of Open-Source Software. DOI:10.21105/joss.00861.
20. **Minato, J., Matsumoto, K., Ren, F., Tsuchiya, S., Kuroiwa, S. (2008).** Evaluation of emotion estimation methods based on statistic features of emotion tagged corpus. International Journal of Innovative Computing, Information and Control, Vol. 4, No. 8, pp. 1931–1941.
21. **Matsumoto, K., Ren, F. (2011).** Estimation of word emotions based on part of speech and positional information. Computers in Human Behavior, Vol. 27, No. 5, pp. 1553–1564.
22. **Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002).** SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, Vol. 16, pp. 321–357.
23. **Wilson, D. (1972).** Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 2, No. 3, pp. 408-421.
24. **Batista, G., Bazzan, A., Monard, M. (2003).** Balancing training data for automated annotation of keywords: A case study. Proceedings of the 2nd Brazilian Workshop on Bioinformatics, pp. 10–18.
25. **Batista, G., Prati, R., Monard, M. (2004).** A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, Vol. 6, No. 1, pp. 20–29.

Article received on 14/06/2021; accepted 05/09/2021.
Corresponding author is Kazuyuki Matsumoto.

ISSN 2007-9737

Resource Search in HPC Systems using Lévy Flights

Apolinar Velarde Martínez

Instituto Tecnológico El Llano Aguascalientes,
Departamento de Sistemas y Computación,
Mexico

apolinar.vm@llano.tecnm.mx

Abstract. Parallel applications represented by Directed Acyclic Graphs (DAGs) as Parallel Task Graphs (PTGs) requiring high execution times with large amounts of storage, are executed on High Performance Computing (HPC) Systems such as clusters. For the execution of these applications, a scheduler performs the scheduling and allocation of the resources contained in the HPC System. One of the activities of the scheduler is the search for idle resources that are geographically dispersed in the clusters to schedule and allocate them to the tasks. The search for idle resources in the clusters is a process that consumes time and system resources due to the geographical distances that must be traveled and the repetitive and permanent execution within the system. A sequential search for resources causes the scheduling and allocation of resources in the HPC System to be slowed down and paused, and increases the waiting times of the tasks that remain in the queue. The techniques that shorten the location times of resource and perform more exhaustive searches in dispersed geographic spaces can reduce the times generated by sequential searches. The open access paper: Scheduling in Heterogeneous Distributed Computing Systems Based on Internal Structure of Parallel Tasks Graphs with Meta-Heuristics presents the Array Method, a scheduler for scheduling and allocation resources in an HPC System. Array Method uses a sequential search process for the idle resources that are geographically dispersed in the clusters and save their location and characteristics in an array, which is updated every time idle resources are located in the clusters. Considering the above, this paper presents a search for idle resources using Lévy random walks, a technique used for searching resources in large geographical spaces; this technique avoids sequential node-by-node searches of each cluster, and promote short and long range searches over the entire geographical extent of the HPC Systems. To obtain experimental results, sequential resource search algorithm versus Lévy random walks with the synthetic loads, different clusters and different numbers of resources per cluster as proposed in the open access paper aforementioned, are used. Obtained results show Lévy random walks locates more idle resources in less time, optimizes the times of the resource searches in the clusters and update the array of available resources more frequently. With more idle resources found, the total execution times of the tasks are reduced.

Keywords. High performance computing systems, clusters, Lévy flights, scheduling resources, allocation resources.

1 Introduction

Parallel tasks represented by Directed Acyclic Graphs (DAGs) referenced as Parallel Task Graphs (PTGs) and constituted by a set of subtasks of scientific and enterprise applications, require large amounts of processor and storage space during their execution in High Performance Computing Systems (HPC Systems) [1], such as clusters [10], in order to achieve desired level of service [16] to users. Clusters or cluster computing refers to the use of interconnected computers geographically distributed [19, 8] as a high-performance computing platform and has become wide-spread [10, 3].

Often the computers used in a cluster are “commodity” computers, that is, low-cost personal computers as used in the home and office [3]; they consist of PCs heterogeneous, running Linux, and connected with Ethernet, and referred to as Beowulf clusters; since the processors are

independent they are examples of the MIMD (Multiple Instruction, Multiple Data) or SPMD (Single Program, Multiple Data) model [7]. In this work, the processor refers to an individually programmable computer resource [5] that can be assigned to a PTG and in the rest of the paper are referred to as nodes, processing elements (PE) and resources. Resource is a collection of components that can be scheduled to perform an operation by an application [15]; some traditional examples of resources are CPU cores for computing, memory spaces for storage, network links for transferring, electrical power, and so on [18].

When Parallel Tasks Graphs in a cluster are executed, a scheduler performs the scheduling and the allocation resources [10]. The scheduling is performed in two phases: first, a search for idle resources in all clusters is performed to determine the positions of the PEs remain idle; the search for idle resources can be performed by a resource allocation control which is a mechanism that manages and control resources in a cluster system [10]. Second a search for the best allocations to optimize the execution times of the tasks is performed.

In [21], the open access article, an array-based scheduler (called the Array Method) for scheduling and resource allocation in clusters is explained. For scheduling, the Array Method uses an array of idle resources or processors (resource array) that is updated as follows, each time a cluster is added to the target system, a task finishes executing or a PE of a cluster is added as a system resource; for locating idle resources, a sequential search is performed on each of the clusters of the HPC System.

The search is a scan that is performed for each of the processors belonging to the clusters, generally initiated on a central node and executed by a Software Agent (SA); the SA initiates the scan on all the processors belonging to the cluster; upon finishing with a cluster, the SA advances to the neighboring clusters and starts again a similar scanning process, processor by processor; every so often SA informs the central node the number of visited processors and activates a process to update the idle resource matrix (resource matrix). The SA resides in the central node of the HPC Systems and because resources can change from a busy state to an idle state at any time, SA should start the sequential search for idle resources in the clusters and update the resource matrix periodically and continuously. Both processes become fundamental and necessary. Finally, for the assignments of idle resources to tasks, a sequential search is executed in the resource matrix and the process of searching is initiated for the best assignments.

Considering the results of the execution times generated by the matrix method and an analysis of each process executed by the scheduler, the most time-consuming process is the sequential scanning of idle resources in the clusters. In order to improve the resource search times in the clusters that constitute the target system, this paper present a technique of idle resources search with Lévy flights, a special class of random walks that have been investigated in different works [19, 15, 22, 4, 13, 23, 11] as a resource search technique in locations geographically distributed; in the following paragraphs the development and implementation of this technique is justified. The term Lévy random walks is used to refer Lévy flights by context in which this technique is used in this research work.

1.1 Justification of the Proposed Method

The use of a technique of resource search in a HPC System, different from sequential search with acceptable service levels, is justified by the next conditions:

— Search for idle resources, is a process that is executed periodically, constantly and permanently that results in a higher consumption of time and resources of the target system, therefore, it must optimize the search for free resources (locate the largest amount of resources each time it is executed).

— Resource searches must be executed in different clusters; the clusters are geographically distributed [19, 8], to meet the demands of geographically distributed applications [1].

— When searching for resources on the target system, it should not be limited by a sequential search on adjacent clusters, but should be able to migrate to non-adjacent clusters if neighboring clusters are disabled.

— Because the allocation of new nodes in the infrastructure can occur at any time [8]. Each time a cluster is added to the system, the distances for searching and extracting features from the target system resources gradually increase; with the ever increasing size of systems, the task scheduling problem has become more challenging and complex [15].

— The search for resources is considered a vital process to avoid sending a subtask to an inactive node of a cluster, and to assume that the subtask starts its execution. This can occur because a resource may join or leave the network at any time due to dynamic nature of resources [19].

For the aforementioned, in this research work an idle resource search technique with Lévy flight is proposed and experiments that compares the proposed technique with sequential search algorithm for idle resources are accomplish. The synthetic loads proposed in [21] with different number of clusters and different number of resources per cluster was used during the experiments. The results obtained show that the Lévy random walks allow more idle resources to be taken in shorter periods of time, optimize the search times of the resources in the clusters and allow the array of available resources of the scheduler to have constant updates in shorter periods of time (as explained in the experiments section).

Note that the purpose of this research work is the performance analysis of the sequential search of resources versus Lévy flights search and not scheduling tasks in HPCS.

This paper is organized as follows: in section 2, four additional definitions to the Basic Definitions section of [21] are proposed; in section 3, problem statement is established, section 4 addresses the works related to this research; in section 5, a very brief explanation of Array Method, proposed in [21] is presented; section 6, how Lévy random walks are used in this work and how is implemented the search for idle resources in HPC System using Lévy random walks are described; the experiments performed are presented in section 7; finally the conclusions and future works can be found in sections 8 and 9 respectively.

2 Basic Definitions

In this paper, the section of basic definitions of [21] is used and the following new definitions are proposed.

Definition 1. The target system consists of C_l clusters, $C_1, C_2, ..., C_l$ where l is the number of clusters contained in the HPC System. Each cluster contains m heterogeneous processors with n processing cores, then $C_{l,m,n}$ is cluster k, processor m, processing core n. In this way each $C_{l,m,n}$ represents a resource that is identified as $R_{l,m,n}$ and can be used by the scheduler.

Definition 2. Let to consider definition 2 from [21]; it follows that each PTG will request η_i resources, which must be extracted from the scheduler's resource array, then the next condition is established:

$$\forall\ \eta_i\ \exists\ R_{l,m,n}. \tag{1}$$

For the PTG to complete its execution.

Definition 3. A Lévy flight is a type of random walk in which the increments are distributed according to a "large tail" probability distribution. Specifically, the distribution used can be approximated by a power law of the form [23]:

$$P(l)\ \propto\ l^{-\mu}\ with\ 1 < \mu \leq 3, \tag{2}$$

where μ is a constant parameter of the distribution known as the exponent or scaling parameter.

Definition 4. Let a software agent SA, which locates any idle $R_{l,m,n}$ found in any C_l using equation 2, then registers it in an array of resources and condition 1 is satisfied.

3 Problem Statment

Let a scheduler running on an HPC System consisting of C_l clusters, $C_1, C_2, ..., C_l$ where l is the number of clusters of the HPC System. Each cluster contains m heterogeneous processors with n processing cores; for searching $R_1, R_2, ..., R_n$ idle resources, a search engine S operated by a software agent (SA), is executed using equation 2.

Then: respecting the PTGs execution constraints, β must be executed continuously and periodically at times $t_1, t_2, ..., t_n$, where t_n, is the time where the scheduler is active in the target system. Any $R_1, R_2, ..., R_n$ idle resource located by S must be stored in an array of dimension $N \times M$ and be available for use at any time t_n.

4 Related Works

In a plethora of literature, research works that use idle resources for scheduling and allocation of tasks in an HPCS, assume a set of resources that are available in a resource pool as in [19, 15]; based on required characteristics, the resource discovery mechanism searches and returns the addresses of the resources that match with the provided descriptions [15], in [1] resources are scheduled prior to path setup request, the allocator has a resources' metadata [2]; in [20] assume that each user contributes a certain number of machines (resources) to its common pool of machines (resources) in the cloud, [8] only proposes a resource management system manage a pool of processing elements (computers or processors) which is dynamically configured (i.e., processing elements may join or leave the pool at any time); other research works such as [14] establishes a model where capabilities of the machines are known: the number of cores, the amount of memory, the disk space, and the host OS running, and supposes that a large data center and cloud systems already have significant monitoring tools that provide near-real-time updates of various systems to their controllers. In [18] the resources are considered to be available on demand, charged on a pay-as-you-go basis, and in one aspect, cloud providers hold enormous computing resources in their data centers, while in the other aspect, cloud users lease the resources from cloud providers to run their applications; [9] the cloud service end user can use the entire stack of computing services, which ranges from hardware to applications. [17] supposses a cloud of resources where tasks are scheduled, the resources are heterogeneous and characterized by power and cost constraints. With the resources, metrics such as makespan, throughput [15], waiting time [19, 16, 10], overall mean task response time [8], load balancing [12] are sought to improve in HPC Systems.

In contradistinction to the works described above, in this research work any set of resources for the schedule are not assumed, but proposes the development of idle resources search engine operated by a software agent; software agent executes a resource discovery procedure [19, 15], which allows visiting a set of available clusters in the shortest possible time.

Considering Lévy random walks [22, 4, 6] as a prominent area of research in various disciplines from ecology to physics [23], and a special class of random walks whose stride lengths are not constant but are selected from a probability distribution with a power law [22, 4, 13, 23], in addition to the hypothesis that Lévy random walks are optimal when exploring unpredictably distributed resources [13] have been proposed in this work as a strategy of search for idle resources in an HPC System. Similarly, Levy's random walks are used in this work as a search strategy, because due to the divergence of the variance, extremely long jumps can occur and the typical trajectories are self-similar in all scales, showing groups of jumps shorter interspersed [6].

5 Array Method

The array method [21], is a dynamic scheduler for scheduling and allocating tasks in an HPC system. The operation of this scheduler is based on a set of arrays that store the results generated from each function. For each of the arrays, iterative processes or loops are executed. A very brief review of the data structures used by the Array Method is presented below.

— The resource matrix. The values of this matrix are obtained through an iterative process of resources searching in the HPC system.

— The matrix of characteristics of the PTGs. The PTG characteristics array stores the characteristics extracted from each PTG using the Depth First Search Algorithm (DFS Algorithm)

— The allocation matrix, is a dynamic array that stores the values produced by the resource allocation algorithm.

— The matrix of task start times. Once the algorithm determines the best assignment for the PTG, start times for each of the PTG subtasks are calculated; these values are stored in the matrix of task start times.

The software agent SA, is executed over all clusters; all characteristics of idle resources found are stored in resource matrix, i.e., when the SA is executed, it only updates the resource matrix so the planning and allocation process can use idle resources.

The Software Agent execution has been defined so far. On the next paragraphs Lévy random walks and its SA interaction with, will be defined; how this interaction operates over the clusters are defined too.

6 Lévy Random Walks

This section defines how Lévy random walks are used in this research work. The exponential increase in step length gives the Lévy flight the property of being scale invariant, and they are used to model data exhibiting clustering. Lévy flights are a special class of random walks whose step lengths are not constant, but are selected from a probability distribution with a power law [22, 4, 13, 23]. The random walk is represented by a succession of random variables X_n with $n \in N$ known as a discrete stochastic process. The sequence $X_n, n \in N$ forms infinite sequences $X_0, X_1, ..., X_i$ with $X_i \in Z$. That is, if a run starts at state 0, at the next time the agent can move to position $+1$, with probability p or to position -1 with probability q, with $p + q = 1$. Position $+1$ is considered the PE to the left of the current PE and position -1 is considered the PE to the right of the current PE.

In this research work we analyze the case of a software agent which starts from a specific cluster and PE, and moves in stages or steps along the target system, at each step it moves a unit distance to the right or to the left, with respectively equal probabilities. For the movement of the agent, let ζ_n as the nth motion of the agent and P as the probability function then,

1. $P(\zeta_n = +1) = p$ the agent moves one PE to the right of the current position.
2. $P(\zeta_n = -1) = q$ the agent moves one PE to the left of the current position.

ζ_n are considered as the independent random variables.

It is further assumed that the agent moves k steps in total, before registering idle resources in the resource matrix (see section 5). For these experiments, Lévy flights are only used as a search procedure for idle resources, not as a prediction algorithm. The steps for the execution of the software agent are explained on next section (a reduced algorithm is provided in Table 1).

6.1 Search for Idle Resources in HPC System using Lévy Random Walks

For the resource allocation in HPC Systems, most of research works (as Related Works section explains) assume a centralized resource manager that has a complete vision of network topology, as well as networking and computing resources status; this assumption is not valid for large-scale worldwide grid networks; practically, grid network comprises geographically distributed heterogeneous resources interconnected by multi-domains networks [1]. In this paper, before performing the scheduling and allocation resources, a resource search engine is proposed in HPC Systems, using Lévy random walks to extract the characteristics of each processing element, assuming that ignoring computational resources capacity and availability may affect the overall performance

significantly specially in computational intensive applications [1]. To generalize the problem, lets consider the figure 1, which has a set of geographically dispersed clusters linked by wireless and wired communication; figure 1 is the most similar to actual architecture used in the experimentations. For reasons of space, details about network bandwidth, multidomain environments, interdomain, and intradomain topology are not specified, in addition to how the different domains interact to provide end-to-end connectivity.

On figure 1 circles represent processing elements; filled circles show occupied PEs, circles filled with lines represent PEs without functionality, while unfilled PEs are idle at time $t + 1$ and can be located by the software agent. Links between processing elements within a cluster and links between clusters are represented by solid lines. The three dots above the solid lines show the possible addition of new clusters to the target system.

The example showed in the Figure 1 up operates like this: the resources search for in the HPC System using Lévy random walks, is shown with a dotted line; the Lévy random walk executed by software agent using algorithm showed in Table 1, starts in cluster 1, executes a set of steps for searching the PEs on this cluster, migrates to cluster 2, 3, 4 and 5; on the cluster 5 a long jumps is executed. The search process is repeated for the time set in the algorithm; at the end of the random walk, twenty idle resources are identified by the SA, resource matrix is updated and resources are available for scheduling and allocation.

Considering the algorithm proposed in [11], Table 1 shows a reduced structure of the resource search algorithm using Lévy's random walks, operated by the software agent. On next sections the results of the experiments performed are described.

7 Experiments

Target System. Experiments are performed on 450 desktop computers and a server farm with 10 servers. Clusters between 5, 10 and 20 computers were built and distributed in different buildings within the campus.

Algorithm 1. Algorithm for resources search

$Input:\ Starting\ position\ for\ search\ in\ HPC$
$Output:\ List\ of\ idle\ resources\ in\ HPC$
$Begin$
  $Assigns\ search\ time\ by\ user;$
  $Identify\ starting\ position\ of\ SA\ in\ HPC;$
    $Location\ of\ last\ search;$
    $Start\ search\ from\ cluster\ identified\ as\ starting\ cluster;$
  $While\ (Search\ time\ exists)$
    $Calculate\ movement\ of\ SA\ using\ equation\ 2;$
    $Identify\ and\ verify\ status\ of\ PE\ where\ SA\ is\ to\ be\ moved;$
    $If\ (PE\ Status\ ==\ Idle)$
      $Save\ position\ and\ characteristics\ of\ PE;$
    End_if
  End_While
  $Update\ resource\ matrix$
end

The server farm is considered the initial cluster and where the task queue remains.

Agent Software. The software agent is programmed with C language, and its movements are allowed through C shell programming within the target system. Once the agent is transported to a PE, the execution of the agent is performed with a Linux Cron utility programmed in each EP to detect the arrival or presence of the agent in the PE.

Considering the purpose of this survey is the performance analysis of the resources search techniques, two metrics for the experiments in this paper are used:

— The percentages of resources located after established number of executions by each algorithm (Lévy random walks and sequential search) in the target system.

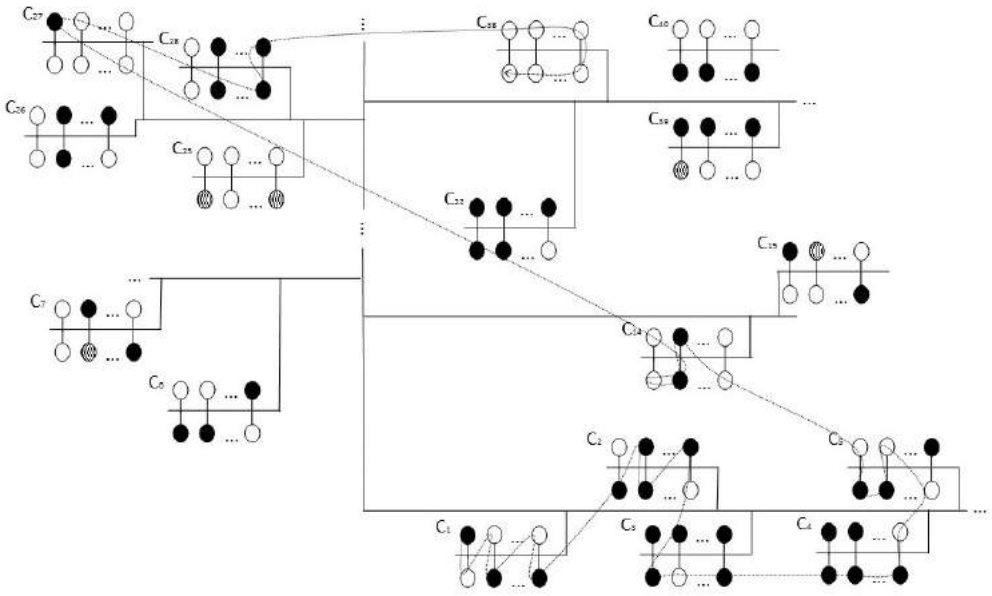


Fig. 1. HPC Systems with clusters showing a search for idle resources using Lévy random walks

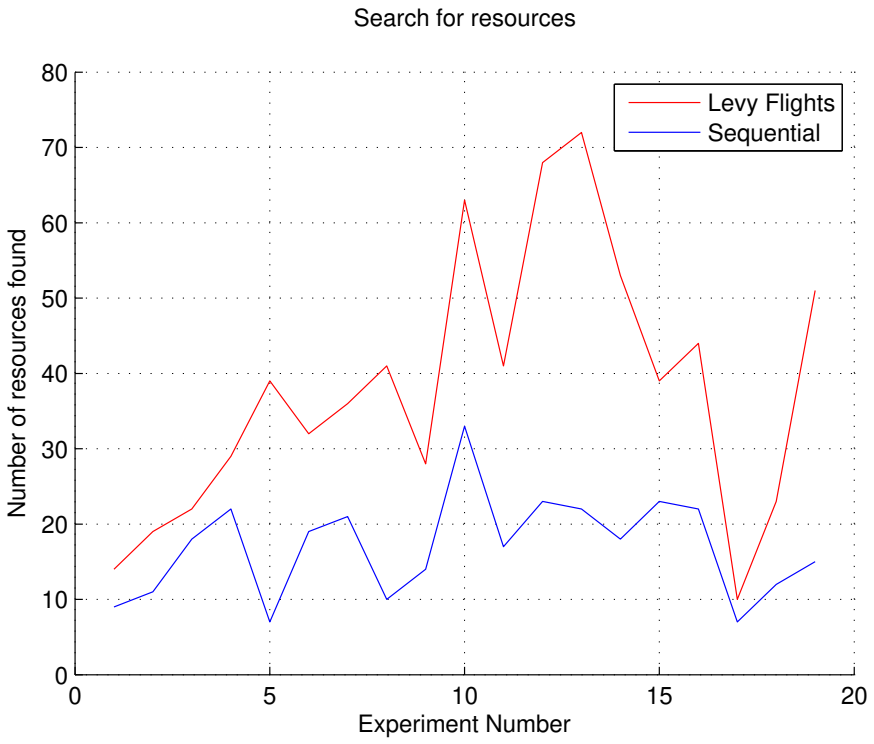


Fig. 2. The percentage of resources located by search engine operated by SA in the target system

— The disaggregation of tasks in the HPC System clusters, during scheduling and resource allocation, i.e., how tasks are allocated in the target system according to the resource search technique used, sequential or Levy flights.

Each experiment and the results obtained are explained in the following paragraphs.

The percentage of resources located by each algorithm in the target system at time t. In these experiments two types of searches are executed as follows: from the server farm and from where the last search ended; with these experiments it is possible to measure the ability of each technique to locate the resources in the target system, in time t.

For each of these experiments, numbered 1 to 20, a set of 100 tests is run; in 50 tests the starting position of the search is in the initial cluster and 50 tests the starting position is the location of the last search; alternating the experiments, a test is run with the first type of search and then with the second type of search.

For experiment 1, the tests are started with a time t of 5 minutes, 100 experiments are run, the result of each of the 100 runs is the number of resources found at time t (5 minutes); the total sum of the resources found in each run is divided by 100, and the result is reported for this experiment. For a second experiment the time t is 10 minutes and the 100 runs, the results are obtained in the same way as experiment 1. For each experiment 5 minutes are incremented. The results of all experiments are shown in graph 2.

Results (Experiment 1). The results obtained in these experiments show that the random walk search is more efficient for locating geographically distributed resources, selected with a probability distribution; this search method allows moving to clusters that may be without assigned tasks (completely idle).

During the execution of this search method, subgroups with different amounts of free resources were found in the clusters, before the migration of the SA from one cluster to another; with these groups, the system can send tasks with a number of subtasks that are equal to the subgroups of resources found, allowing the execution of the tasks with their subtasks in the same cluster.

In contrast to first experiment, we propose an experimentation related to the way in which the tasks are disaggregated into the resources found by each type of search executed: the sequential search and the search using Lévy flights. The following paragraphs explain this experimentation.

Disaggregation of the tasks in the HPC System clusters, during the resource allocation process. Once the resources are located the resource matrix and the matrix of characteristics are updated; the next step of the algorithm is to assign the tasks to the available resources.

For this experimentation, the total execution time of the tasks was measured considering that Lévy random walks searches can locate resources with longer geographic distances. When the subtasks of a task are executed in geographically distant clusters with long distances, the communication times between tasks increase gradually, depending on the speed of the network devices of the target system. The workloads of [21] were chosen for this experiment.

The tasks of each workload were scheduled and allocated to the resources that each algorithm found in the HPC System. The completion times of the tasks of each workload were measured considering the communication time between subtasks of each task. The results of these experiments are shown in graph 3.

Results (Experiment 2). Results obtained in the experiments of scheduling and allocation of localized resources by each method have shown the outperformance of the Lévy random walks, since lower task execution times are obtained.

More resources that this technique can locate in the resource search executions, allow to make more task assignments in geographically distributed clusters; assigning tasks with their subtasks grouped in the same cluster can decrease communication times and the task execution is faster. When scheduling is executed, the scheduler can have more available resources for assignment, and allocation can be done in close clusters of resources.

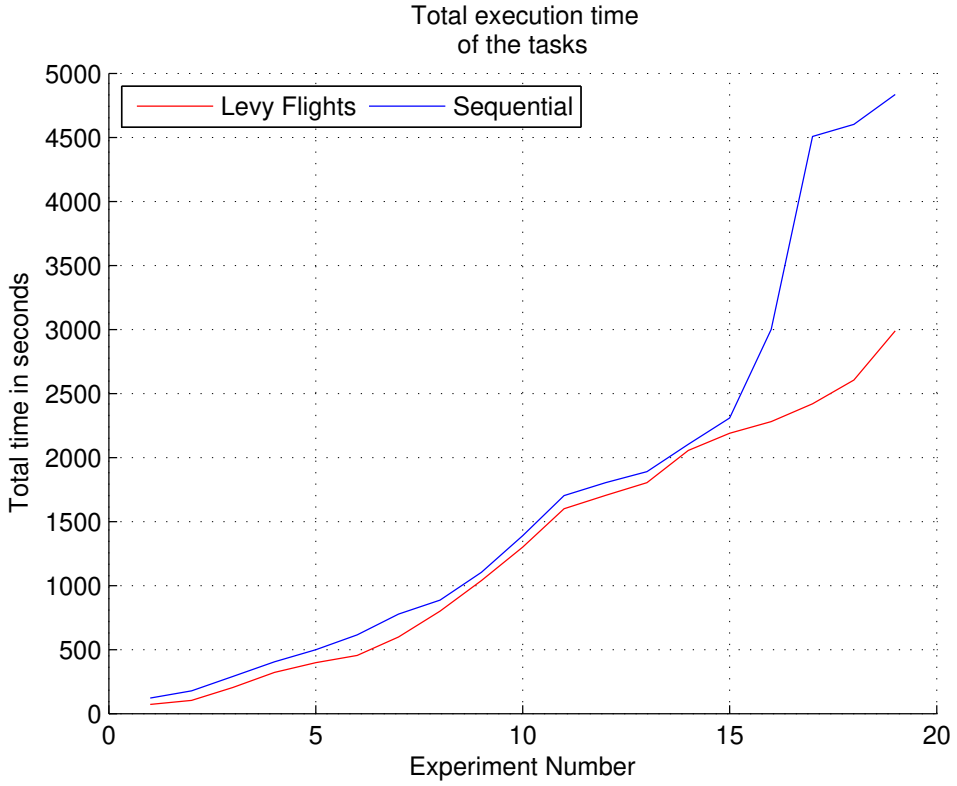


Fig. 3. The percentage of resources located by each search engine in the target system at time t

8 Conclusion and Future Work

The emulation in computational environments of scale-invariant phenomena observed in biological systems has led to propose solutions to problems of the non-deterministic polynomial time type. In this research work we conducted an exploratory study to propose a solution that allows the search for idle resources in an HPC System, specifically in a cluster system using Lévy random walks.

The justification of this work is the development of a software agent which migrates between clusters of an HPC system and simulates Lévy random walks. With this software agent a comparison is made with two performance metrics between a sequential search of resources and Lévy random walks: the number of resources located in a given time, and in contrast the total execution time of the tasks using the resources located by each technique is measured.

In the experiments with the first metric, Lévy random walks allow visiting clusters in geographic distances that a sequential resource-by-resource search would take a longer amount of time. With the second metric, the total execution time of the tasks using the resources obtained with each algorithm of search has been evaluated; this total time is measured from the assignment of the resources to the task and the dispatch of the task, until the task finishes its execution and returns to the central node.

In the experiments, it is verified that, the total times generated with the resources obtained with the Lévy random walks' algorithm, are not so far from the total times obtained with the resources located in the sequential search, when the workloads contain less than 500 PTGs for their processing; shorter total execution times of the tasks are highlighted when the workloads are dense.

Finally, it should be noted that the resources, architecture and configuration of the computer network can affect the performance of the algorithms when the configurations are not adequate.

8.1 Future Works

Research projects under development preceding this research work, include experiments with real application workloads using the same number of clusters in real environment, in order to measure task execution times and performance of the underlying computer network.

Second work, includes design and programming of the Array Method using hybrid programming with OpenMP and MPI (Message Passing Interface), in order to perform experiments with sequential executions and parallel executions; these execution comparisons will allow evaluating the performance of the Array Method.

Acknowledgments

This project is funded by Tecnológico Nacional de México TecNM. I express special thanks for support to Instituto Tecnológico El Llano Aguascalientes.

References

1. **Abouelela, M., El-Darieby, M. (2012).** Multidomain hierarchical resource allocation for grid applications. Journal of Electrical and Computer Engineering, Vol. 2012.
2. **Alkhalaileh, M., Calheiros, R., Nguyen, Q., Javadi, B. (2019).** Dynamic Resource Allocation in Hybrid Mobile Cloud Computing for Data-Intensive Applications. pp. 176–191.
3. **Baronchelli, A., Radicchi, F. (2013).** Lévy flights in human behavior and cognition. Chaos Solitons & Fractals, Vol. 56.
4. **Casanova, H., Desprez, F., Suter, F. (2010).** On cluster resource allocation for multiple parallel task graphs. Journal of Parallel and Distributed Computing, Vol. 70, No. 12, pp. 1193–1203.
5. **Chechkin, A., Metzler, R., Klafter, J., Gonchar, V. (2008).** Introduction to the Theory of Lévy Flights. pp. 129–162.
6. **Eijkhout, V., van de Geijn, R., Chow, E. (2016).** Introduction to High Performance Scientific Computing.
7. **El-Zoghdy, S., Nofal, M., Shohla, M., El-sawy, A. (2013).** An efficient algorithm for resource allocation in parallel and distributed computing systems. International Journal of Advanced Computer Science and Applications, Vol. 4.
8. **Gawali, M., Shinde, S. (2018).** Task scheduling and resource allocation in cloud computing using a heuristic approach. Journal of Cloud Computing, Vol. 7.
9. **Hussain, H., Malik, S. U. R., Hameed, A., Khan, S. U., Bickler, G., Min-Allah, N., Qureshi, M. B., Zhang, L., Wang, Y.-J., Ghani, N., Kolodziej, J., Zomaya, A. Y., Xu, C.-Z., Balaji, P., Vishnu, A., Pinel, F., Pecero, J. E., Kliazovich, D., Bouvry, P., Li, H., Wang, L., Chen, D., Rayes, A. (2013).** A survey on resource allocation in high performance distributed computing systems. Parallel Comput., Vol. 39, pp. 709–736.
10. **Kamaruzaman, A., Zain, A., Yusuf, S., Udin, A. (2013).** Levy flight algorithm for optimization problems – a literature review. Applied Mechanics and Materials, Vol. 421.
11. **Kumar, R., Chaturvedi, A. (2021).** Improved Cuckoo Search with Artificial Bee Colony for Efficient Load Balancing in Cloud Computing Environment. pp. 123–131.
12. **Murakami, H., Feliciani, C., Nishinari, K. (2019).** Lévy walk process in self-organization of pedestrian crowds. Journal of The Royal Society Interface, Vol. 16, pp. 20180939.
13. **Pillai, P., Rao, S. (2016).** Resource allocation in cloud computing using the uncertainty principle of game theory. IEEE Systems Journal, Vol. 10, pp. 637–.
14. **Qureshi, M., Dehnavi, M., Min Allah, N., Qureshi, M., Hussain, H., Rentifis, I., Tziritas, N., Loukopoulos, T., Khan, S., Xu, C.-Z., Zomaya, A. (2014).** Survey on grid resource allocation mechanisms. Journal of Grid Computing, Vol. 12, pp. 399–441.
15. **Qureshi, M., Qureshi, M., Fayaz, M., Mashwani, W., Brahim Belhaouari, S., Hassan, S., Shah, A. (2020).** A comparative analysis of resource allocation schemes for real-time services in high performance computing systems. International Journal of Distributed Sensor Networks, Vol. 16, pp. 2020.

16. **Qureshi, M., Qureshi, M., Fayaz, M., Zakarya, M., Aslam, S., Shah, A. (2020).** Time and cost efficient cloud resource allocation for real-time data-intensive smart systems. Energies, Vol. 13.
17. **Ren, Z., Zhang, X., Shi, W. (2015).** Resource Scheduling in Data-Centric Systems. Springer New York, New York, NY, pp. 1307–1330.
18. **Shukla, A., Kumar, S., Singh, H. (2019).** An improved resource allocation model for grid computing environment. International Journal of Intelligent Engineering and Systems, Vol. 12, pp. 104–113.
19. **Tang, S., Lee, B., He, B. (2016).** Fair resource allocation for data-intensive computing in the cloud. IEEE Transactions on Services Computing, Vol. PP, pp. 1–1.
20. **Velarde, A. (2020).** Scheduling in heterogeneous distributed computing systems based on internal structure of parallel tasks graphs with meta-heuristics, pp. 1–22.
21. **Viswanathan, G., Afanasyev, V., Buldyrev, S., Murphy, E., Prince, P., Stanley, H. (1996).** Lévy flight search patterns of wandering albatrosses. Nature, Vol. 381.
22. **Wilkinson, B., Allen, M. (2005).** chapter Parallel programming techniques and Applications Using Networked Workstations and Parallel Computers. Pearson Education, pp. .
23. **Zhao, K., Jurdak, R., Liu, J., Westcott, D., Kusy, B., Parry, H., Sommer, P., McKeown, A. (2015).** Optimal Levy-flight foraging in a finite landscape. Journal of the Royal Society, Interface / the Royal Society, Vol. 12.

Article received on 25/06/2021; accepted on 07/10/2022.
Corresponding author is Apolinar Velarde Martinez.

ISSN 2007-9737

Deep Learning Approach for Aspect-Based Sentiment Analysis of Restaurants Reviews in Spanish

Bella-Citlali Martínez-Seis[1], Obdulia Pichardo-Lagunas[1], Sabino Miranda[2,3], Israel-Josafat Perez-Cazares[1], Jorge-Armando Rodriguez-González[1]

[1] Instituto Politécnico Nacional,
Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas,
Mexico

[2] INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación,
Mexico

[3] CONACyT Consejo Nacional de Ciencia y Tecnología,
Dirección de Cátedras,
Mexico

{bcmartinez, opichardola}@ipn.mx, sabino.miranda@infotec.mx

Abstract. Online reviews of products and services have become important for customers and enterprises. Recent research focuses on analyzing and managing those kinds of reviews using natural language processing. This paper focuses on aspect-based sentiment analysis for reviews in Spanish. First, the reviews data sets are normalized into different inputs of the neural networks. Then, our approach combines two deep learning models architectures to determine a positive or negative assessment and identify the most important characteristics or aspects of the text. We develop two architectures for aspect detection and three architectures for sentiment analysis. Merging the deep learning models, we tested our approach in restaurant reviews and compared them with state-of-the-art methods.

Keywords. Customer reviews, polarity classification, sentiment analysis.

1 Introduction

Nowadays, it is common to write a review of a product or service. 84% of consumers trust online reviews as much as a personal recommendation [22]; therefore, review sites have become crucial to consumers. Enterprises consider those reviews as feedback for their products [4]. It allows them to analyze strengths and weaknesses in order to improve the service or product.

Recent research focuses on analyzing and managing those reviews using natural language processing. Aspect-based sentiment analysis not only determines a positive or negative assessment, but also identifies the most important characteristics or aspects of the text [17, 7]. For example, the following review, *A bad service cannot be saved by a good food*, should be classified as positive on the food area, but negative on the service area. It is a major technological challenge [20] because even humans often disagree on the sentiment of a given text, and moreover, on the aspect that the text is talking about.

Enterprises pursue a positive reputation as one of the most powerful marketing assets. This paper focuses on processing, analyzing, and categorizing the large accumulations of information generated from the reviews.

Our approach combines two deep learning models for aspect-based sentiment analysis. The reviews were normalized into five different data

sets inputs for the test of the neural networks. We compare state-of-the-art approaches, and the performance of our aspect classification proposal is promising, but there is still work to do in the sentiment detection.

The rest of the paper is organized as follows. Section 2 is a review of the research involving aspect-based sentiment analysis in Spanish, mainly based on subtask 1 of task 5 within the 2016 edition of SemEval competition [19]. After, Section 3 describes the architecture of our approach, describing the used architectures of the models of deep learning. Following that, Section 4 compares our different architectures, and then, in Section 4.3, we compare our approach with state-of-the-art approaches. Finally, Section 5 concludes the paper and presents future works.

2 Related Work

The interest in sentiment analysis is growing by the need of knowing the polarity of the opinions published on the Internet. Recent research focuses on aspect-based sentiment analysis. For aspect-based sentiment analysis, there are two main tasks: aspect detection and sentiment detection. For aspect detection, we have two possibilities, to recognize the general aspect, for example, ”Food” or to identify not only the aspect but its sub-aspect, for instance, ”Food, prices.”

Earlier approaches for aspect category detection were based on word frequency [13]. Some recent works use Latent Dirichlet Allocation (LDA) where each topic is characterized by a distribution over words [9] [23]. A different approach is in [3], they propose a modular approach focus on Spanish tweets; it is based on a graph-based algorithm for the general aspect classification and a large number of features and polarity lexicons for sentiment detection.

Supervised methods had been used for this task. Some of the most common classifiers are Support Vector Machine (SVM) [2] [15] [1][18], Maximum Entropy (ME) [11, 18], and Conditional Random Field (CRF) [1] [15]. Some of them use more than one. Other hybrids methods had been proposed [10] using rule based methods with optimization.

Considering the ability to learn useful features from low-level data[14], Deep Learning (DL) has become a popular approach for Aspect-Based Sentiment Analysis [8] [10]. It uses multiple layers to progressively extract higher-level features from the raw input. It allows to capture the correlation between non-consecutive words focusing the attention on the specific significant words [24].

2.1 SemEval 2016 Competition

SemEval competition [19] boosts the research on this area. They publish different tasks to be solved. In the 2016 Edition, the subtask 1 of the task 5 was related to aspect-based sentiment analysis. The Spanish language is known for its complexity. The main competitors in Spanish are described below:

Focus on general aspects, [1] achieves a high performance using Support Vector Machine (SVM) with a list of words with a preprocessing stage using the Freeling tagger and dictionaries.

Focus on both tasks, IIT-TUDA group [15] also uses SVM and combines with several tools such as dependency graphs, distributional thesaurus (DT), scores, and a bag of words; they achieve a better result in each task. Similar to IIT-TUDA, the UWB team [11] uses different approaches to optimize the results of the tasks, and they use a Max Entropy Classifiers as their primary classifier.

TGB team [6] uses binary and multi-class linear classifiers. The INSIGHT-1 team [21] uses a Convolutional Neural Networks (CNN) to obtain a similar score. Conditional Random Fields (CRF) improves the performance of the algorithms when they are used in sub-aspects detection [1][15]. Also, having an excellent preprocessing module is of great importance; some works use taggers, parsers, tokenization, filters, dictionaries, among others.

3 System Architecture

The system architecture includes two Convolutional Neural Network (CNN) models with a previous normalization stage. Figure 1 shows that the reviews are preprocessed. The data cleaning process eliminates and replaces emojis, URLs, and special characters.

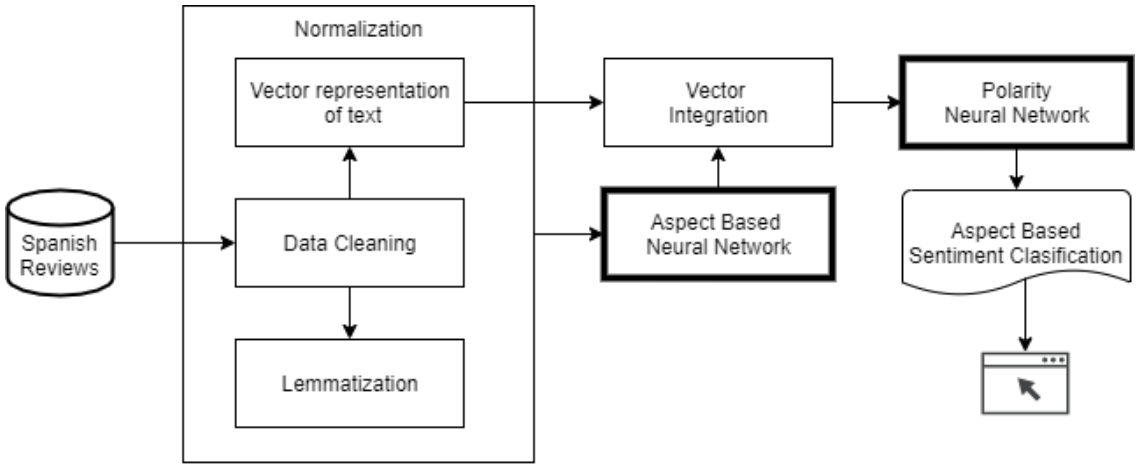


Fig. 1. Simplified block diagram of the system architecture

```
Token: Buen, Lemma: Buen, Norm: buen , Tag: ADJ__Gender=M
Token: servicio, Lemma: servicio, Norm: servicio , Tag: N
Token: ambiente, Lemma: ambientar, Norm: ambiente , Tag:
Token: Acogedor, Lemma: Acogedor, Norm: acogedor , Tag: P
```

Fig. 2. Output of lemmatization process

Once the text is clean, the corresponding word embedding vector is generated using fastText. Also, the cleaned text goes through lemmatization using the spaCy tool, and then it is normalized.

The normalized text is the input of the neural network. The aspect-based neural network model calculates the vector of aspects. The aspect vector combined with the Word Embedding vector are the inputs of the polarity neural network. It produces an aspect-based sentiment classification of restaurant reviews.

It is important to mention that each review has several sentences; then, parallel detection of the classification of aspects and feelings is not recommended. It is because the relation between the polarity and its aspect is lost.

3.1 Normalization

The following processes are essential to prepare the text before processing the reviews into the neural network architecture.

3.1.1 Data Cleaning

First, we remove mentions, URLs, emoticons, and special characters using regular expressions. We keep accented letters and punctuation marks. Because of the unbalanced data sets, the samples of the negative, neutral, and conflict classes were augmented.

3.1.2 Lemmatization

In recent years, the spaCy API [12] has been popular in applications of Natural Language Processing (NLP). We used the model *es_core_we_lg*, which is the largest spaCy model for Spanish; it is pre-trained with texts from the web of general purpose.

The first step is to tokenize the text. Then, a tagger process is used to label the tokens of the previous step according to the part-of-speech. Then, the labels of each token are obtained from the parsing process. Finally, the approach detects and labels the entities of the text.

3.1.3 Vector Representation

The reviews were represented as vectors of real numbers using word embeddings. This approach uses the fastText *Multi-lingual word embeddings or word vectors*. This model supports 157 languages, one of them is Spanish. It was previously trained using Common Crawl (a non-profit organization that crawls the web and provides its files and data sets to the public for free) and Wikipedia (free online encyclopedia).

Figure 2 shows an output of this process.

We iterate between 100 and 300 dimensions; and also include the linguistic components obtained from spaCy to improve the performance of the neural networks. The output of the normalization gives five possible inputs for the deep learning phase:

— **A100**. A word embedding vector of 100 dimensions of lemmas from spaCy and fastText,

— **A300**. A word embedding vector of 300 dimensions of lemmas from spaCy and fastText,

— **B100**. A word embedding vector of 100 dimensions of normalized tokens from spaCy and fastText,

— **B300**. A word embedding vector of 300 dimensions of normalized tokens from spaCy and fastText,

— **C300**. A word embedding vector of 300 dimensions extracted with word2Vec using spaCy.

3.2 Aspect-Based Convolutional Neural Network (AB-CNN)

We defined a Convolutional Neural Networks (CNNs) architecture using Tensorflow and Keras for the aspect classification. We defined two architectures, AB1-CNN with a sequential model and AB2-CNN with multi-channel output. Figure 3 shows both architectures, and they are described as follows.

3.2.1 AB1-CNN

The input layer has a One-Dimensional (1D) Convolutional Neural Network with a kernel of size three, and the activation function Rectified Linear Unit (ReLU). The next layer is a 1D Max Pooling; it has an outstanding performance in conjunction with the convolutional layers. To reduce overfitting, a Dropout layer was placed to disable and activate different neurons during the forward-propagation and backward-propagation processes.

For this layer, approximately 20% of neurons remain deactivated. Then, those three layers are repeated. In addition, the architecture has a Flatten Layer and a Dropout of 50%. Finally, two dense layers were added, with a Dropout in the middle, using different activation functions.

3.2.2 AB2-CNN

It is a model with multi-channel output. There are four 1D Convolutional layers with kernels of 3, 3, 5, and 4; all of them with ReLU activation function. After them, a Max Pooling layer and an LSTM (Long-Short Term Memory) are added. The LSTM is an extension of recurrent neural networks, they expand their memory to learn from previous experiences. The output was n dense layers with sigmoid activation function, all of them are connected to the LSTM.

3.3 Model of the Polarity Neural Network (P-CNN)

The polarity was identified by defining Convolutional Neural Networks (CNNs) using TensorFlow and Keras. We defined three architectures: P1-CNN and P3-CNN are sequential models differing in one layer, and P2-CNN has two channels. Figure 4 shows them, and they are described below.

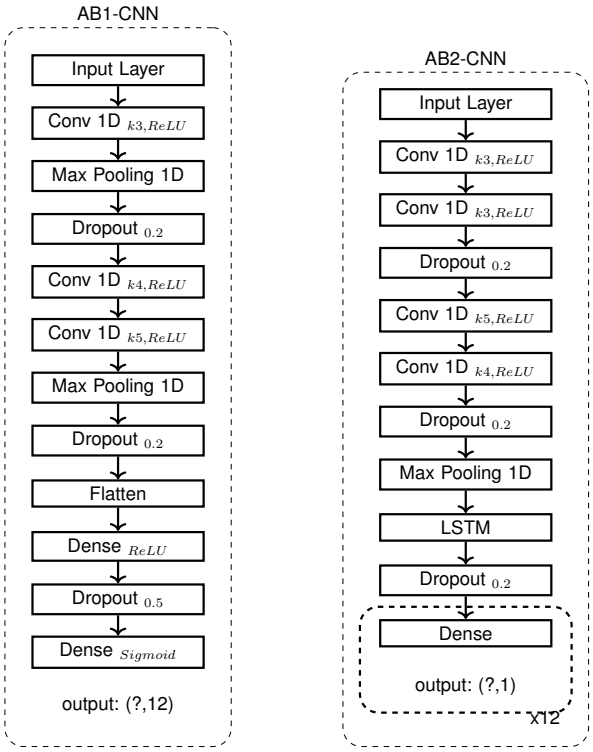


Fig. 3. Two architecture approaches of the Aspect-Based Convolutional Neural Network (AB-CNN)

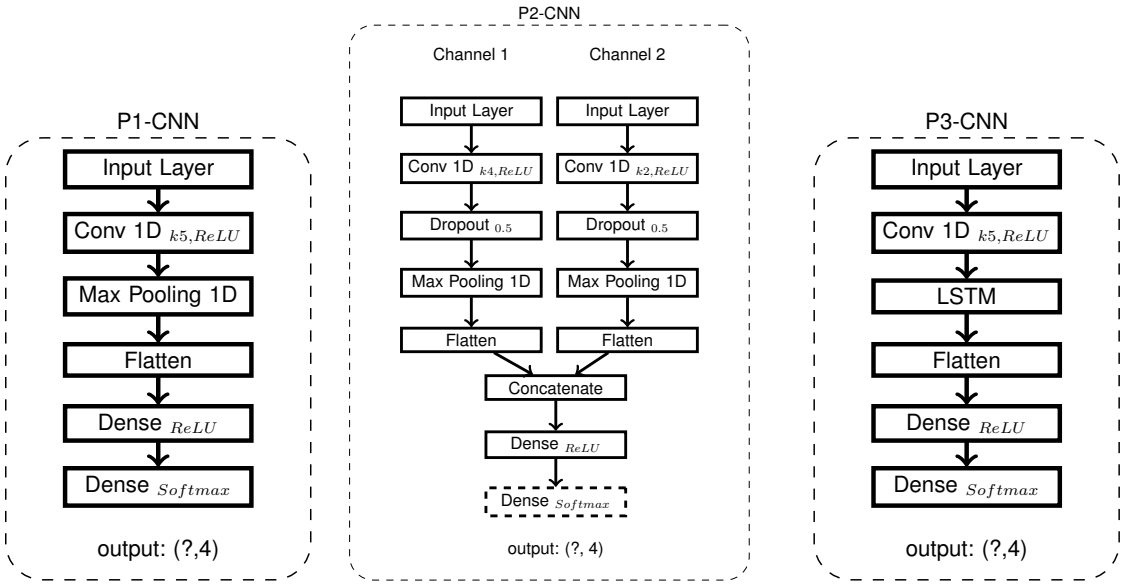


Fig. 4. Three architecture approaches of Polarity Convolutional Neural Network (P-CNN)

3.3.1 P1-CNN

It has a 1D Convolutional Layer with a total of 128 filters and a stride of 5 positions. Then, we used a Max Pooling Layer with a stride of 2 positions to take the most relevant values of the input vector.

Then, there is a Flatten Layer followed by two Dense Layers, the first one with 10 neurons and the activation function of ReLU, and the second one with 4 neurons corresponding to the polarities.

3.3.2 P2-CNN

This architecture has two channels, the first one is related to the vector representation of the text in the normalization phase, and the second one is related to the vector from AB-CNN. Those channels are concatenated, and two Dense Layers are applied.

3.3.3 P3-CNN

The third architecture is similar to the first one. The main difference is in the second layer; instead of the Max Pooling, it has an LSTM layer to expand the memory for learning.

4 Experiments and Validation

The experiments were done in reviews of restaurants given by a data set provided in the 2016 Edition of SemEval competition [19]. The aspect-based sentiment analysis was addressed in subtask 1 of task 5 of the competition. There are three slots in this subtask; this work focuses on slot1 and slot3.

Slot1 refers to the extraction of aspects; those aspects are an Entity **E** and attribute **A** pairs (E#A pair) towards which an opinion is expressed in a text. For example, for the entity *restaurant* and the aspect *price*, the E#A pair is RESTAURANT#PRICE.

Slot3 identifies the sentiment classification. For each pair E#A, will there be a sentiment polarity, such as positive or negative (OTE).

Because of the Spanish language complexity, there were just seven teams in the competition, compared to the 27 teams in the English language. Of those seven, only four of them participated in Slot1 and Slot3, as we present in this research.

4.1 Data Description

SemEval competition provides the training set and the test set. Each review has several sentences, each sentence and each review has several opinions. For each opinion, there is a category and a polarity. The category has an aspect, e.g., *food*, and a subaspect, e.g., *quality* or *price* for the category of *food* (E#A pair).

There are six aspects, namely, *restaurant, service, ambience, food, drinks,* and *location*. Considering the subaspects, we tested for 12 different aspects (see Table 1). Each sentence is classified in four possible polarities:

— **Positive**. When the aspect has positive assessment.

— **Negative**. When the aspect has a negative assessment.

— **Neutral**. When the aspect is not positive nor negative.

— **Conflict**. When the aspect has a positive and negative assessment.

4.1.1 Training Data

The provided training data set includes 627 reviews with 2070 sentences. In Table 1, we show the total of sentences for each aspect, and in Table 2 the number of sentences for each polarity.

Our approach uses this training set for the architecture comparison (Section 4.2). We can see that it is an unbalanced data set. It means that there is a difference between the number of examples belonging to each class, e.g., aspects related to *drinks* are minority classes compared to *service*. It affects the model because the learning system may have difficulties to learn the concept related to the minority class. The model should have a predilection to classify a sentence as positive than conflict. In order to solve it, one can look for specific models [16] or balance by eliminating in the majority classes [5] or augmenting the minority classes, as we do.

Table 1. Number of reviews per aspect in the training data set

Aspect	Total
RESTAURANT#GENERAL	602
SERVICE#GENERAL	389
AMBIENCE#GENERAL	219
FOOD#QUALITY	458
FOOD#PRICES	113
RESTAURANT#PRICES	108
FOOD#STYLE_OPTIONS	134
DRINKS#QUALITY	29
DRINKS#STYLE_OPTIONS	19
RESTAURANT#MISCELLANEOUS	13
LOCATION#GENERAL	15
DRINKS#PRICES	10

Table 2. Number of reviews per polarity in the training data set

Polarity	Total
positive	1512
negative	437
neutral	103
conflict	57

4.1.2 Test Data

The test data set of this subtask [19] includes 268 Spanish restaurant reviews with 881 sentences annotated with {E#A, OTE} tuples at the sentence level. Our approach uses this test set for the architecture comparison (Section 4.2) and the experiments (Section 4.3).

4.2 Architecture Comparison

All architectures were compiled using binary cross-entropy loss function, Adam optimizer, and F1-measure for evaluation.

For the AB-CNN architectures we used the outputs of the normalization phase described in Section 3.1. The data sets are *A300*, *A100*, *B300*, *B100*, and *C300*, where numbers *300* and *100* refer to the number of dimensions. Moreover, data sets with letter *A* correspond to lemmas from spaCy and fastText, data sets with letter *B* correspond to tokens from spaCy and fastText, and data sets with letter *C* use word2Vec representation.

First, we compared the results of the two architectures AB-CNN. We used F1-measure like it is in SemEval challenge [19]. In Table 3, we present two evaluations. **F1 in Training** refers to the output of calling the training method *fit()* which uses a split of the training set to validate, the split is given by *validation_batch_size* with a value of 64. **F1 in Test** corresponds to the evaluation of the model in the Test Data.

Because of the multi-channel output, the AB2-CNN architecture has lower results. Moreover, the unbalance in some classes affects the performance. Then, the selected architecture was AB1-CNN.

For the sentiment classification (slot3) we have three architectures (Section 3.3). In Table 4, we compared with metric accuracy for the training and test data sets. It shows that the P1-CNN architecture has better performance in the training data sets but in the test data set the P2-CNN architecture is better. In both cases, the P3-CNN has the worst results.

In these experiments, we can see that the constructed set A300 has better results in all the aspect-based experiments. The best data set for each experiment is in italics. Additional to the training and test sets, we did field tests with written reviews and manual classification. It considered the 12 classes of aspects of the AB1-CNN architecture and its combination with P1-CNN and P2-CNN. Table 5 shows the results of the field test with a better result than the performed in the test data set. It allows us to identify that the mistaken classification was related to the minority classes, even if we balanced them.

For the task of sentiment analysis of each aspect, the results of the experiments are shown in Table 6. The results of all the architectures are still similar, so it was decided to use the architecture that obtained the best results with the A300 training set since this set was the one with the best results for the aspects classifier model, the best with an accuracy of 93.33%.

Table 3. F1-measure in training and test data sets for AB-CNN architectures

Normalized Data Set	F1 in Training		F1 in Test	
	AB1-CNN	AB2-CNN	AB1-CNN	AB2-CNN
A300	***0.7995***	*0.5618*	***0.6540***	*0.5570*
A100	**0.6640**	0.5448	0.5017	**0.5402**
B300	**0.6562**	0.5497	**0.6273**	0.5457
B100	**0.6530**	0.5399	0.5038	**0.5357**
C300	**0.5945**	0.5112	0.4456	**0.5058**

Table 4. Accuracy in training and test data sets for P-CNN architectures

Normalized Data Set	Accuracy in Training			Accuracy in Test		
	P1-CNN	P2-CNN	P3-CNN	P1-CNN	P2-CNN	P3-CNN
A300	***0.8640***	*0.8382*	0.7937	0.7789	**0.7845**	0.6921
A100	**0.8504**	0.8143	0.8021	0.7553	**0.7879**	0.7093
B300	**0.8494**	0.8372	0.8274	*0.7811*	***0.7969***	*0.7302*
B100	**0.8593**	0.8241	0.8192	0.7520	**0.7699**	0.7192
C300	**0.8427**	0.8023	*0.8294*	0.7710	**0.7789**	0.7203

Table 5. F1-measure in field tests for AB1-CNN architectures

Normalized Data Set	F1
A300	**0.9333**
A100	0.9286
B300	0.9226
B100	0.9198
C300	0.9008

Table 6. Accuracy and F1-measure in field tests for P1-CNN and P2-CNN architectures

Normalized Data Set	Accuracy	
	P1-CNN	P2-CNN
A300	0.9333	0.9000
A100	0.9000	0.9333
B300	0.9333	0.9333
B100	0.9333	0.9333
C300	0.9333	0.9120

4.3 Evaluation

We consider the four teams of the 2016 edition of SemEval that participate in the aspect-based sentiment classification that identifies the aspect and polarity.

Table 7. Results of the aspect extraction

Team	F1
IIT-TUDA	0.5980
INSIGHT-1	0.6137
TGB	0.6355
UWB	0.6196
AB1-CNN	**0.6540**

Table 7 shows the results of those fours competitors and our approach AB1-CNN. We can see that in the aspect extraction, we got better results than other competitors.

Table 8 shows the results of the same fours competitors and our approach P-CNN with the data set of B300. We did not achieve a better result than most of the competitors in the accuracy metric.

The accuracy measures all the correctly identified polarities, but it is useful when all the classes are equally important.

Although all classes are essential, the training and test data set do not have the same number of samples per class. The positive class represents 71.69% of the complete data set, and the remaining 28.3% is distributed in the other three classes.

Table 8. Results of the sentiment classification

Team	Accuracy
IIT-TUDA	**0.8358**
INSIGHT-1	0.7957
TGB	0.8209
UWB	0.8134
AB1-CNN	0.7969

5 Conclusions and Future Work

Aspect-based sentiment analysis is a major technological challenge. Our proposal focuses on processing, analyzing, and categorizing reviews and was tested in restaurant reviews.

Our approach combines two deep learning models for aspect-based sentiment analysis. The performance of our aspect classification proposal is promising, but there is still work to do in the sentiment detection.

The preprocessing stage was a core part of the proposal. The text was represented as word vectors of real numbers; it allows us to avoid losing context and relate the polarities to each aspect, even if there were more than one aspect in a sentence. We used lemmas and tokens, but there is also the possibility of using n-grams.

The proposal combines two deep learning models architectures to determine the sub-aspect and the polarity for the classification task. We develop two architectures for the aspect detection and three architectures for the sentiment analysis to get our final architecture merging the preprocessing with the deep learning models.

We got better results than state-of-the-art models in aspect classification but not as good in polarity classification. It was affected by the unbalanced data set and by the CNN, even when we try three different architectures. Other works that also used CNN, such as INSIGHT-1, were also affected in the polarity evaluation.

There is still room for improvement; for instance, including dictionaries in the preprocessing stage to replace core words.

References

1. **Alvarez-López, T., Juncal-Martínez, J., Fernández-Gavilanes, M., Costa-Montenegro, E., González-Castano, F. J. (2016).** Gti at Semeval-2016 task 5: Svm and crf for aspect detection and unsupervised aspect-based sentiment analysis. Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pp. 306–311.
2. **Ananiadou, S., Rea, B., Okazaki, N., Procter, R., Thomas, J. (2009).** Supporting systematic reviews using text mining. Social Science Computer Review, Vol. 27, No. 4, pp. 509–523.
3. **Araque, O., Corcuera, I., Román, C., Iglesias, C. A., Sanchez-Rada, J. F. (2015).** Aspect based sentiment analysis of Spanish tweets. TASS@ SEPLN, pp. 29–34.
4. **Ayuve (2019).** La importancia de obtener buenas reseñas en Google. `https://www.ayuve.net/blog/la-importancia-de-las-resenas-en-google/`. Accessed: March 24, 2021.
5. **Batista, G. E., Carvalho, A. C., Monard, M. C. (2000).** Applying one-sided selection to unbalanced datasets. Mexican International Conference on Artificial Intelligence, Springer, pp. 315–325.
6. **Çetin, F. S., Yıldırım, E., Özbey, C., Eryiğit, G. (2016).** TGB at Semeval-2016 task 5: multi-lingual constraint system for aspect based sentiment analysis. Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 337–341.
7. **Cruz, I., Gelbukh, A., Sidorov, G. (2014).** Implicit aspect indicator extraction for aspect based opinion mining. Int. J. Comput. Linguistics Appl., Vol. 5, No. 2, pp. 135–152.
8. **Do, H. H., Prasad, P., Maag, A., Alsadoon, A. (2019).** Deep learning for aspect-based sentiment analysis: a comparative review. Expert Systems with Applications, Vol. 118, pp. 272–299.
9. **García-Pablos, A., Cuadros, M., Rigau, G. (2018).** W2VLDA: almost unsupervised system for aspect based sentiment analysis. Expert Systems with Applications, Vol. 91, pp. 127–137.
10. **Goldberg, Y. (2017).** Neural network methods for natural language processing. Synthesis lectures on human language technologies, Vol. 10, No. 1, pp. 1–309.

11. **Hercig, T., Brychcín, T., Svoboda, L., Konkol, M. (2016).** Uwb at Semeval-2016 task 5: Aspect based sentiment analysis. Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pp. 342–349.
12. **Honnibal, M., Montani, I. (2017).** spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.
13. **Hu, M., Liu, B. (2004).** Mining and summarizing customer reviews. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 168–177.
14. **Kelleher, J. D. (2019).** Deep learning. MIT press.
15. **Kumar, A., Kohail, S., Kumar, A., Ekbal, A., Biemann, C. (2016).** Iit-tuda at Semeval-2016 task 5: Beyond sentiment lexicon: Combining domain dependency and distributional semantics features for aspect based sentiment analysis. Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pp. 1129–1135.
16. **Li, S., Song, W., Qin, H., Hao, A. (2018).** Deep variance network: An iterative, improved CNN framework for unbalanced training datasets. Pattern Recognition, Vol. 81, pp. 294–308.
17. **Miranda, C. H., Buelvas, E. (2019).** AspectSA: Unsupervised system for aspect based sentiment analysis in Spanish. Prospectiva, Vol. 17, No. 1, pp. 87–95.
18. **Pannala, N. U., Nawarathna, C. P., Jayakody, J., Rupasinghe, L., Krishnadeva, K. (2016).** Supervised learning based approach to aspect based sentiment analysis. 2016 IEEE International Conference on Computer and Information Technology (CIT), IEEE, pp. 662–666.
19. **Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., others (2016).** Semeval-2016 task 5: Aspect based sentiment analysis. International workshop on semantic evaluation, pp. 19–30.
20. **Román, J. V., Cámara, E. M., Morera, J. G., Zafra, S. M. J. (2015).** Tass 2014-the challenge of aspect-based sentiment analysis. Procesamiento del Lenguaje Natural, Vol. 54, pp. 61–68.
21. **Ruder, S., Ghaffari, P., Breslin, J. G. (2016).** Insight-1 at Semeval-2016 task 5: Deep learning for multilingual aspect-based sentiment analysis. arXiv preprint arXiv:1609.02748.
22. **WebParaRestaurantes (2017).** La importancia de las opiniones online de clientes para tu restaurante. `http://webpararestaurantes.com/resenas-clientes-restaurante/`. Accessed: August 16, 2019.
23. **Weichselbraun, A., Gindl, S., Fischer, F., Vakulenko, S., Scharl, A. (2017).** Aspect-based extraction and analysis of affective knowledge from social media streams. IEEE Intelligent Systems, Vol. 32, No. 3, pp. 80–88.
24. **Zuheros, C., Martínez-Cámara, E., Herrera-Viedma, E., Herrera, F. (2021).** Sentiment analysis based multi-person multi-criteria decision making methodology using natural language processing and deep learning for smarter decision aid. case study of restaurant choice using tripadvisor reviews. Information Fusion, Vol. 68, pp. 22–36.

Article received on 25/06/2021; accepted on 15/11/2021.
Corresponding author is Sabino Miranda.

ISSN 2007-9737

Deep Learning and Feature Extraction for Covid 19 Diagnosis

Nadir Berrouane[1], Mohammed Benyettou[1], Benchennane Ibtissam[2]

[1] University Ahmed Zabana,
Mathematics and Informatics,
Algeria

[2] Ecole nationale polytechnique d'Oran-Maurice Audin (ENPO-MA),
Algeria

nadir.berrouane@univ-relizane.dz

Abstract. Recently, medical images analysis is becoming the center of interest in the medical field, with the helpful opportunities offered by artificial intelligence, especially deep learning techniques. Computers are becoming more and more capable of learning how to be diagnosing certain medical pathologies and diseases. In this domain, deep learning is a major choice, more precisely Convolutional Neural Networks (CNN) due to its powerful performance with images classification. In this paper, a new approach is proposed which is about using feature extraction from images and deep learning algorithms to avoid the issue of the necessity of a large dataset. This work aims to improve the diagnostic of the Covid 19 virus in X-ray images, by extracting the features and applying the deep learning algorithm. This approach is composed of two main phases. The first one is based on feature extraction from images using feature extraction algorithms: Pyramid Histogram of Gradient (PHOG), Fourier, Gabor, and Discrete Cosine Transform (DCT). The second phase is based on using the last layers of CNN of deep learning for the classification problem. The experimentation of our approach is demonstrated by utilizing chest X-ray images obtained by PyImageSearch. Analysis of results shows that the proposed approach provides a satisfactory result. Our approach could be so beneficial in the future that it can be used to solve real-life problems even though insufficient data especially in urgent cases where there is not enough time to collect the data.

Keywords. Classification, feature extraction, deep learning.

1 Introduction

Recently, a new disease called COVID 19 (Coronavirus) appeared in December 2019 in Wuhan, the capital of Hubei, China. COVID 19 spreads near contaminated surfaces with an age ranging from several hours to several days depending on the nature of the surface, it spreads as well by coughing or sneezing (the virus can be transmitted to another person through saliva droplets). This virus has several symptoms such as fever, cough, tiredness and in more advanced stages can lead to difficulty in breathing medically referred to as dyspnea.

On March 11th, 2020, the World Health Organization (WHO) declared a pandemic cannot be controlled. According to the Worldometer website, there have been 259,645,518 cases touched by the virus and 5,190,691 deaths. In this case, if we want to prevent or at least reduce the spread of this disease, we must try something that can speed up the diagnosis. So, the idea is to find out if an individual is infected with the coronavirus in the early stages that it is easier to deal with and the contagion can be stopped. Among the used solutions is the intercalation of Deep learning in the medical field.

Deep Learning precisely convolutional neural networks (CNN), has rapidly become the method of choice for the analysis of radiological images. In general, the convolutional neural network process includes the feature extraction phase, yet it requires a huge amount of input images for the network to be capable of learning. Providing data needs a lot of time but as is mentioned previously the more time we take, the more the epidemic spreads. Several works have been done in deep learning for medical diagnosis as discussed below.

In [1], the authors proposed a combination of deep learning, natural language processing, and medical imaging to improve medical diagnosis. This work is a survey of deep learning in medical diagnosis. In [2] the authors present a survey of the therapeutic areas and deep learning models for diagnosis. In [3] the authors present an overview of the deep learning approach for COVID 19 diagnosis. In [4], Medjahed et al proposed a new approach for COVID-19 diagnosis based on feature selection and meta-heuristic called Multi-Verses Optimizer.

In our approach, the main idea consists of combining feature extraction and deep learning to enhance the quality of medical diagnosis and to give a better performance. We will introduce the two main keys of Deep Learning and feature extraction. The first one is the representation of an image as a vector of features, among the methods of feature extraction, we cite Pyramid Histogram of Gradient, Local binary patterns, Color histograms, Fourier, Gabor, Discrete cosine transform, etc. In our work, we propose to use four of the most relevant feature extraction methods to extract the features of the image dataset: Pyramid Histogram of Gradient, Fourier, Gabor, and Discrete cosine transform. Secondly, we train CNN with the data gained from the first step by introducing the features extracted.

The experiment is conducted on x-ray images of people infected with the Corona epidemic and others who are not infected.

The rest of this paper is organized as follows. Section 2 presents the state of the art of feature extraction and deep learning. Section 3 illustrates the proposed approach. Section 4 presents the experimental results. Section 5 draws some perspective.

2 State of the Art

2.1 Feature Extraction

In this section, we have focused on four feature extraction methods used in our work:

Histogram of The Pyramid Orientation Gradients (PHOG), divides the image into sub-regions that have different resolutions, it is generally used for object detection [4, 7].

Histograms of oriented gradients are feature descriptors used for object detection. It was first introduced by Navneet Dalal and Bill Triggs, researchers for the French National Institute for Research in Computer Science and Control (INRIA), [10].

The technique works by counting the occurrence of gradient orientation computed on a dense grid of uniformly spaced cells on an image. The idea behind this algorithm is that the local appearance of objects in an image can be described using the distribution of edge directions. The HOG descriptor is, in particular, useful for pedestrian detection [11].

Pyramid histogram of gradients (PHOG) is an extension to HOG features. Extending HOG to PHOG is by analogy very similar to the extension of HOW (histogram of visual words) to PHOW. In PHOG, the spatial layout of the image is preserved by dividing the image into sub-regions at multiple resolutions and applying the HOG descriptor in each sub-region.

To program the PHOG features, the Canny edge detector is usually applied on grayscale images, then a spatial pyramid is created with four levels [12]. The histogram of oriented gradients is then calculated for all bins in each level. All histograms are then concatenated to create the PHOG representation of the input image.

Fourier Functions

Fourier function is widely used in image processing. It is divided into sine and cosine components. The number of pixels in the image represents the number of frequencies [4, 7].

Fourier transform is a mathematical function that decomposes a waveform, which is a function of time, into the frequencies that make it up.

The result produced by Fourier transform is a complex-valued function of frequency. The absolute value of the Fourier transform represents the frequency value present in the original function and its complex argument represents the phase offset of the basic sinusoidal in that frequency.

Fourier transform is also called a generalization of the Fourier series. This term can also be applied to both the frequency domain representation and the mathematical function used.

Fourier transform helps in extending the Fourier series to non-periodic functions, which allows viewing any function as a sum of simple sinusoids.

Gabor Feature

This method combines the characteristics of scale, spatial location, and orientation to recognize a region [4, 7].

This feature relies on using Gabor filters for character recognition in gray-scale images is proposed in this paper. Features are extracted directly from gray-scale character images by Gabor filters which are specially designed from statistical information of character structures. An adaptive sigmoid function is applied to the outputs of Gabor filters to achieve better performance on low-quality images. To improve the discriminability of the extracted features, the positive and the negative real parts of the outputs from the Gabor filters are used separately to construct histogram features.

Experiments show us that the proposed method has excellent performance on both low-quality machine-printed character recognition and cursive handwritten character recognition.

Discrete Cosine Transforms (DCT)

DCT divides the image depending on the visual into sub-blocks of different importance [4, 7].

The discrete cosine transform (DCT) is a real transformation that has great advantages in energy compaction. Its definition for spectral components DP u,v is:

$$DP_{u,v} = \begin{cases} \frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} P_{x,y} \ \ if\ u=0\ and\ y=0 \\ \frac{2}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} P_{x,y} \times \cos\left(\frac{(2x+1)u\pi}{2N}\right) \times \cos\left(\frac{(2y+1)v\pi}{2V}\right). \end{cases} \quad (1)$$

There are many variants of the definition of the DCT, and we are concerned only with principles here. The inverse DCT is defined by:

$$P_{u,v} = \frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} DP_{u,v} \times \cos\left(\frac{(2x+1)u\pi}{2N}\right) \times \cos\left(\frac{(2X+1)v\pi}{2N}\right). \quad (2)$$

A fast version of the DCT is available, like Fourier Functions Transform (FFT), and calculation can be based on the FFT. Both implementations offer about the same speed. The Fourier transform is not optimal for image coding since the DCT can give a higher compression rate, for the same image quality. This is because the cosine basis functions can afford high-energy compaction.

2.2 Deep Learning

Deep learning is a sub-domain of machine learning, it concerns algorithms inspired by the structure and function of the human brain. These algorithms are called artificial neural networks (ANNs). Deep learning consists of neural networks with a large number of layers and parameters. There are three fundamental network architectures: artificial neural networks (ANNs), recurrent neural networks (RNN), recursive neural networks, and convolutional neural networks (CNN). The automatic feature extraction is one of the main facets, indeed, summarizing this step to a simple raw image introduction seems like one of the great advantages of deep learning [8].

Activation Function

It matches the inputs of a node to its corresponding output, e.g., Sigmoid, Tanh, ReLU, etc. These functions are constructed using different mathematical techniques. There are several types of activation functions, but the most popular activation function is the rectified linear unit function, also known as the ReLU function. It is well-known to be a better activation function than the sigmoid function and the Tanh function because it performs the descent of the slope faster. Indeed, in the sigmoid and Tanh function when the input (x) is very large, the slope is very small, which slows down the descent of the gradient considerably [8].

Cost Function

Similar to any other machine learning model, it measures the "quality" of a neural network in relation to the values it predicts in relation to the actual values. The cost function is inversely proportional to the quality of a model - the better the model, the lower the cost function. In other

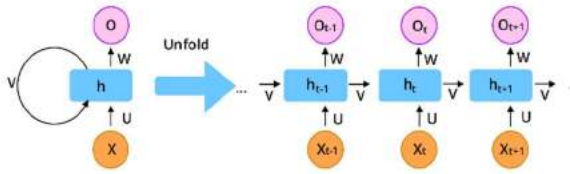


Fig. 1. RNN SCHEMA, the image provided via Wikimedia Commons

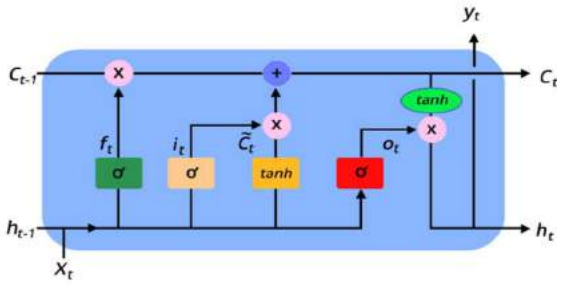


Fig. 2. Long short-term memory neural network, Image provided via improving long-horizon forecasts with expectation-biased LSTM networks

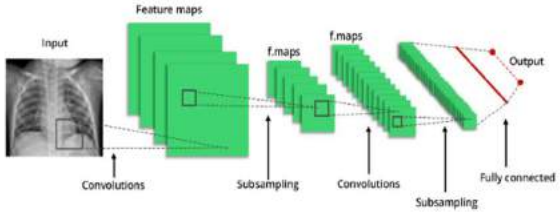


Fig. 3. Convolutional neural network, an image inspired from Wikimedia commons

words, the more the cost function is minimized, the more the weights obtained and the parameters are optimal for the model, resulting in a powerful model.

There are several commonly used cost functions, including quadratic cost, cross-entropy cost, exponential cost, Hellinger distance, Kullback-Leibler divergence [8].

Back Propagation (BP)

BP algorithm is a method to monitor learning. It utilizes the methods of mean square error and gradient descent to realize the modification to the connection weight of the network.

The modification to the connection weight of the network is aimed at achieving the minimum error sum of squares. In this algorithm, a little value is given to the connection value of the network first, and then, a training sample is selected to calculate the gradient of error relative to this sample [9].

2.3 Fundamentals Network Architectures

In this section, we have focused on the three basic network architectures known in deep learning and have briefly explained their principles:

Recurrent Neural Networks

A Recurrent Neural Network (RNN) is known for its ability to ingest inputs of varying sizes. They take into account both the current input and the previous inputs given to it, meaning that the same input can technically produce a different output based on the previous input data. In RNNs the connections between nodes form a digraph along a time sequence, allowing them to use their internal memory to process sequences of inputs of variable length.

RNNs are a type of neural network that is mainly used for sequential data or time series [8].

Long-term and Short-term Memory Networks (LSTM)

Created to fill one of the gaps in ordinary RNNs, they have a short-term memory. Specifically, if a sequence is too long, i.e., if there is a time lag of more than 5-10 steps, LSTMs tend to reject information that has been provided in previous steps. The LSTMs has therefore been created to solve this problem of Vanishing gradient [8].

Convolutional Neural Networks

A convolutional neural network (CNN) is a type of neural network that takes an input (usually an image), assigns importance to different features in the image, and produces a prediction.

What makes CNN's better than forward neural networks (FNN), they are better at capturing spatial dependencies (pixels) throughout the image, which means they can better understand the composition of an image.

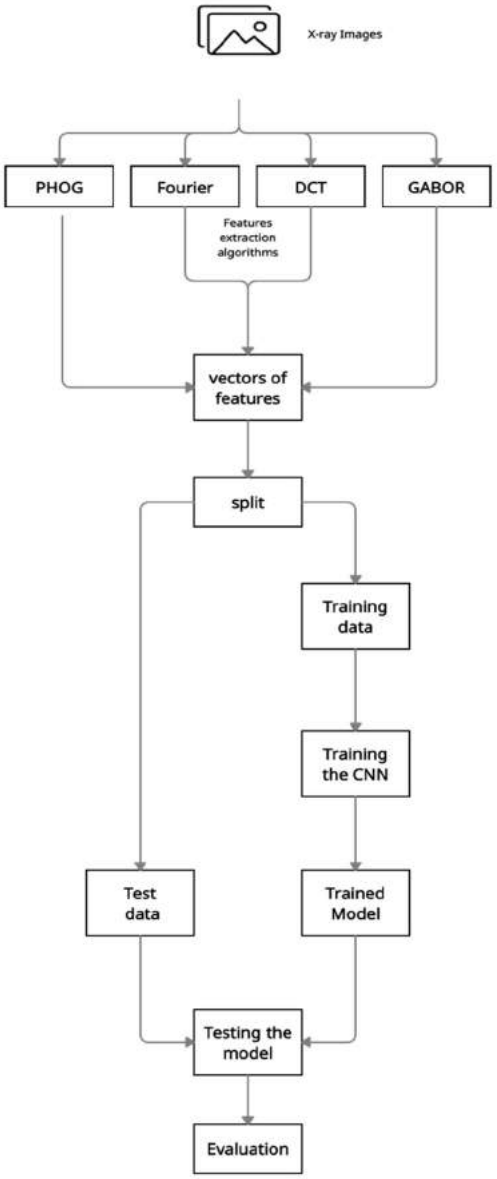


Fig. 4. Proposed approach framework

CNN's uses a mathematical operation called convolution. In the literature, convolution is defined as a mathematical operation on two functions that produces a third function expressing how the shape of one is changed by the other.

This convolution is used by CNNs instead of the matrix multiplication in at least one of their layers. CNN's are mainly used for image classification [8].

3 Proposed Approach

Based on what we have mentioned in the previous section, we have decided to use CNN reason of its effective results on a dataset containing images specially on classification problems.

The approach is divided into four phases (Fig. 4):

- Data preprocessing phase.
- Model training phase.
- Model testing phase.
- Model evaluation phase.

3.1 Data Preprocessing Phase

First of all, the data we are going to utilize must be well prepared for the training phase, to be so, many functions will be applied to this data, and these functions are the following:

- Reading the grayscale images from two folders, each folder contains 25 images (images are in black-and-white color).
- Adding the label of each image in a [0-1] Data-frame, the value is 1 if the person has Coronavirus and 0 if he is not having the virus (Target Data-frame).
- Applying image features extraction algorithms on each uploaded image (PHOG, DCT, FOURIER, GABOR). Each algorithm of the four mentioned algorithms takes an image as input and its output is a vector.
- Concatenating all the produced vectors to one single data frame, each row of this Data-frame is a vector.
- At this level, we obtain two Data-frame, the target Data-frame, and the new converted Data-frame.
- Concatenating these two Data-frame to one dataset.
- Choosing randomly 70% of data for the training process assuring that this 70% has 50% of each label target and the left 30% for the test phase.
- Finally, splitting the training dataset into X_train and Y_train and the test dataset into X_test and Y_test.

3.2 Model Training Phase

This phase consists of using the CNN model for the training model with the 70% dataset from phase A. the model architect is defined to many layers as below:

- The first layer is the input layer used to read and normalize the data.
- The second layer multiplies input data by weight and adds a bias vector.
- The Batch normalization layer is applied to allow every layer of the network to do learning more independently.
- The fourth layer uses the rectified linear unit.
- The Fifth layer multiplies the data by weight and adds a bias vector, in this layer using the SoftMax activation function.
- The last layer is the classification layer that produces given outputs (0 or 1).

The phases A and B are illustrated in Fig. 5.

3.2 Model Testing Phase

This phase consists of testing the trained model from phase B utilizing a 30% dataset from phase A.

3.3 Model Evaluation Phase

This phase is about evaluating the model using classification metrics such as Confusion Matrix, Accuracy, Precision, Sensitivity, and Specificity.

These metrics help us to be able to compare the results of different classification models (Our approach, SVM, KNN, NB)

We will show in section 4 that our approach gives the best results.

4 Experimental Results

In this section, we present the experimental results obtained by the proposed approach and compare them to several classification methods.

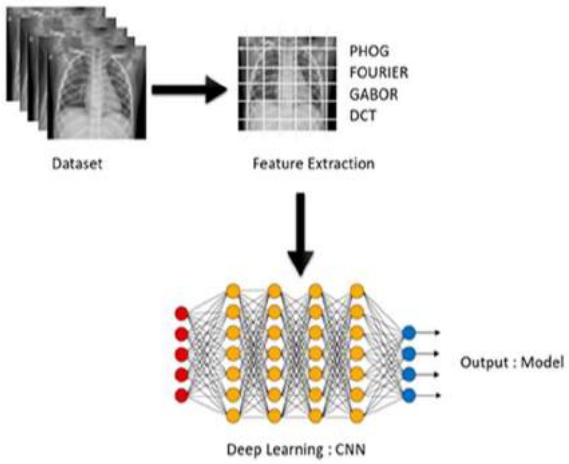


Fig. 5. The proposed approach

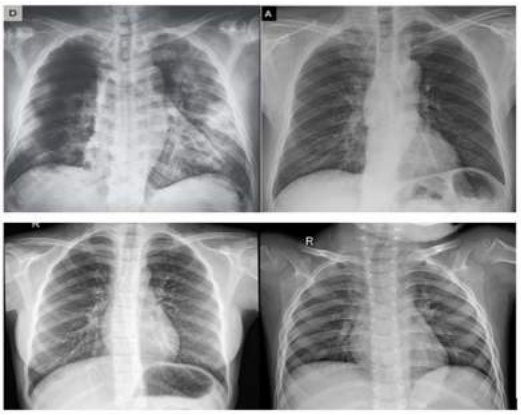


Fig. 6. The images used in this work

4.1 Dataset Collection

The used dataset in this experience is collected by Adrian Rosebrock and it is available on Pyimageseach website[1].

The data is composed of 50 images of chest X-rays and it is divided into two categories: 25 images of healthy people and the rest are those who have Covid19 [5], [6].

The images used in this work are illustrated in Fig. 6.

The figure shows the images used for experimentation. The first row is a normal image and the second row shows a Covid 19 image.

[1] www.pyimageseach.com

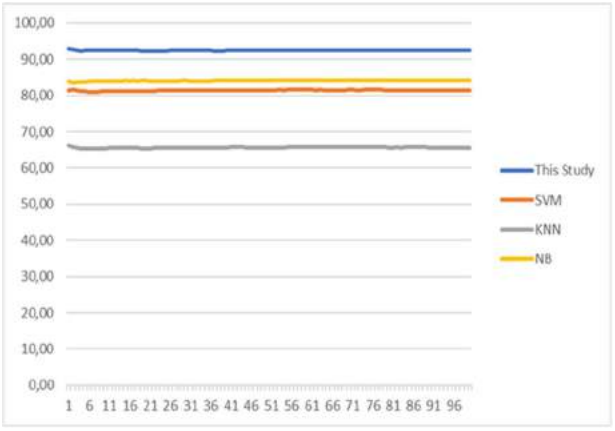


Fig. 7. The results obtained by all the approaches versus the number of executions

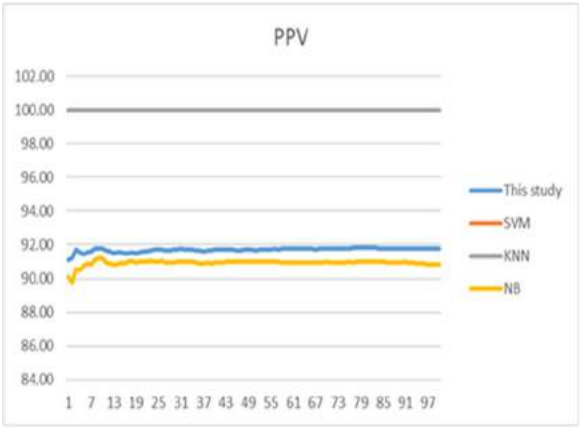


Fig. 8. The PPV obtained by all the approaches versus the number of executions

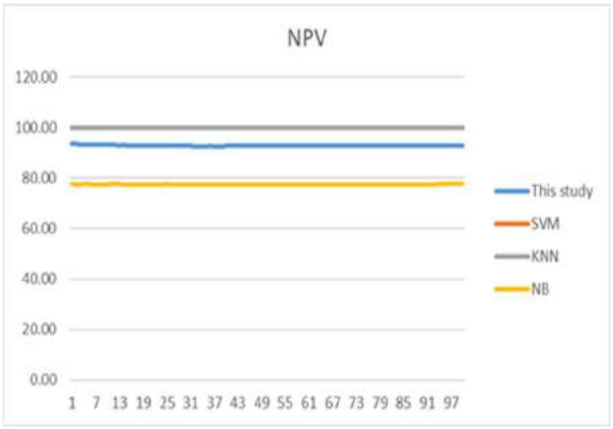


Fig. 9. The NPV obtained by all the approaches versus the number of executions

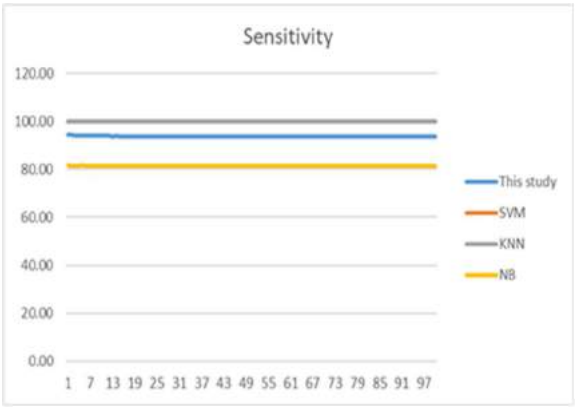


Fig. 10. The Sensitivity obtained by all the approaches versus the number of executions

4.2 Dataset Collection

Generally, in deep learning, it is common knowledge that too little training dataset results in a poor approximation, underfit the model, and poor performance but our approach demonstrated that it can train a model with a small dataset.

The proposed approach is compared with other machine learning classification algorithms using accuracy metrics.

Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Naïve Bayes (NB) are used with the same training, test datasets that we used in our approach.

Table 1 illustrates the results of this study compared to other methods.

Classification accuracy is reported in Table 1. The second column presents the results obtained by the proposed approach

FE-DL, the third column represents the results obtained by SVM, the fourth column is the results obtained by KNN and the last column presents the results obtained by Naïve Bayes.

We have run 100 times, each time containing 100 iterations for all the models (FE-DL, SVM, KNN, NB) and we have recorded the worst, the average, and the best accuracy values. We have also calculated the standard deviation in order to

Table 1. The results of this study compared to other methods

	Proposed method (FE-DL)	Support Vector Machine (SVM)	K-nearest neighbor (KNN)	Naïve Bayes (NB)
Worst (%)	92.24	80.88	65.21	83.60
Best (%)	92.95	81.68	66.25	84.26
Average (%)	92.43	81.43	65.58	84.12
Standard Deviation	± 0.074	± 0.14	± 0.13	± 0.11

Table 2. Definition of the positive test and negative test

	Sick patients	Not-sick patient
Positive test	True Positive	False Positive
Negative test	False Negative	True Negative

be able to see if the model's training is stable or not.

As we have seen in Table 1, the proposed approach produced a high classification accuracy rate competed to other approaches. We note 92.43% of the average classification accuracy rate. The best value is 92.95% and the worst value is 92.24%.

Naïve Bayes (NB) has provided a good result, the average is 84.12%, the worst is 83.60% and the best is 84.26%. We record for SVM 81.68% for the best classification accuracy rate, 80.88%, for the worst, and 81.43% for the average value. KNN has produced no satisfactory results, the average is 65.62%, the worst is 65.21% and the best is 66.25%.

The best value of standard deviation is noted for FE-DL and SVM approaches. Fig. 7 describes the results obtained by all the approaches versus the number of executions.

We clearly remark that the proposed approach is very stable, even if the training and testing set changed.

In order to outperform the stability and the performance of the proposed approach, we calculated the negative predictive value (NPV) and the positive predictive value (PPV), also the sensitivity and specificity. Fig. 8, 9, 10, and 11 illustrate the last values.

Sensitivity called also "Selectivity" and Specificity are two important parameters used for medical diagnosis. Sensitivity measures the ability to give positive results when the instance is verified. Specificity is opposed to sensitivity, it measures the ability to give negative results when the instance is not verified.

Sensitivity and Specificity can be seen as probability and a rate of a dataset.

The analysis of the obtained results shows that the proposed approach is efficient. We remark a 92.15% minimum classification rate and 95.55% as maximum classification rate over the 100 run times. In this case, we can say that the proposed approach is more stable. For SVM we note an 81.47% for the minimum and 82.12% as a maximum. The worst results are obtained by KNN with a 64.20% minimum of classification rate and 64.41% as maximum.

Fig. 8, 9, 10, and 11 show the PPN, NPV, Sensitivity, and Specificity obtained by the proposed approach and compared to SVM, KNN, and NB for all execution times. The proposed approach has provided a satisfactory result compared to the other approaches. We note that

Table 3. Maximum, average, and minimum values of specificity

	Proposed method (FE-DL)	**Support Vector Machine (SVM)**	**K-nearest neighbor (KNN)**	**Naïve Bayes (NB)**
Worst(%)	92.15	81.47	64.20	83.68
Best(%)	92.33	81.58	65.27	84.19
Average (%)	92.55	82.12	65.41	84.36

Table 4. Maximum, average, and minimum values of classification accuracy rate

Classification Accuracy rate (%)			
	Worst	**Best**	**Average**
Proposed Approach	92.24	92.95	92.43
Proposed Approach with PHOG	91.10	92.05	91.42
Proposed Approach with Fourier	81.65	83.17	81.83
Proposed Approach with GABOR	89.62	89.96	89.77
Proposed Approach with DCT	77.02	77.99	97.22
VGG16	93.01	93.52	93.29

Sensitivity is the percentage of true positive and Specificity is the percentage of the true negative. PPV and NPV are used to determine the likelihood of a diagnostic test.

To analyze the performance of the proposed architecture in-depth, we propose to compare each feature extraction approach without combination.

We run the algorithm 100 times using PHOG, Fourier, GABOR, and DCT, and we record the minimum, average, and maximum classification accuracy rates. In addition, the proposed approach is compared to VGG16, which is a convolution neural network (CNN) considered as the best model architecture for deep learning. VGG16 was proposed by K. Simonyan and A. Zisserman [15].

Table 4 and Figure 13 illustrate the obtained results. Table 4 and figure 13 show the worst, best, and average classification accuracy rate obtained by the proposed approach and compared to each feature extraction approach and VGG16.

The analysis of results shows that the proposed approach provides satisfactory results compared to others.

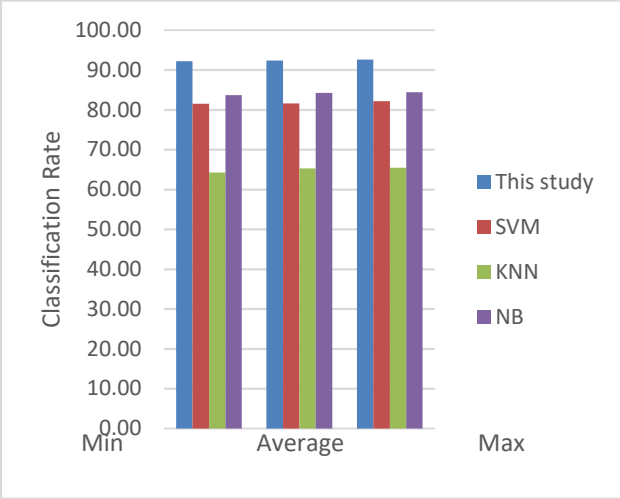


Fig. 11. Maximum, average, and minimum values of specificity

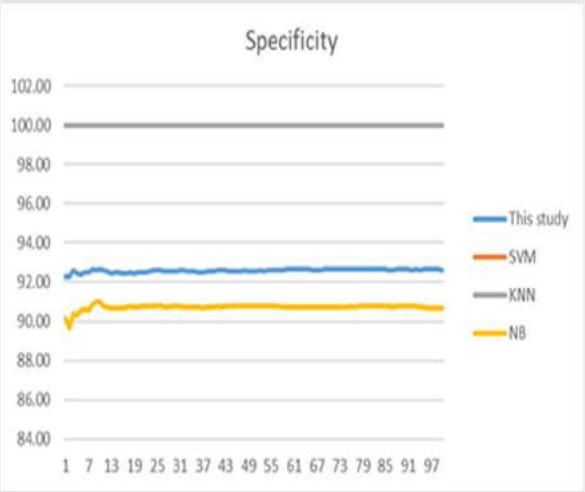


Fig. 12. The Specificity obtained by all the approaches versus the number of executions

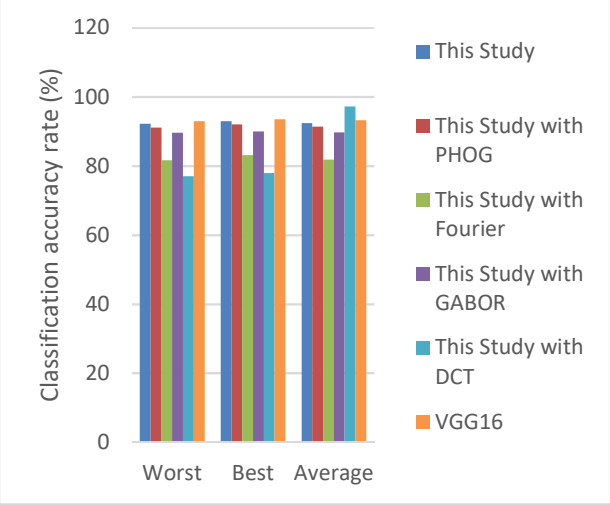


Fig. 13. Classification accuracy rate obtained by the proposed approach compared to each feature extraction method and VGG16

The best result is recorded for VGG16 with 93.52% of classification accuracy and compared to the proposed approach, which provides 92.95% of classification accuracy rate. VGG16 is slightly higher than the proposed approach.

The worst results are obtained using DCT. In addition, PHOG produces a high classification accuracy rate compared to Fourier, Gabor, and DCT.

As future work, we can combine VGG16 with the proposed approach to improve the image classification accuracy.

5 Conclusion

These last years, deep learning has been a very interesting method and active research in many

fields. In this paper, we proposed a hybrid approach based on two phases. The first one is the extraction of features using PHOG, Fourier, Gabor, and DCT. The second phase consists of using deep learning to classify the images. The proposed approach is trained and tested on the X-ray images of Covid 19.

The experimental results demonstrate the performance of the proposed approach. The proposed approach was compared to SVM, KNN, and NB. The results show that the proposed approach FE-DL outperforms compared to the others. Our approach could be so beneficial for further future that it can be used to solve real-life problems even though insufficient data especially in urgent cases where there is not enough time to collect the data for instance Covid 19 virus.

References

1. **Pandey, B., Pandey, D.K., Mishra, B.P., Rhmann, W. (2021).** A Comprehensive Survey of Deep Learning in the field of Medical Imaging and Medical Natural Language Processing: Challenges and research directions. Journal of King Saud University - Computer and Information Sciences. DOI: 10.1016/j.jksuci.2021.01.007.
2. **Nogales, A., García-Tejedor, Á.J., Monge, D., Vara, J.S., Antón, C. (2021).** A survey of deep learning models in medical therapeutic areas. Artificial Intelligence in Medicine, Vol. 112, pp. 1–17. DOI: 10.1016/j.artmed.2021.102020.
3. **Bhattacharya, S., Maddikunta, P.K.R., Pham, Q.V., Gadekallu, T.R., Krishnan-S, S.R., Chowdhary, C.L., Alazab, M., Piran, J. (2021).** Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey, Vol. 65, pp. 1–18. DOI: 10.1016/j.scs.2020.102589.
4. **Medjahed, S.A., Ouali, M. (2020).** Automatic system for COVID-19 diagnosis. Computación y Sistemas, Vol. 24, No. 3, pp. 1131–1138. DOI: 10.13053/CyS-24-3-3366.
5. **Rosebrock, A. (2017).** Deep Learning for Computer Vision with Python, Practitioner Bundle. PyImageSearch.
6. **Rosebrock, A. (2017).** Face Alignment with OpenCV and Python. PyImageSearch.
7. **Medjahed, S.A. (2015).** A Comparative Study of Feature Extraction Methods in Images Classification. International Journal Image, Graphics and Signal Processing, Vol. 3, pp. 16–23. DOI: 10.5815/ijigsp.2015.03.03.
8. **Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van-der-Laak, J.A.W.M., Ginneken, B. -v., Sánchez, C.I. (2017).** A Survey on Deep Learning in Medical Image Analysis. Medical Image Analysis, Vol. 42, pp. 60–88. DOI: 10.1016/j.media.2017.07.005.
9. **Li, J., Cheng, J.H., Shi, J.Y., Huang, F. (2012).** Brief introduction of the backpropagation (BP) neural network algorithm and its improvement. **Jin, D., Lin, S., eds.,** Advances in Computer Science and Information Engineering, Advances in Intelligent and Soft Computing, Vol. 169, pp. 553–558. DOI: 10.1007/978-3-642-30223-7_87.
10. **Dalal, N., Triggs, B. (2005).** Histograms of oriented gradients for human detection. IEEE computer society conference on computer vision and pattern recognition (CVPR'05), Vol. 1, pp. 886–893. DOI: 10.1109/CVPR.2005.177.
11. **El Ansari, M., Lahmyed, R., Tremeau, A. (2018).** A Hybrid Pedestrian Detection System based on Visible Images and LIDAR Data. 13th International Joint Conference on Computer Vision, Imaging, and Computer Graphics Theory and Applications (VISAPP), Vol. 5, pp. 325-334. DOI: 10.5220/0006620803250334.
12. **Khaligh-Razavi, S. M. (2014).** What you need to know about the state-of-the-art computational models of object-vision: A tour through the models. arXiv:1407.2776. DOI: 10.48550/arXiv.1407.2776.
13. **Britanak, V., Yip, P.C., Rao, K.R. (2010).** Discrete cosine and sine transform: general properties, fast algorithms, and integer approximations. Elsevier.
14. **Saïdani, A., Echi, A.K. (2014).** Pyramid histogram of the oriented gradient for machine-printed/handwritten and Arabic/Latin word

discrimination. 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR), pp. 267–272. DOI: 10.1109/SOCPAR.2014.7008017.

15. **Simonyan, K., Zisserman, A. (2015).** Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd International Conference on Learning Representation (ICLR).

Article received on 25/11/2021; accepted on 11/03/2022.
Corresponding author is Nadir Berrouane.

ISSN 2007-9737

How Much Deep is Deep Enough?

Diego Uribe, Enrique Cuan

TecNM Instituto Tecnológico de La Laguna,
Mexico

{duribea, ecuand}@lalaguna.tecnm.mx

Abstract. Typical deep learning models defined in terms of multiple layers are based on the assumption that a better representation is obtained with a hierarchical model rather than with a shallow one. Nevertheless, increasing the depth of the model by increasing the number of layers can lead to the model being lost or stuck during the optimization process.This paper investigates the impact of linguistic complexity characteristics from text on a deep learning model defined in terms of a stacked architecture. As the optimal number of stacked recurrent neural layers is specific to each application, we examine the optimal number of stacked recurrent layers corresponding to each linguistic characteristic. Last but not least, we also analyze the computational cost demanded by increasing the depth of a stacked recurrent architecture implemented for a linguistic characteristic.

Keywords. Recurrent neural networks, stacked architectures, linguistic characteristics.

1 Introduction

Nowadays, the successful application of deep learning models performing tasks without human intervention is part of our daily lives. For example, the use of deep learning models such as convolutional networks in visual recognition exhibited a spectacular success in the largest contest in object recognition known as ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [16]. Natural Language Processing is another difficult task in which the implementation of deep learning models such as recurrent neural networks (RNN) has contributed to improve the state of the art in multiple challenges of natural language understanding.

For example, the work of Jain et al. to produce short stories from short remarks is an evidence of how text generation has exhibited great progress [10]. The work of Wang et al. to predict sentiment polarity from tweets is also an illustration of how sequence classification tasks have attested good results [23].

Bengio, one of the pioneers in the field, argues the necessity of deep architectures for more efficient representation of AI-high-level tasks by making use of multiple hidden layers instead of shallow models [1, 2]. In our case, we study the use of stacked recurrent architectures, that is, deep architectures based on stacking multiple recurrent hidden layers on top of each other.

Since recurrent neural networks, ss fundamental deep learning algorithms for sequence processing, can be built in many different ways, we analyze three basic and popular recurrent architectures: simple recurrent networks [5], Long Short-Term Memory (LSTM) networks [9], and Gate Recurrent Unit (GRU) networks [3]. LSTM y GRU are more sophisticated architectures that have been created to cope not only with the necessity of memory "to remember" previous elements in a sequence but also with the "vanishing gradient problem" that is experimented with neural networks of a big depth, i.e. those feedforward networks with many hidden layers.

To determine the right depth of a stacked recurrent architecture to text processing is the main focus in this investigation. Specifically, our purpose is to provide empirical evidence about the relationship between the linguistic characteristics of the text and the stacked recurrent learning models.

In other words, the research question addressed in this paper is: what is the proper depth of a stacked recurrent architecture corresponding to a particular level of linguistic complexity inherent in the texts?

Nonetheless, in order to give answer to this question, it is first important to elucidate another concern: how to define the linguistic complexity of the textual expressions? Based on the literature review, we have identified some linguistic properties to determine the complexity occurring in the texts.

How much hard is the comprehension of a text? Text comprehension is a crucial linguistic property to determine the complexity of a text. Paul Rhea argues that the difference between easy and complex sentences is related to the number of embeddings, that is, the number of nested clauses within the sentence [19]. For example, the difference between the following sentences is notorious: the second sentence is more complex and demands more computational processing.

(i) *The sad reality is that most Canadians simply don' t care about access to information.*

(ii) *The sad reality is that most Canadians, as long as they can watch Don Cherry on Sat nite and as long as Tim Hortons keeps their donut prices affordable, they simply don' t care about access to information.*

Is it necessary to increase the depth of a recurrent learning model when we process texts similar to the first sentence? About texts containing long-term dependencies as the second sentence, will it be necessary to increase the number of stacked recurrent layers?

In short, which stacked recurrent architectures coping with particular linguistic characteristics provide more representational power than a single-layer recurrent model? To give answer to these questions, it is paramount to determine the linguistic properties of the texts to be processed.

In addition to the assessment of complexity based on the number of embedded clauses in the texts (as we explained above), we also explore the entropy of the texts as a metric of how rich the vocabulary of texts is.

Taking as reference the work of Keller [12] in which he makes evident a correlation between the entropy of a sentence and the complexity required for its comprehension, the consideration of entropy in this work is based on the assumption that if the text is more complex, the author uses a more varied vocabulary.

This study also consider the use of quantitative methods as assessment of textual complexity such as the analysis of how long a text is. In fact, the number of sentences and the number of content words in the text are regarded as textual properties to determine the impact on the performance of a stacked recurrent architecture.

And last but not least, we also analyze the computational cost demanded by increasing the depth of a stacked recurrent architecture put into practice for each linguistic characteristic. In summary, the main contributions of this research work are the following:

— Providing empirical evidence about the impact of linguistic complexity characteristics from texts on a deep learning model defined in terms of a stacked recurrent architecture.

— Using qualitative (embedded clauses and entropy) and quantitative (number of sentences and words) methods, multiple linguistic complexity characteristics of the texts are considered in this research.

— Applying deep learning algorithms (RNN, LSTM and GRU), various stacked recurrent architectures are implemented in this study.

2 Related Work

In this section we first proceed with a brief description of preceding works about the investigation of deep networks for machine learning purpose. Then we comment on previous works about text analysis and the use of linguistic properties for the computational processing of the texts.

2.1 Stacked Architectures

Utgoff and Stracuzzi presented many-layered learning as the need of many layers of knowledge for learning non-trivial concepts [22]. When a difficult problem is broken into a sequence of simple problems, learning is modeled with layered knowledge structures defined in terms of interdependent and reusable concepts named building blocks. In this way, assimilation of new knowledge makes use of previous knowledge.

By using equivalent Boolean functions, where one of them is defined in terms of nested basic elements in a more compact expression, a model learning for the assimilation of the target concept require different number of layers for each equivalent function. In fact, a nested Boolean function, that denotes a complex concept described in terms of simple elements (i.e. building blocks), requires more learning layers to assimilate and reuse the target concept sometime thereafter.

In contrast, the equivalent Boolean function, that has not been described in terms of building blocks, requires less learning layers (i.e. shallow network) to assimilate a concept hardly reusable. In other words, since knowledge reuse is an essential factor for achieving successful learning, this investigation shows how shallow networks make difficult the learning process.

Graves et al. showed how a deep learning model based on a stacked recurrent architecture was effective for phoneme recognition [8]. Even tough RNNs are a type of neural networks suitable for sequential data, in speech recognition better results were obtained by deep feedforward networks (vanilla neural networks). This antecedent was the main reason to investigate the use of deep recurrent neural networks for speech recognition. In particular, a deep LSTM architecture was applied to speech recognition and better performance was obtained over learning models based on single-layer LSTM.

Two key elements were considered in the definition of a deep learning model for speech recognition. First, with the use of a recurrent architecture as LSTM, not only the analysis of previous elements is considered but also the analysis of a long range context in the sequence. Furthermore, the LSTM architecture was enriched with bidirectional layers so the elements in the sequence were examined in both directions.

Secondly, a deep architecture was defined by stacking multiple bidirectional recurrent layers on top of each other, with the output sequence of one layer forming the input sequence for the next. By stacking recurrent layers, the model generates multiple levels of representation which proved to be relevant in the processing of the phonemes. In this way, the combination of multiple levels of representation with long range context analysis of the acoustic sequence was essential for building up an effective stacked recurrent architecture for speech recognition.

Pascanu et al. explored different ways to extend a recurrent neural network (RNN) to a deep RNN [18]. This research was inspired by previous works showing how increasing the depth of a classic neural network proved to be more efficient at representing some functions than a shallow one. Based on the processing of a simple RNN, they analyzed the basic steps carried out by an RNN in order to identify points of deeper extensions.

Since the basic steps: input-to-hidden function, hidden-to- hidden transition and hidden-to-output function are all shallow, that is, there is no intermediate layer, an alternative deeper design was proposed for each shallow point, and in this way, deeper variants of an RNN.

For example, the hidden-to-hidden transition was made deeper by having one or more intermediate nonlinear layers between two consecutive hidden states (h_{t-1} and h_t).

Two deeper variants of an RNN were empirically evaluated on the tasks of polyphonic music prediction and language modeling. The experimental results proved how the depth of the proposed variants of an RNN was essential to outperform the shallow RNNs. Another interesting outcome of the experimentation was that each of the proposed deep RNNs has a distinct characteristic that makes it more, or less, suitable for certain types of datasets.

2.2 Linguistic Features

We now briefly describe some interesting works about text complexity and constructiveness classification. Santucci et al. investigated the complexity level of an Italian text from the classification task perspective [20]. The experimentation made use of a collection of texts produced for second language teaching purpose and a large set of linguistic features were defined to be used among ten machine learning models where the random forest stood out from the rest. But beyond the good accuracy results obtained, the key point was the conduction of a deep analysis to identify the set of linguistic features that influenced the good prediction results.

Another interesting investigation was carried out by Yasseri et al. They investigated the complexity of two categories of English texts from Wikipedia: Simple and Main texts where both text samples exhibit rich vocabulary. Statistical analysis of linguistic units such as n–grams of words and part of speech tags provided empirical evidence of how the language of Simple texts is less complex due to the use of shorter sentences.

With the comparison of these categories by the Gunning readability index was evident the linguistic complexity not only in terms of the linguistic units but also in terms of the topic addressed in texts: the language of conceptual articles is more elaborate compared to biographical and object–based articles [24].

Kolhatkar and Taboada implemented classical (SVM classifiers) and deep (biLSTMs) learning models to recognize constructive comments by making use of two datasets for training: the New York Times Picks as positive examples and the Yahoo News Annotated Comments Corpus as negative examples of constructive online comments.

The learning models were evaluated on a crowd-annotated corpus containing 1,121 comments. In this investigation, multiple sets of constructiveness features were defined: length, argumentation, named–entity and text–quality features. The purpose of such sets was the identification of those crucial features to determine how argumentative or constructive a comment is [13]. These sets of constructiveness features provided the inspiration for a deep work.

3 Learning Models and Linguistic Characteristics

This section describes the stacked recurrent architectures and the linguistic characteristics of the texts contemplated in this study. As we previously mentioned, the motivation of this research is to investigate the impact of linguistic complexity characteristics from texts on a deep learning model defined in terms of a stacked recurrent architecture. We first explain each recurrent neural network and the corresponding linguistic peculiarities that caused it. Then, we describe the linguistic characteristics for the analysis of how complex a text is.

3.1 Stacked Recurrent Architectures

Stacked Recurrent Architectures are the focus of our attention in this work. A model based on vanilla neural networks that increases its representational capacity with more layers and more hidden units per layer was the antecedent for the investigation of stacked recurrent architectures. As we previously said in section 2, stacked recurrent architectures outperform single-layer networks on multiple tasks such as phoneme recognition [8].

A stacked recurrent architecture can be defined as a deep learning model comprised of multiple recurrent layers where the output of one layer serves as the input to a subsequent layer. The intention of stacking recurrent layers is to increase the representation power of a single recurrent neural network in order to induce representations at differing levels of abstraction across layers [11]. Critically important in this stacked architecture is how a recurrent layer provides a full sequence output rather than a single value output obtained at the last timestep. Figure 1 shows the structure of a stacked recurrent architecture.

Now we describe the kind of recurrent layers to be stacked in the definition of a deep learning model, that is, the kind of recurrent neural networks specialized for processing a sequence of elements where the order of the textual elements (e.g.

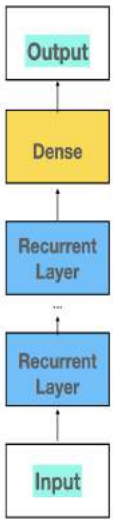


Fig. 1. Stacked Recurrent Architecture

words) is relevant to understand a document. Since recurrent neural networks can be built in many different ways, we study three basic and popular architectures: simple recurrent networks (SimpleRNN), LSTM and GRU.

Each of these architectures was created to address some linguistic concern. Simple recurrent networks (SimpleRNN), also known as Elman Networks [5], are the first neural architecture to represent the temporal nature of language as each element of a sequence is processed at a time. The key point in this model is the computation of the hidden layer: to activate the current hidden layer is necessary the value obtained in the previous hidden layer corresponding to a preceding point in time.

Equation (1) expresses the computation of the hidden layer $\boldsymbol{h}$ where $\boldsymbol{x}$ denotes the sequence (i.e. input) and $\boldsymbol{g}$ an activation function. $\boldsymbol{W}$ denotes the weight matrix corresponding to the input x_t whereas $\boldsymbol{U}$ denotes the weight matrix corresponding to the hidden layer of the previous timestep h_{t-1}. In this way, this connectionist model is concerned with the context corresponding to each element of the sequence:

$$h_t = f(\boldsymbol{U}\, h_{t-1} + \boldsymbol{W}\, x_t). \tag{1}$$

However, since only previous elements are taken into consideration, SimpleRNN cannot keep track of long-term dependencies.

Long Short–Term Memory (LSTM) networks were created to include the consideration of distant constituents and, in this way, to extend the local context to be analyzed. Vanishing gradient problem is another difficulty with SimpleRNNs that arises during the backward process for updating the weights. In order to overcome these problems, LSTM networks were created with three more gates to forget information that is no longer needed and to add information for posterior decisions [9].

It is precisely the addition of these gates that makes of LSTM a complex recurrent network: 4 gates and 2 weights (U and W) to learn for each gate. As an alternative to the LSTM network, the Gated Recurrent Unit (GRU) was created by [3]. By reducing the number of gates to only 2 and removing the context vector, GRU was introduced as an architecture so effective as LSTM but less complicated. Goldberg mentions, in his analysis of the multiple neural network models for NLP [6], the work of comparison between LSTM and GRU carried out by Chung et al [4]. This investigation evaluated these two recurrent architectures on two tasks and found how the performance of GRU was comparable to LSTM.

3.2 Linguistic Characteristics and Complexity

The linguistic characteristics for the analysis of how complex a text is are explained here. As a fundamental unit in the collection of texts, a sentence and its complexity are crucial in this investigation. Since the complexity of a sentence is inherent to the difficulty of its understanding, we analyze the complexity of a sentence from the perspective of the computational difficulties involved in its processing. To be more precise, we analyze the complexity of a sentence from the perspective of the computational difficulties involved in the processing of the diverse recurrent neural architectures.

A review of the literature on linguistic complexity lead us to the work of Miller and Chomsky who showed how a embedded structure, that is, a syntactic structure that contains another syntactic

structure nested within itself, is particularly difficult for understanding and parsing processing [17].

In children education, an important area of assessment is the understanding of complex sentences [21]. Since a complex sentence is mainly used for expressing a more elaborate thought, it is crucial the use and identification of complex sentences. In this study, the identification of a complex sentence is based on the definition of R. Paul [19]: a complex sentence is an embedded sentence (it contains an independent clause and a dependent clause) or conjoined sentence (it contains two or more independent clauses joined together using a conjunction).

In addition to the analysis of complexity based on the number of embeddings, we also explore the use of entropy, as how it is well known, entropy is a measure of information randomness. We explore in this work the use of entropy of a text as a metric of how rich the vocabulary of the text is. Taking as reference the work of [12] in which he makes evident a correlation between the entropy of a sentence and the complexity required for its comprehension, the consideration of entropy in this work is based on the assumption that if the text is more complex, the author uses a more varied vocabulary. We represent a text as a sequence of words W = $\{w_1,w_2, \ldots, w_n\}$ to compute the entropy as follows:

$$H(w_1, w_2, \ldots, w_n) = -\sum_{i=1}^{n} p(w_i) \log p(w_i). \quad (2)$$

Finally, we also contemplate textual properties for the analysis of how long a text is. Based on the assumption that if the text is more argumentative, the author makes use of more words and sentences, the number of sentences and the number of content words in the text are analyzed to determine the impact on the performance of a learning model [13].

4 Experimental Evaluation

In order to provide empirical evidence for the relationship between different linguistic characteristics of the texts and performance of stacked recurrent architectures, our experimentation is based on text classification.

In particular, we focus our attention on the identification of constructive comments. In this way, we begin this section with the description of the corpus, that is, the collection of constructive and non-constructive comments used in the experimentation. Then, we describe the framework of the experimentation and conclude with the presentation of the obtained results for each particular linguistic characteristic and each deep learning model.

4.1 Data

The corpus used in our work, known as Constructive Comments Corpus (C3), is a collection of 12,000 online news comments with metadata information about constructiveness and toxicity [14].

The distribution of comments to classes represents an almost balanced dataset: 6,516 constructive comments (54%) and 5,484 non-constructive comments (46%). The comments to be submitted to the annotation process were obtained from the SFU Opinion and Comments Corpus (SOCC) which contains a collection of opinion articles and the comments posted by readers in response to the article [15].

What does constructive mean? To give answer to this question, a set of characteristics was defined for the concept of constructive as well as a set of characteristics for the notion of non-constructive. The sub-characteristics corresponding to each concept are shown in Table 1.

In this way, when the comment exhibits properties as evidence and dialogue there is a high probability that annotators classify the comment as constructive. And the opposite is also possible: when the comment exhibits properties as irrelevant and sarcastic there is a high probability that annotators classify the comment as non-constructive.

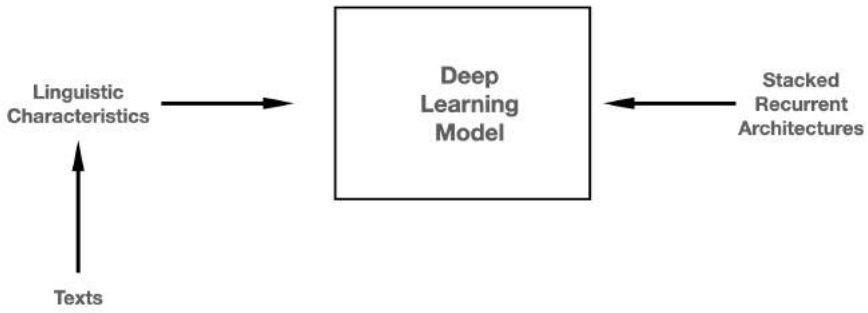


Fig. 2. Experimentation process

Table 1. Characteristics

	sub-characteristic
Constructive	constructive
	dialogue
	solution
	specific_points
	personal_story
	evidence
Non-Constructive	non_constructive
	provocative
	sarcastic
	no_respect
	unsubstantial
	non_relevant

4.2 Experimental Setup

The Figure 2 illustrates the experimentation process: from the comments collection we extract a subset according to a particular linguistic characteristic, and then, the extracted subset is provided to the deep learning model corresponding to each of the stacked recurrent architectures to be considered in this study.

Before displaying the results for each linguistic characteristic, we introduce the setup for the definition of the deep learning models. The first layer of each deep learning model is defined by learning a distributed representation corresponding to the subset of complex comments previously extracted.

In other words, we define a deep learning model with a word embedding as input where a word is represented with an output vector size of 32.

The stack is created by progressively increasing the recurrent layers from 1 to 5 so we can analyze the impact of systematically increasing the layers in the performance of the learning model.

The model is trained by using Adam as the stochastic gradient descent optimizer for 10 epochs. Additional parameters for training are: batch size = 64, learning rate = 0.01, and loss function = 'binary crossentropy'.

In this way, a deep learning model is defined for each recurrent neural architecture analyzed in this work: simple recurrent network, LSTM and GRU. As the number of available comments for each linguistic characteristic varies, we perform three-fold cross validation to use all of the pertinent comments in the subset.

4.3 Experimental Results

Embedded Clauses: in this case we first extract the subset of comments which contains embedded or conjoined sentences (from 0 to 3). The obtained results for each deep recurrent architecture are shown in Figure 3 where we can see how the performance of the SimpleRNN model improves as the number of layers increases regardless the number of clauses the text contains. On the other hand, increasing the layers to the GRU model doesn't influence the initial performance except when the text contains three clauses at least. However, the best performance is obtained with the LSTM model when the comments contains two clauses at least, that is, texts of great linguistic complexity, and the number of layers is increased.

Entropy: in this case we first extract the subset of comments for each entropy value (from 1 to 4) as a useful indicator of how rich the vocabulary of the text is. The obtained results for each deep recurrent architecture are shown in Figure 4 where we can see how the best performance is obtained with the SimpleRNN model regardless the entropy value. By increasing the number of layers, the SimpleRNN model proves to be able to cope with lexical variation. On the other hand, increasing the layers to the LSTM and GRU models doesn't cause any impact on the initial performance.

Number of sentences and words: in this case we first extract a subset of comments for each limit of sentences (from 1 to 4) and for each limit of words (from 10 to 40) as an indicator of the computational processing demanded by the use of more sentences and words. The obtained results for each deep recurrent architecture are shown in Figure 5 and 6.

About the impact of the number of sentences, in Figure 5 we see how the performance of the SimpleRNN model improves as the number of layers increases regardless the number of sentences the comment contains. On the other hand, increasing the layers to the GRU model doesn't influence the single-layer performance except when the text contains four sentences at least. However, the best performance is clearly obtained with the LSTM model when the linguistic complexity is high, that is, when comments contains three sentences at least. Said in another way, when the comment substantiates its claims by providing more details or reasons, increasing the number of recurrent layers notoriously improves the performance of the LSTM model.

Now, considering the number of words, in Figure 6 we see how, by increasing the number of layers, the best performance is obtained with the SimpleRNN model except when the comment contains 40 words at least. However, the best performance is obtained with the LSTM model when the comments contains 40 words at least, that is, by increasing the number of layers, the LSTM model is able to cope with long and more complex texts. On the other hand, increasing the layers to the GRU model doesn't influence the initial performance so a shallow model seems to be a good option.

5 Discussion

The benchmark for a discussion is the obtained results by the creators of the Constructive Comments Corpus known as C3 [14]. They implemented multiple experiments with two different learning models: a classic model based on predetermined features and deep learning models. Multiple sets of features were analyzed with a classic SVM model in order to figure out what properties cause high impact in predicting constructiveness. In our discussion, we make reference to those features related to the linguistic characteristics analyzed in this work only.

Embedded clauses: to have a better analysis of the impact of the linguistic complexity denoted by the number of embedded clauses in the texts, Figure 7 shows how each stacked recurrent architecture behaves in different levels of complexity. We see how all recurrent architectures struggle when the complexity increases. Thus, this is an empirical evidence of how the complexity of a text is related to the number of embeddings, that is, the number of nested clauses within the sentence [19].

Now, which has been the impact of a stacked architecture? We see how by increasing the number of recurrent layers the best performance is obtained with a SimpleRNN model when the level of complexity is low (stack size = 2). On the

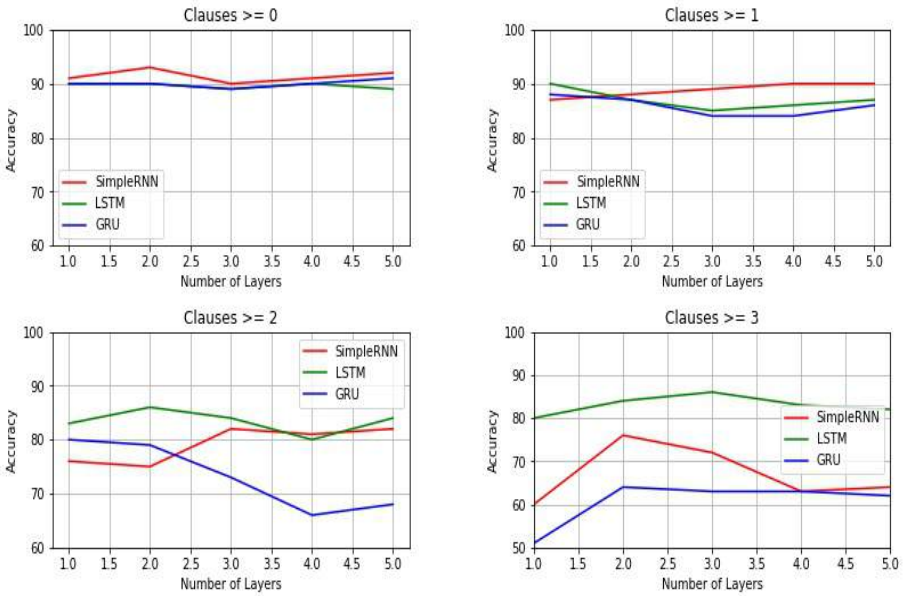


Fig. 3. Comment's clauses and stacked recurrent architectures

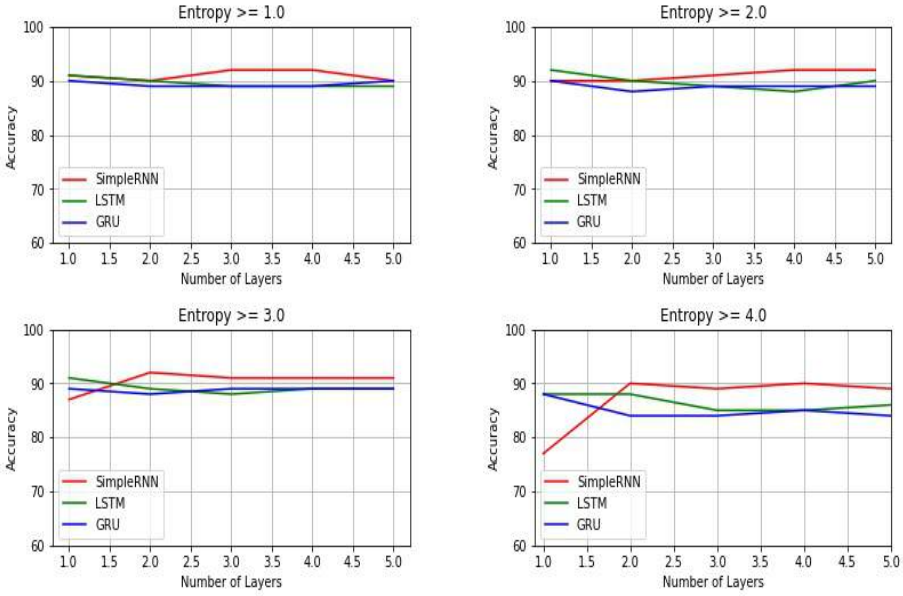


Fig. 4. Comment's entropy and stacked recurrent architectures

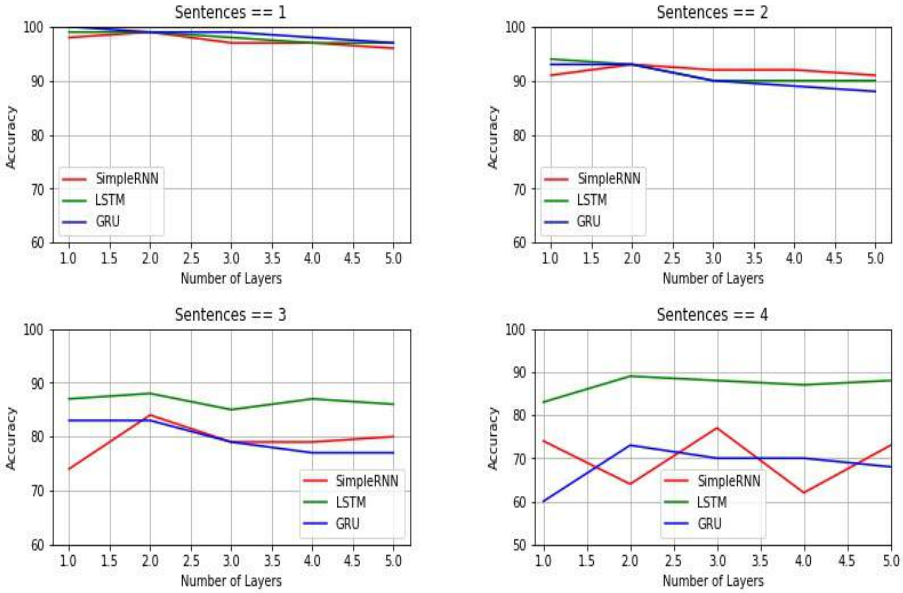


Fig. 5. Comment's sentences and stacked recurrent architectures

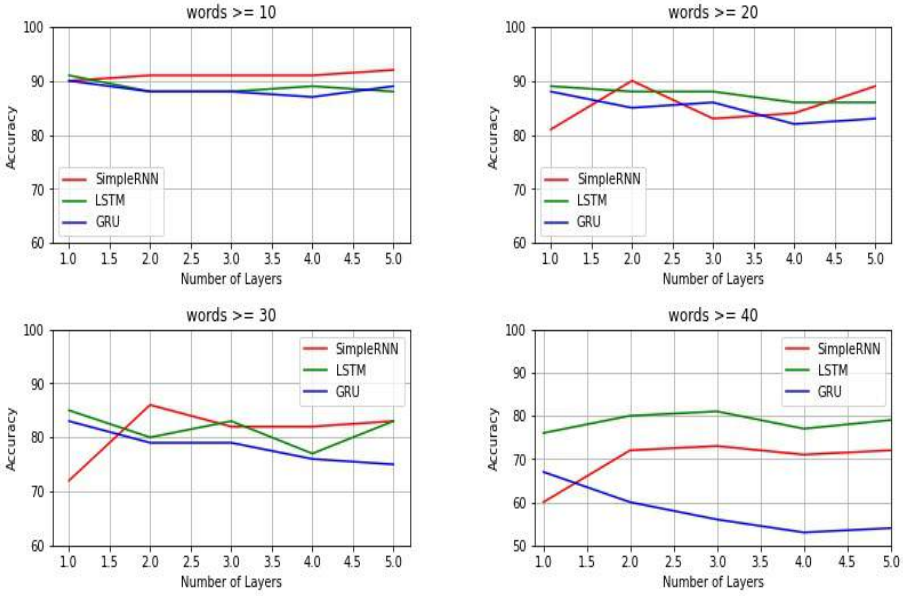


Fig. 6. Comment's words and stacked recurrent architectures

other hand, when the level of complexity is high, the best performance is obtained with a LSTM by increasing the number of layers (stack size = 3). Thus, stacked recurrent architectures prove to be useful to cope with high and low values of this particular linguistic complexity.

We now compare the performance of our models with the results obtained by Kolhatkar et al. [14]. They implemented a classic SVM model with multiple sets of features where *argumentation* is the set of features comparable to the complex sentences analyzed in our case. The set of *argumentation* features is:

— presence of discourse connectives (therefore, due to),

— reasoning verbs (cause, lead),

— abstract nouns (problem, issue, decision, reason),

— stance adverbials (undoubtedly, paradoxically).

These features were selected based on the assumption that an argumentative text is one that exhibits reasons and explanations. The result obtained by the SVM model was a 0.76 F1 score whereas ours results for different levels of complexity are shown in Figure 7. As we can see in Table 2 (the values are obtained from Figure 7), when the comments contains one or two complex sentences at least, the results obtained by the recurrent architectures are higher than the SVM model.

Table 2. Two embedded clauses and stacked recurrent architectures

model	stack size	acc
SimpleRNN	3	0.82
LSTM	2	0.86
GRU	1	0.80

However, when the comments contains three or more complex sentences, the result obtained by SimpleRNN is identical to the SVM model whereas the result obtained by GRU is lower.

Table 3. Three or more embedded clauses and stacked recurrent architectures

model	stack size	acc
SimpleRNN	2	0.76
LSTM	3	0.86
GRU	2	0.64

Table 3 shows a stacked LSTM model clearly able to cope with the classification of argumentative comments, that is, the highest level of linguistic complexity.

Entropy: Figure 8 shows the impact of the linguistic complexity denoted by the entropy of the texts where different levels of entropy for each stacked recurrent architecture is displayed. In this case we also see how the performance of all recurrent architectures drops when the complexity increases. For this reason the consideration of textual entropy as linguistic complexity makes evident the correlation between the entropy of a sentence and the complexity required for its comprehension suggested by Keller [12].

About the impact of a stacked architecture we see in Figure 2 how by increasing the best performance is obtained with a SimpleRNN model for any level of complexity. For example, when the level of complexity is low (entropy = 2), a stack of four recurrent layers obtains good performance; and when the level of complexity is high (entropy = 4), a stack of four recurrent layers outperform the LSTM and GRU architectures. On the other hand, a deep model based on LSTM and GRU architectures had no positive effect so a shallow model was the best option in these cases.

In order to compare the performance of our models with the results obtained by Kolhatkar et al. [14], *Text quality* is the set of features comparable to the use of entropy as an indicator of a more varied vocabulary. The set of *Text quality* features is:

— Readability score,

— Personal experience score.

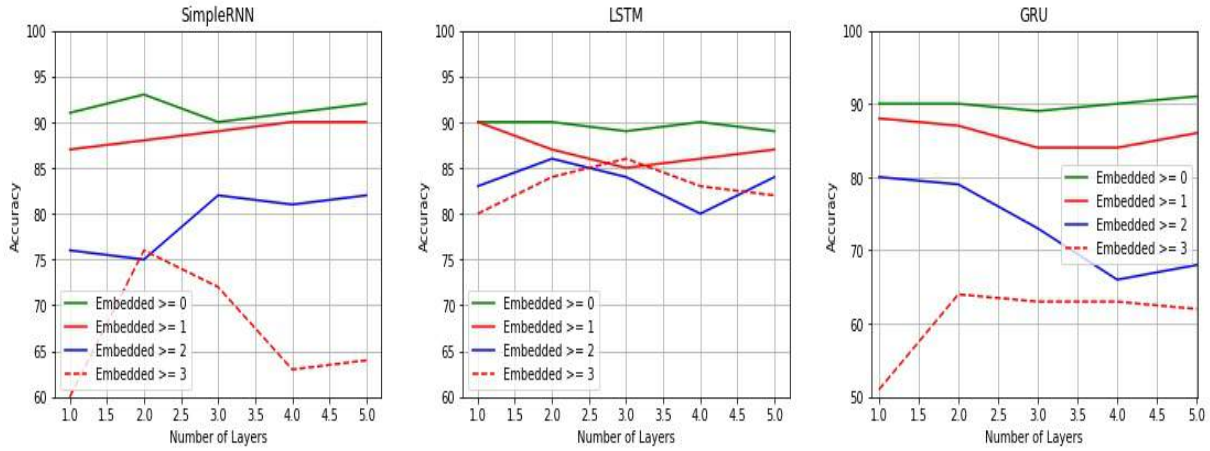


Fig. 7. Stacked recurrent architectures and Embedded clauses

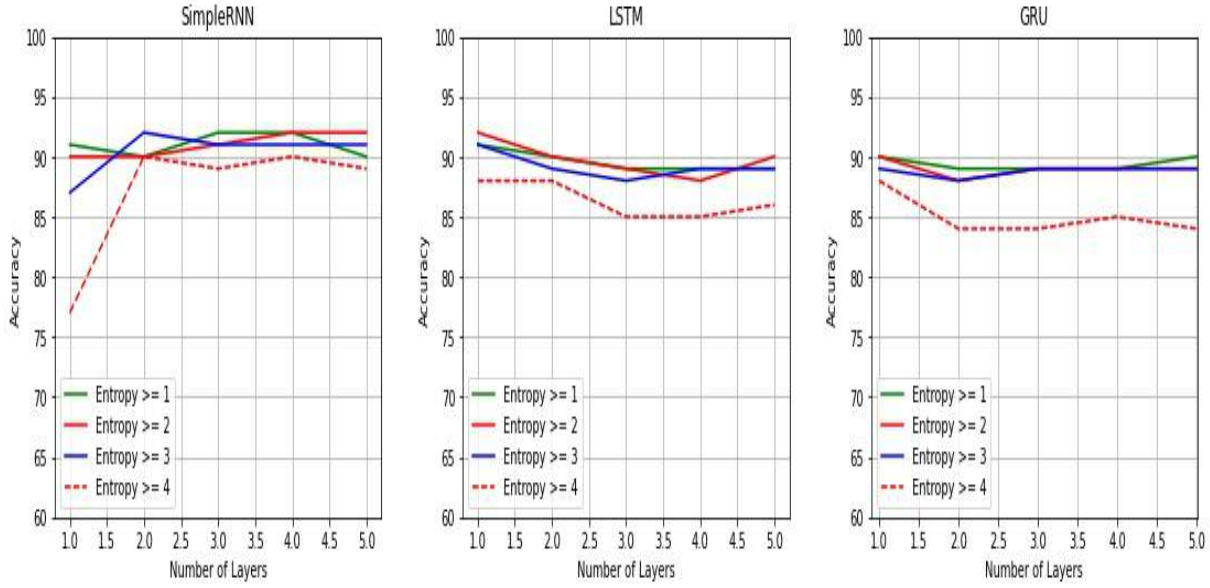


Fig. 8. Stacked recurrent architectures and Entropy values

These features were selected as an attempt to quantify how hard a text is to read. The result obtained by the SVM model was a 0.90 F1 score whereas ours results for different entropy values are shown in Figure 8. As we can see in Table 4 (the values are obtained from Figure 8), when the comments contains a low entropy value ($entropy \geq 2$), the results obtained by the deep recurrent architectures are higher than the SVM model.

However, when the comments contains a high entropy value ($entropy \geq 4$), the result obtained by SimpleRNN is identical to the SVM model whereas the results obtained by LSTM and GRU are lower. Table 5 shows a stacked SimpleRNN model achieving similar performance to the SVM model when the readability of the comments demands a rich vocabulary, that is, when the comments exhibit the highest level of linguistic complexity.

As we previously said, Table 4 and Table 5 also show how a deep model based on LSTM and GRU architectures had no positive effect so a shallow model was the best option in both cases.

Number of sentences and words: in this case, from the results obtained by the analysis of number of sentences and words, we can see in Figure 9 and Figure 10 how precision decreases as the number of sentences and words in the comments increases. This decline in the performance of the recurrent architectures is a clear-cut evidence that the complexity of constructive comments is related to the number of sentences and words.

Figure 9 and Figure 10 also illustrates the impact of a stacked SimpleRNN architecture. In fact, we see how by increasing the number of recurrent layers the best performance is obtained with a SimpleRNN model when the level of complexity is low (stack size = 2 for sentences and words). Now, when the level of complexity is high, the best performance is obtained with a stacked LSTM architecture: stack size = 2 for sentences and stack size = 3 for words. On the other hand, a deep model based on GRU architecture had minimum impact on the classification task.

In order to compare the performance of our models with the results obtained by Kolhatkar et al. [14], *Length* is the set of features comparable to the use of number of sentences and words as an indicator of a constructive comment. The set of *Length* features is:

— Number of tokens in the comment,

— Number of sentences,

— Average word length,

— Average number of words per sentence

These features were selected to verify the premise that the length of the text is a good predictor of constructiveness. The result obtained by the SVM model was a 0.93 F1 score whereas ours results for various number of sentences and words are shown in Figure 9 and Figure 10. As we can see in Table 6 (the values are obtained from Figures 9 and 10), when the comment is short (number of sentences = 2), the results obtained by a deep SimpleRNN architectures and GRU are identical to the SVM model. The performance of a single layer LSTM network is a bit higher on this level of complexity.

Table 4. Low entropy ($entropy \geq 2$) and stacked recurrent architectures

model	stack size	acc
SimpleRNN	4	0.92
LSTM	1	0.92
GRU	1	0.90

Table 5. High entropy ($entropy \geq 4$) and stacked recurrent architectures

model	stack size	acc
SimpleRNN	4	0.90
LSTM	1	0.88
GRU	1	0.88

Table 6. Low number of Sentences and Words

	$Sentences = 2$		$Words \geq 20$	
model	stack size	acc	stack size	acc
SimpleRNN	2	0.93	2	0.90
LSTM	1	0.94	1	0.89
GRU	1	0.93	1	0.88

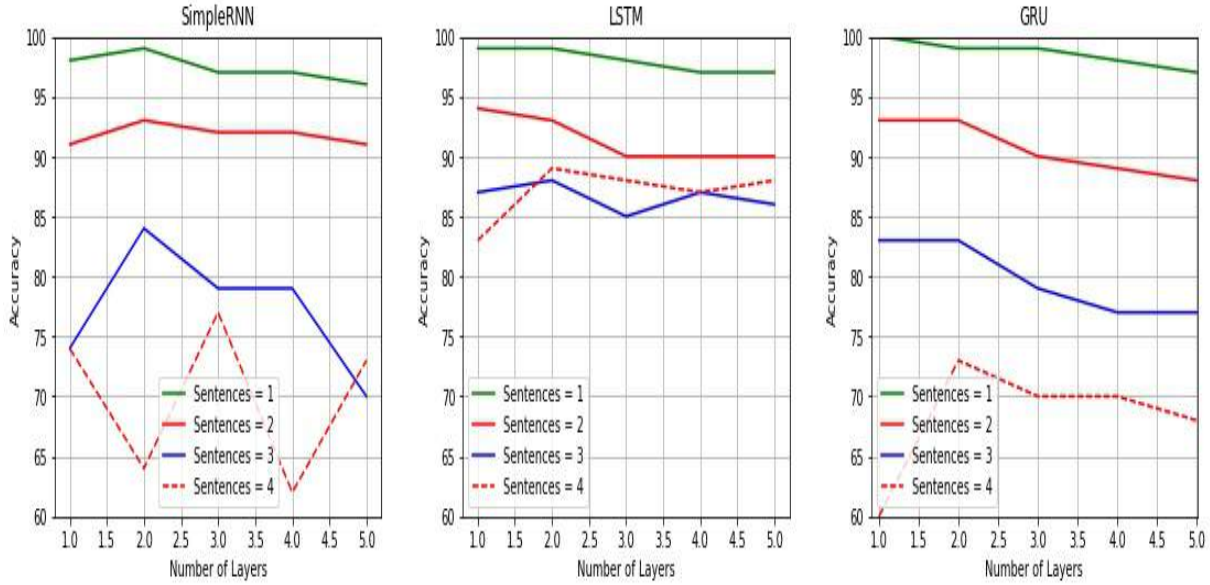


Fig. 9. Stacked recurrent architectures and Sentences values

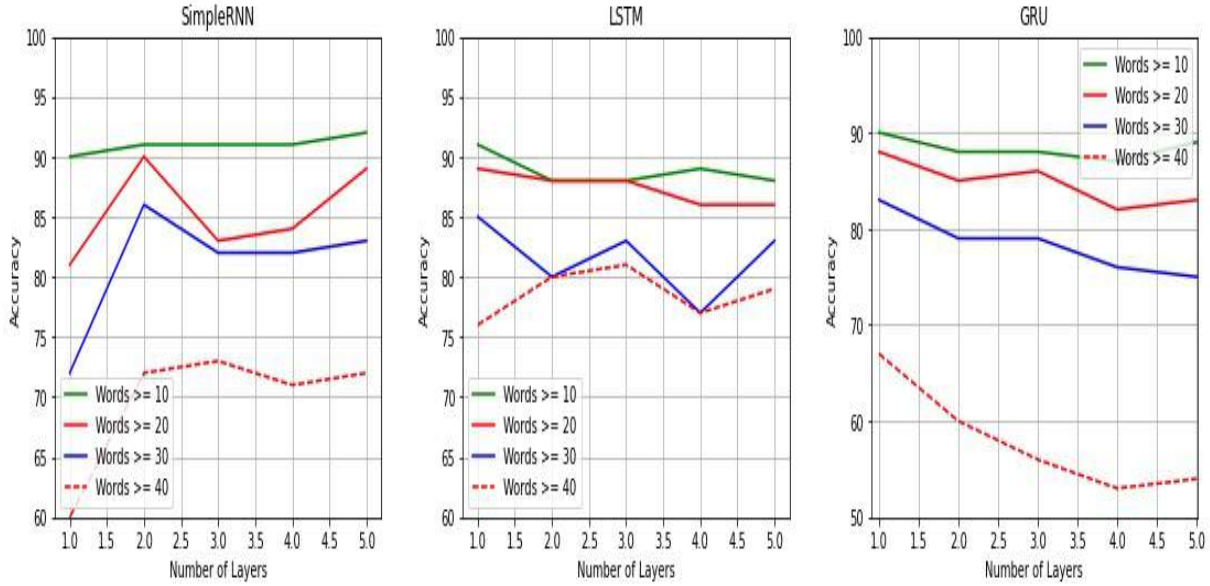


Fig. 10. Stacked recurrent architectures and Words values

Now, when the linguistic complexity of the text is high, that is, when long comments contain at least 4 sentences or at least 40 words, a stacked model implemented with a LSTM architecture obtains the best performance. Table 7 shows how a stacked LSTM model outperforms the SimpleRNN and GRU architectures.

Table 7. High number of Sentences and Words

	$Sentences = 4$		$Words \geq 40$	
model	stack size	acc	stack size	acc
SimpleRNN	3	0.77	3	0.73
LSTM	2	0.89	3	0.81
GRU	2	0.73	1	0.67

Computational cost: although increasing the representational power of the network, stacking recurrent layers entails a tradeoff to be considered. In fact, stacking recurrent layers generates a tradeoff between increasing network capacity and demanding higher computational resources. Since determining the runtime and memory requirement of the recurrent architectures is highly platform-dependent, we do not describe in this work the computational cost in absolute terms. We describe rather the computational cost as a degree of runtime.

In this way, we show evidence of the computational cost of a stack recurrent architecture from two perspectives:

— Processing time required for multiple number of recurrent layers,

— Processing time required for each recurrent architecture.

Figures 11 and 12 display the percent of processing time required for each number of recurrent layers considered in this investigation: from 1 to 5 layers. Figure 11 shows the computational resources demanded by a LSTM model processing texts that contain one embedded clause at least whereas Figure 12 shows the same LSTM model processing texts that contain two sentences. A linear correlation between depth and time is observed in both figures: the computational processing increases as the depth of the model increases.

Now, Figures 13 and 14 display the percent of processing time required for each recurrent architecture considered in this investigation: SimpleRNN, LSTM and GRU.

Figures13 and 14, corresponding to the processing of texts that contain one embedded clause at least and texts that contain two sentences respectively, show how a stacked learning model based on the GRU architecture proves to be the most demanding model. On the other hand, a deep model based on the primitive SimpleRNN architecture does not demand substantial computational resources.

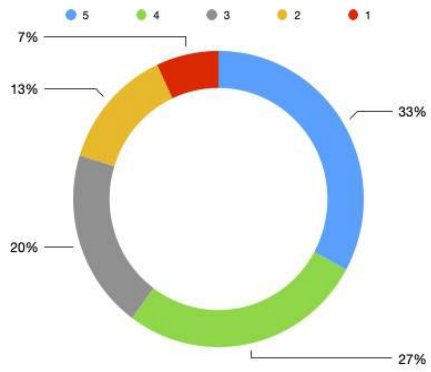


Fig. 11. LSTM model processing texts that contain one embedded clause

6 Conclusion and Future Work

We have analysed in this paper the implications of linguistic complexity characteristics in the performance of a deep learning model defined in terms of a stacked recurrent architecture. To be more specific, we explore linguistic characteristics for the analysis of how complex a text is and, in this way, to investigate the relationship between the linguistic complexity of the texts and the depth of the learning model.

By using qualitative (embedded clauses and entropy) and quantitative (number of sentences and words) methods, our experimentation based

Fig. 12. LSTM model processing texts of two sentences

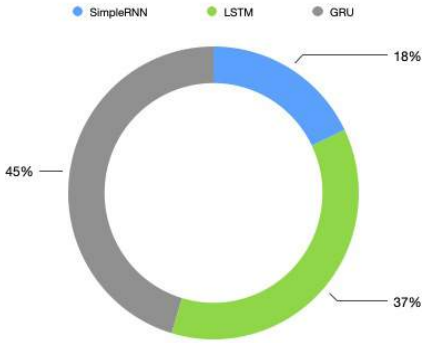


Fig. 13. LSTM model processing texts that contain one embedded clause

on the classification of constructive comments provides empirical evidence of how the linguistic complexity characteristics of the comments impact the stacked recurrent architecture for the identification of constructive comments.

For example, when the number of complex sentences in the comments increases, a single-layer recurrent architecture struggle on the identification of constructiveness. Something similar occurs when the entropy in the comments increase. We show how by increasing the number of recurrent layers, that is, by implementing a stacked recurrent architecture, a better performance is achieved.

In fact, by applying stacked recurrent architectures based on deep learning algorithms such as

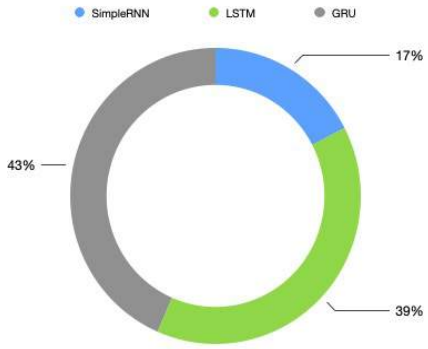


Fig. 14. LSTM model processing texts of two sentences

SimpleRNN, LSTM and GRU, it was posible to observe both the depth of the model improving the results and the number of layers for a model to be lost or stuck (i.e. overfitting) during the optimization process. Finally, we also show how increasing the representational power of the network by stacking recurrent layers entails a tradeoff between increasing network capacity and demanding higher computational resources.

About future work, we are interested in exploring different datasets. The experimentation conducted exhibited how the number of comments available for identification of constructiveness decreases when the linguistic complexity increases. In order to extend our conclusions, it is necessary the use of a different corpus that allows to put aside this limitation.

We are also interested in exploring the intuition that adding a custom attention layer to recurrent neural networks can improve their performance when the linguistic complexity of the texts increases. Since adding attention component to the network has shown significant improvement in tasks such as text summarization and machine translation [7], we want to investigate how an attention component contribuyes to a better representation of complex texts (which contain embedded clauses and long sentences) for classification tasks.

References

1. **Bengio, Y. (2009).** Learning deep architectures for ai. Foundations and Trends in Machine Learning, Vol. 2, No. 1, pp. 1–127.
2. **Bengio, Y., LeCun, Y., Hinton, G. (2021).** Deep learning for ai. Communications of the ACM, Vol. 64, No. 7, pp. 58–65.
3. **Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014).** Learning phrase representations using RNN encoder–decoder for statistical machine translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, pp. 1724–1734.
4. **Chung, J., Gulcehre, C., Cho, K., Bengio, Y. (2014).** Empirical evaluation of gated recurrent neural networks on sequence modeling. NIPS 2014 Deep Learning and Representation Learning Workshop, pp. 1–9.
5. **Elman, J. L. (1990).** Finding structure in time. Cognitive Science, Vol. 14, No. 2, pp. 179–211.
6. **Goldberg, Y. (2016).** A primer on neural network models for natural language processing. Journal of Artificial Intelligence Research, Vol. 57, pp. 345–420.
7. **Goodfellow, I., Bengio, Y., Courville, A. (2016).** Deep Learning. MIT Press.
8. **Graves, A., rahman Mohamed, A., Hinton, G. E. (2013).** Speech recognition with deep recurrent neural networks. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6645–6649.
9. **Hochreiter, S., Schmidhuber, J. (1997).** Long short-term memory. Neural Computation, Vol. 9, No. 8, pp. 1735–1780.
10. **Jain, P., Agrawal, P., Mishra, A., Sukhwani, M., Laha, A., Sankaranarayanan, K. (2017).** Story generation from sequence of independent short descriptions. arXiv, Vol. 1707.05501, pp. 1–7.
11. **Jurafsky, D., Martin, J. H. (2022).** Speech and language processing. 3rd ed. To be published.
12. **Keller, F. (2004).** The entropy rate principle as a predictor of processing effort: An evaluation against eye-tracking data. Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Barcelona, Spain, pp. 317–324.
13. **Kolhatkar, V., Taboada, M. (2017).** Using new york times picks to identify constructive comments. Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism,, Copenhagen, Denmark, pp. 100–105.
14. **Kolhatkar, V., Thain, N., Sorensen, J., Dixon, L., Taboada, M. (2020).** Classifying constructive comments. arXiv, Vol. 2004.05476, pp. 1–24.
15. **Kolhatkar, V., Wu, H., Cavasso, L., Francis, E., Shukla, K., Taboada, M. (2019).** The sfu opinion and comments corpus: A corpus for the analysis of online news comments. Corpus Pragmatics, Vol. 4, pp. 155–190.
16. **Krizhevsky, A., Sutskever, I., Hinton, G. (2012).** Imagenet classification with deep convolutional neural networks. Proceedings of NIPS'2012, Curran Associates, Inc., pp. 1–9.
17. **Miller, G. A., Chomsky, N. (1963).** Finitary models of language users, volume II, chapter Handbook of Mathematical Psychology. John Wiley & Sons, pp. 419–491.
18. **Pascanu, R., Gulcehre, C., Cho, K., Bengio, Y. (2014).** How to construct deep recurrent neural networks.
19. **Paul, R. (1981).** Analyzing complex sentence development, chapter 2. University Park Press, pp. 36–71.
20. **Santucci, V., Santarelli, F., Forti, L., Spina, S. (2020).** Automatic classification of text complexity. Applied Sciences, Vol. 10, No. 20, pp. 1–19.
21. **Steffani, S., Dachty, L. (2007).** Identifying embedded and conjoined complex sentences: Making it simple. Contemporary Issues in Communication Science and Disorders., Vol. 34, No. 2, pp. 44–54.
22. **Utgoff, P. E., Stracuzzi, D. J. (2002).** Many–layered learning. Neural Computation, Vol. 14, pp. 2497–2539.
23. **Wang, X., Liu, Y., Sun, C., Wang, B., Wang, X. (2015).** Predicting polarities of tweets by composing word embeddings with long short-term memory. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the

7th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, Beijing, China, pp. 1343–1353.

24. Yasseri, T., Kornai, A., Kertész, J. (2012). A practical approach to language complexity: A wikipedia case study. PLOS ONE, Vol. 7, No. 1, pp. 1–8.

Article received on 02/10/2021; accepted on 19/01/2022.
Corresponding author is Diego Uribe.

ISSN 2007-9737

wPOI: Weather-Aware POI Recommendation Engine

Rajani Trivedi, Bibudhendu Pati, Chhabi Rani Panigrahi

Rama Devi Women's University,
India

{rtrivedi2576,patibibudhendu, panigrahichhabi}@gmail.com

Abstract. Weather is an integral part of the decision-making process for travelers and, in particular, certain locations or events will not even be recommended during unsafe poor weather. In this article, we introduce a weather assistant framework called wPOI, which calculates weather forecasts in places of interest (POI) that can be suggested. We demonstrate that experience of climatic patterns at a POI and previous insights about how visitors rank their destinations in many different weather situations can be useful in improving the reliability of the choosing. The findings of our research indicate the significantly greater validity of the recommendations and greater comfort with the suggested solution.

Keywords. POI, tourism, itinerary.

1 Introduction

The decision to buy a tourist item or to visit a POI is the result of a difficult decision-making procedure [24]. So many considerations influence tourist decision, a few of them are "internal" to tourists, for example, psychological or experiences of prior experiences, some are "external" (for example, suggestions or feedback, item knowledge, or climate) [17, 20]. Environment and weather are particularly significant considerations in judgment in tourists and affect tourism business' efficient operations [17]. Although visitors can quickly forecast common weather patterns, they are going to face the real weather while they visit a location that can vary between various situations.

In this article, we are focusing on applications and methods which can help forecast POI rankings for tourists and provide the most relevant suggestions for tourists keeping in mind tourist interest, tour popularity and traveling cost.

We are aiming for this purpose by considering the effect of the weather at a particular POI on the tourist assessment of the location in the framework recommended system.

2 Related Work

Recently, the POI recommendation becomes a popular field of study [3, 21, 22]. Several applications [6, 14, 26, 29] have also been built to deliver awesome, peaceful, pleasant tours [19] and random walk [16].

2.1 Background on the Orienteering Problem

In the case of orienteering problems, the various control spots with the associated scores are placed at many places [23]. Competition members strive to maximize their overall performance in the shortest time achievable by reaching as many controls as possible. The most important thing is to reach the highest rating despite the short lifespan [8, 28].

2.2 Tour Recommendation based on Orienteering Problem and its Variants

Lim et al. [11] changed the tourism orienteering problems according to the importance of the POI tour guidance model. Vansteenwegen et al. [27] proposed an approach for adapting the tour schedule so that it would improve the overall balance between the defined degree of involvement from the starting and end, such as expenditure and all POIs. Lim and al. [12] have identified places of importance and reputation in the form of the minimum queuing time.

Tours that satisfy the different levels of tourist interest within the group have been developed in such a manner as the concept of orienteering problem [2, 13].

2.3 Different Tourism Related Work

The time concerned measurement technique that considers tourist visitation at an attraction was proposed by Ying et al. [30]. Furthermore, over a while, the result was shown to make this technique complicated. Aliannejadi et al. [1] developed a possible model to assess the connection between the tourist comments label and a similar attraction. The findings were measuring in combination with studying to rank techniques from different LBSN tools. Given the considerable time, Zhao et al. [31] proposed a latent spatial time model to propose optimal subsequent destinations.

Li et al. [10] recommended that all information from spatial and temporal inspections be stored using the Time-aware Factorizing Personalized Markov Chain (TA-FPMC). The study analyzed the time-decay factor by comparing the gap across two concurrent experiments. Both calculations were built on both check-in experiences and the design was highly complex. The authors also consider the customer as an extra layer that is not necessary for this work. Consequently, if included in the recommended framework, the productivity of a future customer will decrease.

Our suggested model differs significantly from the current POI and tour recommendation scheme: Our algorithms categorize the interests of visitors dynamically depending on time and popularity with geo-tagged images. The POI tourist costs and local distance between the past POI of the initial path and the initial POI of the subsequent path are also reduced in consideration.

If a tourist goes to a new geographical area without a history of his experience, the suggested methodology can offer a suggestion. While some of the solutions suggested have a similar viewpoint, those only have one route in unexplored locations, depending on the kind of POI he/she has experienced The existing approach discovers the connection between a familiar and an unfamiliar location.

The proposed approach recommends several POIs and the associated POIs are also coordinated.

3 Background and Problem Definition

$\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, ...\mathcal{P}_n\}$ is being used to describe the series of POIs in a given city. A POI is graded as Class $\mathbb{C}$ if it meets those requirements, like music, movies, or a park. The cumulative exterior distance between $\mathcal{P}_{\dot{x}}$ and $\mathcal{P}_{\dot{x}+1}$, as well as the distance, traveled at each POI, define the distance covered by a person.

The sum of the distances between both POIs $\mathcal{P}_1$ and $\mathcal{P}_2$ is used to measure the distance among them. According to [14], we used a traveling speed of 4 km/hour. In this study, we recognize 2 different categories of tourists.

3.1 Average POI Visit Duration for Local and Global Users

Every tourists $\mathbb{U}$ are aware of the tour's past information. In a specific POI, Eqn. 1 can be used to measure the duration:

$$\Xi(\mathcal{P}) = \frac{\sum_{\dot{u}=1}^{\dot{k'}} \sum_{\jmath=1}^{\dot{m}} (t_{\jmath}^{d} - t_{\jmath}^{a}) \delta(\mathcal{P}_j = \mathcal{P})}{\sum_{\dot{u}=1}^{\dot{k'}} \nabla_u \, \delta(\mathcal{P}_j = \mathcal{P})} \forall \mathcal{P} \in P, \tag{1}$$

where $\jmath = \{1, 2, ..., \dot{m}\}$, $u = \{1, 2, ..., k'\}$ and ∇ and denotes the number of travels to a given POI by $\delta(\mathcal{P}_j = \mathcal{P}) = \{^{1 \; if \; (\mathcal{P}_j = \mathcal{P})}_{0, \; otherwise}$. In the case of all tourists, $\Xi(\mathcal{P})$ is often employed [4, 7].

3.2 Time-based user interest for local and global users

$\mathbb{C}_{\mathcal{P}}$ represents POI group $\mathcal{P}$, as shown in the prior section. Eqn. 2 helps to register the interest of a particular tourist $\dot{u}$ in POI group $\dot{c}$:

$$\overline{Int}_{u_{\mathcal{P}}} \dot{c} = \sum_{\jmath=1}^{\dot{m}} \frac{(t_{\mathcal{P}_\jmath}^{d} - t_{\mathcal{P}_\jmath}^{a})}{\Xi(\mathbb{P}_\jmath)} \delta(\mathcal{C}_{\mathcal{P}_\jmath} = \dot{c}) \; \forall \dot{c} \in \mathcal{C}, \tag{2}$$

where $\delta(\mathcal{C}_{\mathcal{P}_\jmath}) = \{^{1 \; if \; \mathcal{C}_{\mathcal{P}_\jmath} = \dot{c}}_{0, \; otherwise}$.

The tourist interest for the POI group $\dot{c}$ is measured by Eqn. 2 as per the spending time on

the POI group $\dot{c}$ based on the total spending time by all the tourists. It seems obvious that the tourist will devote most of his or her time at that POI.

3.3 Similarity of Local and Global Users

The identity of local and global tourists are computed based on the Cosine Similarity measure based on the interest of a given destination for both local and global tourists and which is computed utilizing Eqn. 3:

$$\mathbb{S}(\dot{u}_{\dot{x}}, \dot{u}_{\dot{y}}) = \frac{\overline{Int}_{\dot{u}_{\dot{x}}} \cdot \overline{Int}_{\dot{u}_{\dot{y}}}}{||\overline{Int}_{\dot{u}_{\dot{x}}}|| \cdot ||\overline{Int}_{\dot{u}_{\dot{y}}}||}, \tag{3}$$

where $\dot{u}_{\dot{x}}$ and $\dot{u}_{\dot{y}}$ are the 2 distinct tourists.

3.4 Itinerary from Travel History

The traveling reports have been defined for a particular tourist $\dot{u} \in \mathbb{U}$, based on the sequence $\dot{n}$ travel POIs $\mathbb{S}_{\dot{u}} = ((\mathcal{P}_1, t^a_{\mathcal{P}_1}, t^d_{\mathcal{P}_1}), ..., (\mathcal{P}_{\dot{n}}, t^a_{\mathcal{P}_{\dot{n}}}, t^d_{\mathcal{P}_{\dot{n}}}))$, wherein a triplet $(\mathcal{P}_{\dot{y}}, t^a_{\mathcal{P}_{\dot{y}}}, t^d_{\mathcal{P}_{\dot{y}}})$, where $\mathcal{P}_{\dot{y}}$ is the tourist's traveled POI, $t^a_{\mathcal{P}_{\dot{y}}}$ and $t^d_{\mathcal{P}_{\dot{y}}}$ are the time of entry and exit.

The difference between the time of entry and exit indicates the duration of the POI $\mathcal{P}_{\dot{y}}$. Here, $Su = ((\mathcal{P}_1, t^a_{\mathcal{P}_1}, t^d_{\mathcal{P}_1}), ..., (\mathcal{P}_{\dot{n}}, t^a_{\mathcal{P}_{\dot{n}}}, t^d_{\mathcal{P}_{\dot{n}}}))$ could be re-written as $\mathbb{S}_{\dot{u}} = (\mathcal{P}_1, ..., \mathcal{P}_{\dot{n}})$.

3.5 Time-based user Interest of a POI

The POI interest $\mathcal{P}_{\dot{y}}$ is a part of $\mathbb{S}_{\dot{u}}$ and could be calculated with the help of Eqn. 4:

$$\mathcal{P}_{\dot{y}}(int) = \sum_{\jmath=1}^{\dot{k}} \frac{(t^a_{\mathcal{P}_\jmath} - t^d_{\mathcal{P}_\jmath})}{\Xi(\mathcal{P}_\jmath)}. \tag{4}$$

3.6 Popularity of a POI Category

The POI popularity is measured by using Eqn. 5 as per the overall number of tourists visiting on the POI with respect to to the number of tourists visits to every POI:

$$\mathbb{C}(\overline{pop}) = \sum_{\jmath=1}^{\dot{k}} \frac{\overline{pop}_{\mathcal{P}_\jmath}}{\varphi(\mathcal{P}_\jmath)}, \tag{5}$$

where the $\mathbb{C}(\overline{pop})$ represents category $\mathbb{C}$ of popularity and $\varphi(\mathcal{P})$ represents the number of instances of all tourists visiting a specific POI.

3.7 Traveling Cost

The cost of travel is determined using the actual path the traveler travels through. While a few previous studies have taken into account the full length of the trip, distances are a relevant consideration for the tourist recommendation that the costs trigger when a tourist selects a long-distance pattern to visit different POIs.

We will minimize traveling time by utilizing quicker modes of transportation. If the gap between the two POIs increases and in that situation traveling costs will therefore arise, a quicker form of travel is necessary. We however intended to minimize the length of the tour. The transportation costs are calculated by Eqn.6:

$$\Gamma^{cost}(\dot{x}) = \sum_{\dot{y}=1}^{\dot{n}} \sum_{\jmath=2}^{\dot{y}} \gamma^{intr}(\mathcal{P}^{\mathcal{P}_{\jmath-1,\jmath}}_{\dot{y}}) + \sum_{\dot{y}=1}^{\dot{n}} \gamma^{extr}(\mathcal{P}^{\mathcal{P}_n}_{\dot{y}}, \mathcal{P}^{\mathcal{P}_1}_{\dot{y}+1}). \tag{6}$$

The first portion of Eqn. 6 consists of the inner length of all POIs in the pac plan. The inner length of the POI $\mathcal{P}_{\dot{y}}$ is calculated by the total length of all POIs. The second portion of Eqn. 6 reflects the consecutive real length among $\mathcal{P}_{\dot{y}}$ and $\mathcal{P}_{\dot{y}+1}$ POIs, which could be calculated by taking account of the length between the two consecutive POIs.

3.8 Problem Definition

This portion deals with the problem of different POIs for one person. Our main objective is to maximize the interest and popularity of visitors and to reduce expenditures. A type of orienteering issue [12] could be used to resolve this issue:

$$\mathbb{O}_{\dot{y}} = \frac{\left(\Theta\mathcal{P}_{\dot{y}}(\overline{int}) + (1-\Theta)\mathcal{P}_{\dot{y}}(\overline{pop})\right) + W(inte)}{Cost(\mathcal{P}_{\dot{y}})}. \tag{7}$$

Our aim here is to propose an itinerary focused on tourist interest for specific POI , tour popularity and weather interest. The weight parameter can be adjusted as needed. The key goal of this arch is to propose an itinerary to increase the interest, popularity of visitors, weather interest and minimize travel expenses. The user's interest in various weather conditions, such as winter, summer, and so on, is used to quantify the weather interest, which is denoted by $W(inte)$. For eg, if a visitor prefers to visit a location in the winter, this indicates that the atmosphere is appealing to him.

The aim is to find an itinerary tour plan I= ($\mathcal{P}_1, \mathcal{P}_2 ..., \mathcal{P}_{\dot{n}}$) that:

$$Max(\frac{\left(\Theta\mathcal{P}_{\dot{y}}(\overline{int}) + (1-\Theta)\mathcal{P}_{\dot{y}}(\overline{pop})\right) + W(inte)}{Cost(\mathcal{P}_{\dot{y}})}). \tag{8}$$

Let $T_{\mathcal{P},\mathcal{P}'} = 1$, if the traveler has explored the POIs $\mathcal{P}$ and $\mathcal{P}'$ sequentially. That means a tourist can travel from $\mathcal{P}$ to a $\mathcal{P}'$. $T_{\mathcal{P},\mathcal{P}'} = 0$, otherwise [11]. Then Eqn. 8 could be overcome with 19-22 restrictions:

$$\sum_{\mathcal{P}'=2}^{\mathbb{N}} T_{1,\mathcal{P}'} = \sum_{\mathcal{P}=1}^{\mathbb{N}-1} T_{\mathcal{P},\mathbb{N}} = 1, \tag{9}$$

$$\sum_{\mathcal{P}=1}^{\mathbb{N}-1} T_{\mathcal{P},k} = \sum_{\mathcal{P}'=2}^{\mathbb{N}} T_{k,\mathcal{P}'} \leq 1; \forall\, k = 2, ..., \mathbb{N}-1, \tag{10}$$

$$2 \leq L_{\mathcal{P}'} \leq \mathbb{N}; \forall\, \mathcal{P}' = 2, ..., \mathbb{N}, \tag{11}$$

$$L_{\mathcal{P}} - L_{\mathcal{P}'} + 1 \leq (\mathbb{N}-1)(1 - T_{\mathcal{P},\mathcal{P}'}) \forall\, \mathcal{P}, \mathcal{P}' = 2, ..., \mathbb{N}, \tag{12}$$

$$|cost(x)| \leq \mathbb{B}. \tag{13}$$

In Eqn. 8, the multi-objective issue is solved by increased visitor interest and popularity, weather interest and a decreased expense. The limitation set out in Eqn. 9 means that the proposed plan should be started from the first POI and the last POI can be completed. The limitation provided in Eqn. 10 shows that there is no POI viewed multiple times.

The restrictions imposed in Eqn. 11 and 12 argue that another response would not include a sub-tour based on the issue of the traveler with the sub-tour eliminating problem 13. Section 15 guarantees that the overall gap for the kit is provided by budget $\mathbb{B}$. $L_{i'}$ represents the i'-th itinerary herein.

The issue is an NP-hard problem since it relies on the cost function $Cost\ (\mathcal{P}_1 ... \mathcal{P}_{\dot{n}})$. This is also affected by the multiple POIs selected from a wide variety of variants. To address these problems, we are proposing the wPOIapproach is dependent on Monte-Carlo Tree Search (MCTS) and is described herein.

3.9 Monte Carlo Tree Search Algorithm

The Monte Carlo Tree Search (MCTS) method is used in board games including Othello, Chess, and Go [5, 28]. The MCTS algorithms are based on the tree search idea. All the boards in the board graph are referred to as the node and the game's result score is considered the leaf node. The gameplay can take multiple runs to hit the endpoint based on the MCTS formula.

Any execution starts with a set of randomized nodes and reports the results. The following steps are performed to get the winning score. A set of trials have been conducted in MCTS (e.g. 100 trials) and it is repeated for a specific duration (e.g. 10 seconds). The below are the basic core tasks of MCTS:

1. **Selection**: Let θ be root and spread into a randomized selection of the child node t with defined criteria to reach the end/leaf node. Likewise, θ be the initial/ root for the board, the t child's node, and the present condition for the board.

2. **Expansion**: Any child node can be expanded to a leaf node by using a randomized collection of the unexplored child node.

3. **Simulation**: At certain instances up to the end of the game the first and second moves are replicated.

4. **Back-propagation**: Phases 1-4 lead to one MCTS run when crossing the root of the leaf node. Also, every crossed node is labeled with a win / lose (1/0), on every run (back-propagation). The procedure is repeated over a certain amount of cycles.

MCTS solution is used for various networking problems, such as the issue of travel salesmen [18] or car navigation [9]. This issue is fixed based on two main factors by the use of the MCTS approach [5].

1. Rather than finding the whole tree which reduces operating time, MCTS often recognizes specific areas with the best probability of the remedy.

2. It could be configured for practical uses with a specific iteration.

Due to the below reasons – MCTS should not be directly implemented:

1. The expense of the decisions on a tour depends on the length of the tourist traveled and the spending time on every board game find, and the varying cost of it. 1.

2. The bonus score is in the win/loss state in the board game, either 1 or 0. Similarly, the arrangement of the award is quite difficult due to various traveling costs the popularity of the tour, and the degree of interest to visitors for the proposed POIs in the case of the routing advisory scheme.

The Upper Confidence Bound (UCT) is used to navigate the POI $\mathcal{P}_{\dot{n}}$, which maximizes Eqn. 14:

$$UCT_{\mathcal{P}_j}^{original} = \frac{\mathbb{T}_{\mathcal{P}_j}^{Reward}}{\mathbb{V}_{\mathcal{P}_j}^{Count}} + 2\mathbb{C}\sqrt{\frac{2ln\ \mathbb{V}_{\mathcal{P}_{j'}}^{Count}}{\mathbb{V}_{\mathcal{P}_j}^{Count}}}. \quad (14)$$

The initial UCT (Eqn. 14) is the improvement of our suggested technique where a probabilistic selection is made for the succeeding POI and is described in Eqn. 15.

$$UCT_{\mathcal{P}_j}^{wPOI} = \left(\frac{\overline{Int}(\mathcal{P}) + \overline{pop}(\mathcal{P})}{\Gamma^{cost}(x)}\right) + \frac{\mathbb{T}_{\mathcal{P}_j}^{Reward}}{\mathbb{V}_{\mathcal{P}_j}^{Count}} + 2\mathbb{C}\sqrt{\frac{2ln\ \mathbb{V}_{\mathcal{P}_{j'}}^{Count}}{\mathbb{V}_{\mathcal{P}_j}^{Count}}}, \quad (15)$$

where $\mathbb{V}^{Count}$ denotes is the number of visits of viewed nodes and $\mathbb{T}^{Reward}$ Reward is the cumulative reward from proposed POIs.

3.10 Simulation and Back-Propagation

Inside the MCTS, the test started at the root node, evaluating the reward as 1 (win) and 0 (loss). For binary 1 and 0 numbers, the reward for some POIs is not fully defined. The reward depending, on various parameters including tourist interest, tour popularity, and travel expenses, and is described as:

$$Reward = \frac{\left(\overline{Int}(\mathcal{P}) + \overline{pop}(\mathcal{P})\right)}{\Gamma^{cost}(\dot{x})}. \quad (16)$$

For each loop, the reward score is computed by Eqn. 16. If the itinerary is successful, the reward score will be back-propagated to all viewed nodes Furthermore, the number of trips is replicated and then raised by one.

4 Experimental Methodology

4.1 Dataset

In this analysis, we utilized the data provided in [12]. The dataset contains images and videos by Yahoo! Flickr Creative Commons 100M (YFCC100M) [25]. Furthermore, the YFCC100M data set provided in [12] was used and geo-tagged images from different areas of the globe have been obtained. The data set comprises the photo's meta-data. It includes visiting dates and times. The dataset also contains data from the

Geo-coordinate to identify the length among POIs. The data sets utilized in this research could be accessed[1] freely.

4.2 Baseline Algorithms

Based on the work [12] we have taken into account all the benchmark algorithms beginning at one POI and then choosing the next following POI before the budget is achieved. We utilize the tour series by the tourist to suggest multiple POIs.

— **Greedy Nearest (GNEAR)**: We utilize this algorithm for our next unexplored POI by selecting the three closest attractions [15].

— **Greedy Most Popular (GPOP)**: By picking the three most popular attractions, we select an unvisited POI [15].

— **Tour Recommendation With Interest Category (TOURINT)**: This compulsory group is described as the most frequently viewed group in several tourist visits [11]. This shows the issue with a suggested tour with a compulsory group that the visitor can explore at least once on the proposed itinerary.

— **Trip Builder (TRIPBUILD)**: This builds a personalized tourist itinerary according to the attraction's interest and popularity. An interest in a POI will be calculated as the number of the POI visits in a certain group compared with his/her overall visit [4].

4.3 Real-life Evaluation

Only visitors who have completed at least two travel sequences and two groups are assessed. The method is applied to both local and global datasets [24], as well as visitors who are comparable. We compare similar visitors in this study by looking at the top 10 associated visitors from global data sets. To equate different baselines with our method, we chose the preceding formulas. For our experiments, categories of real traveling series are chosen based on the history of associated visitors in a given area.

[1] https://sites.google.com/site/limkwanhui/datacode?authuser=0

— **Tour Recall (TourRec(I))**: The $Tour\ Recall$ is identified as the section of the actual tourist's series still portion of the suggested POI C_{rec} is supposed to be a list of recommended groups. C_{real} presents in its real-life tourism series a set of categories visited by a tourist. Eqn. 17 describes the $Tour\ Recall$:

$$TourRec(I) = \frac{|C_{rec} \cap C_{real}|}{|C_{real}|}. \tag{17}$$

— **Tour Precision (TourPre(I))**: The $Tour\ Precision$ is defined in the I itinerary as the proportion of proposed categories still part of the tourist's actual life. C_{rec} is assumed to include a list of categories suggested. C_{real} is a list of categories seen in his traveling sequence by a traveler. As displayed in Eqn. 18, $Tour\ Precision$ has been represented:

$$TourPre(I) = \frac{|C_{rec} \cap C_{real}|}{|C_{rec}|}. \tag{18}$$

— **Tour F1 (TourF1(I))**: the mean harmonic value of $Precision$ and $Recall$ for the proposed itinerary I is termed $Tour\ F1-Score$ available in Eqn. 19:

$$TourF1(I) = \frac{2 \times TourPre(I) \times TourRec(I)}{TourPre(I) + TourRec(I)}. \tag{19}$$

4.4 Comparison of $Precision$, $Recall$ and $F1$

The performance of the $wPOI$ algorithm is higher than other baseline algorithms such as GPOP TOURINT and GNEAR. $wPOI$ is more effective than those baseline approaches like GPOP TOURINT and GNEAR. The results are more efficient. Tables 1, 2 and 3 indicate the $Precision$, $Recall$ and $F1$-$Score$ measurements to represent the variations among $wPOI$ and other baseline approaches.

The results show that the $Precision$, $Recall$ and $F1$-$Score$ metrics are more significant for $wPOI$ than the baseline approaches. The $Recall$ value changes for $wPOI$ approach range from 3.4%-22.6% compared to other baseline

Table 1. Comparison of $Precision$ between our proposed approach and other baseline methods

Algorithms	wPOI	TOURINT	GPOP	GNEAR	RAND
Delhi- Edinburgh	0.404±0.037	0.353±0.038	0.321±0.029	0.294±0.024	0.265±0.035
Osaka-Edinburgh	0.389±0.014	0.34±0.038	0.314±0.03	0.286±0.023	0.26±0.013
Vienna-Edinburgh	0.39±0.038	0.359±0.022	0.325±0.017	0.293±0.048	0.275±0.047
Delhi-Osaka	0.71±0.019	0.56±0.025	0.536±0.038	0.614±0.014	0.421±0.026
Glasgow-Edinburgh	0.404±0.037	0.356±0.019	0.341±0.029	0.286±0.044	0.261±0.027

Table 2. Comparison of $Recall$ between our proposed approach and other baseline methods

Algorithms	wPOI	TOURINT	GPOP	GNEAR	RAND
Delhi- Edinburgh	0.362±0.023	0.31±0.05	0.293±0.043	0.259±0.014	0.224±0.019
Osaka-Edinburgh	0.382±0.039	0.327±0.022	0.291±0.032	0.255±0.036	0.236±0.017
Vienna-Edinburgh	0.372±0.033	0.326±0.024	0.302±0.015	0.279±0.031	0.256±0.043
Delhi-Osaka	0.396±0.009	0.333±0.018	0.292±0.042	0.271±0.027	0.229±0.035
Glasgow-Edinburgh	0.365±0.023	0.308±0.034	0.288±0.05	0.269±0.016	0.231±0.039

Table 3. Comparison of $F1 - Score$ between our proposed approach and other baseline methods

Algorithms	wPOI	TOURINT	GPOP	GNEAR	RAND
Delhi- Edinburgh	0.382±0.04	0.33±0.049	0.306±0.011	0.275±0.025	0.243±0.016
Osaka-Edinburgh	0.385±0.018	0.333±0.045	0.302±0.021	0.269±0.015	0.248±0.016
Vienna-Edinburgh	0.381±0.016	0.341±0.041	0.313±0.026	0.286±0.034	0.265±0.02
Delhi-Osaka	0.388±0.049	0.34±0.034	0.304±0.045	0.283±0.048	0.247±0.043
Glasgow-Edinburgh	0.384±0.028	0.33±0.011	0.313±0.021	0.277±0.017	0.245±0.018

approaches (see 2). $Recall$ measurements depending upon $|C_v|$ and $|C_{rec} \cap C_{real}|$ as per in Eqn. 17.

Here the values of $|C_{rec} \cap C_{real}|$ is better compared to the various baseline approaches which can be computed utilizing the $wPOI$ algorithm. Typically, the suggested $wPOI$ algorithm is based on two datasets, local and global, and ultimately suggests many POIs, resulting in better $Recall$ scores for various baseline approaches.

Taking into consideration the tour popularity or interest of visitors, the various baseline approaches such as GPOP, TOURINT and GNEAR do not assist weather interest. In contrast with other baseline approaches, the increase in $Precision$ scores for the $wPOI$ algorithm suggested is 3.4%-22.6%. For the $wPOI$ algorithm, the $Precision$ scores are more because they are dependent on $|C_{rec}|$ and $|C_{rec} \cap C_{real}|$ as per in Eqn. 18.

We found that C_{rec} scores vary for various baseline approaches during the analysis. The values of $|C_{rec} \cap C_{real}|$ are higher for the suggested $wPOI$ algorithm. The $F1$-$Score$ increase in the suggested wPOI algorithm based

on $Precision$ and $Recall$, from 3.4%-22.6% compared to other baseline approaches.

5 Conclusion and Future Work

In this research, we have offered a method $wPOI$ that contributes to maximise the tourist interest, popularity,weather interest and reduced costs. Geo-tagged photos are used by $wPOI$ to show the tourists' actual travel patterns. Tourist interest, tour popularity, weather interest and traveling costs are calculated effectively for training the $wPOI$ algorithm.

The suggested method is dependent on the selection of many POIs by taking into account the POI time visiting factor. $wPOI$ will not depend on the traveling history of a certain individual in new locations.

The case in which a visitor wants to visit new places is therefore taken into consideration. b) tourist has many POIs (c) the weather interest is calculated . Given the Flickr data in several cities, we matched $wPOI$ with various baselines that take multiple criteria such as $Precision$, $Recall$, and F1-Score.

The findings of the study demonstrate that the suggested $wPOI$ algorithm in most situations surpasses baseline approaches. We wish to enhance this research in the future to several travelers who intend to be staying in a new location over many days.

References

1. **Aliannejadi, M., Crestani, F. (2018).** Personalized context-aware point of interest recommendation. ACM Trans. Inf. Syst., Vol. 36, No. 4, pp. 45:1–45:28.
2. **Anagnostopoulos, A., Atassi, R., Becchetti, L., Fazzone, A., Silvestri, F. (2017).** Tour recommendation for groups. Data Min. Knowl. Discov., Vol. 31, No. 5, pp. 1157–1188.
3. **Borras, J., Moreno, A., Valls, A. (2014).** Intelligent tourism recommender systems: A survey. Expert Systems with Applications, Vol. 41, No. 16, pp. 7370–7389.
4. **Brilhante, I. R., de Macêdo, J. A. F., Nardini, F. M., Perego, R., Renso, C. (2014).** Tripbuilder: A tool for recommending sightseeing tours. Advances in Information Retrieval - 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings, pp. 771–774.
5. **Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., Colton, S. (2012).** A survey of monte carlo tree search methods. IEEE Transactions on Computational Intelligence and AI in Games, Vol. 4, No. 1, pp. 1–43.
6. **Castillo, L., Armengol, E., Onaindía, E., Sebastiá, L., González-Boticario, J., Rodríguez, A., Fernández, S., Arias, J. D., Borrajo, D. (2008).** Samap: An user-oriented adaptive system for planning tourist visits. Expert Syst. Appl., Vol. 34, No. 2, pp. 1318–1332.
7. **Chen, C., Zhang, D., Guo, B., Ma, X., Pan, G., Wu, Z. (2015).** Tripplanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints. IEEE Transactions on Intelligent Transportation Systems, Vol. 16, No. 3, pp. 1259–1273.
8. **Gunawan, A., Lau, H. C., Vansteenwegen, P. (2016).** Orienteering problem: A survey of recent variants, solution approaches and applications. European Journal of Operational Research, Vol. 255, No. 2, pp. 315–332.
9. **Kenyon, A. S., Morton, D. P. (2003).** Stochastic vehicle routing with random travel times. Transportation Science, Vol. 37, No. 1, pp. 69–82.
10. **Li, X., Jiang, M., Hong, H., Liao, L. (2017).** A time-aware personalized point-of-interest recommendation via high-order tensor factorization. ACM Trans. Inf. Syst., Vol. 35, No. 4, pp. 31:1–31:23.
11. **Lim, K. H. (2015).** Recommending tours and places-of-interest based on user interests from geo-tagged photos. Proceedings of the 2015 ACM SIGMOD on PhD Symposium, ACM, New York, NY, USA, pp. 33–38.
12. **Lim, K. H., Chan, J., Karunasekera, S., Leckie, C. (2017).** Personalized itinerary recommendation with queuing time awareness. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, ACM, New York, NY, USA, pp. 325–334.
13. **Lim, K. H., Chan, J., Leckie, C., Karunasekera, S. (2016).** Towards next generation touring: Personal-

ized group tours. Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, ICAPS 2016, London, UK, June 12-17, 2016., pp. 412–420.

14. **Lim, K. H., Chan, J., Leckie, C., Karunasekera, S. (2018).** Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. Knowl. Inf. Syst., Vol. 54, No. 2, pp. 375–406.
15. **Lim, K. H., Wang, X., Chan, J., Karunasekera, S., Leckie, C., Chen, Y., Loong Tan, C., Quan Gao, F., Ken Wee, T. (2016).** Perstour: A personalized tour recommendation and planning system. pp. .
16. **Lucchese, C., Perego, R., Silvestri, F., Vahabi, H., Venturini, R. (2012).** How random walks can help tourism. Proceedings of the 34th European Conference on Advances in Information Retrieval, ECIR'12, Springer-Verlag, Berlin, Heidelberg, pp. 195–206.
17. **Luigi, D., Fuciu, M. (2015).** Consumer behaviour in the tourist segmentation process – a marketing research. Studies in Business and Economics, Vol. 10, pp. 66–76.
18. **Miller, C. E., Tucker, A. W., Zemlin, R. A. (1960).** Integer programming formulation of traveling salesman problems. J. ACM, Vol. 7, No. 4, pp. 326–329.
19. **Quercia, D., Schifanella, R., Aiello, L. M. (2014).** The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. Proceedings of the 25th ACM Conference on Hypertext and Social Media, HT '14, ACM, New York, NY, USA, pp. 116–125.
20. **Ramasamy, V., Gomathy, B., Obulesu, O., Sarkar, J., Panigrahi, C., Pati, B., Majumder, A. (2020).** Machine Learning Techniques and Tools: Merits and Demerits. pp. 23–55.
21. **Sarkar, J. L., Majumder, A. (2021).** A new point-of-interest approach based on multi-itinerary recommendation engine. Expert Systems with Applications, Vol. 181, pp. 115026.
22. **Sarkar, J. L., Majumder, A. (2022).** gtour: Multiple itinerary recommendation engine for group of tourists. Expert Systems with Applications, Vol. 191, pp. 116190.
23. **Sarkar, J. L., Majumder, A., Panigrahi, C. R., Ramasamy, V., Mall, R. (2021).** Triptour:a multi-itinerary tourist recommendation engine based on poi visits interval. 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–7.
24. **Sarkar, J. L., Majumder, A., Panigrahi, C. R., Roy, S. (2020).** Multitour: A multiple itinerary tourists recommendation engine. Electronic Commerce Research and Applications, Vol. 40, pp. 100943.
25. **Thomee, B., Shamma, D., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.-J. (2016).** The new data and new challenges in multimedia research. Communications of the ACM, Vol. 59, No. 2, pp. 64–73.
26. **Vansteenwegen, P., Oudheusden, D. V. (2007).** The mobile tourist guide: An or opportunity. OR Insight, Vol. 20, No. 3, pp. 21–27.
27. **Vansteenwegen, P., Souffriau, W., Berghe, G. V., Oudheusden, D. V. (2011).** The city trip planner. Expert Syst. Appl., Vol. 38, No. 6, pp. 6540–6546.
28. **Vansteenwegen, P., Souffriau, W., Oudheusden, D. V. (2011).** The orienteering problem: A survey. European Journal of Operational Research, Vol. 209, No. 1, pp. 1–10.
29. **Wörndl, W., Hefele, A. (2016).** Generating paths through discovered places-of-interests for city trip planning. Information and Communication Technologies in Tourism, pp. 441–453.
30. **Ying, H., Wu, J., Xu, G., Liu, Y., Liang, T., Zhang, X., Xiong, H. (2018).** Time-aware metric embedding with asymmetric projection for successive poi recommendation. World Wide Web, Vol. 22, No. 5, pp. 2209–2224.
31. **Zhao, S., Zhao, T., Yang, H., Lyu, M. R., King, I. (2016).** Stellar: Spatial-temporal latent ranking for successive point-of-interest recommendation. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16, AAAI Press, pp. 315–321.

Article received on 20/08/2021; accepted on 08/11/2021.
Corresponding author is Rajani Trivedi.

ISSN 2007-9737

Design and Analysis of a New Reduced Switch Scalable MIN Fat-Tree Topology

Abhijit Biswas[1], Anwar Hussain[2]

[1] Assam University,
Department of CSE, Silchar,
India

[2] North Eastern Regional Institute of Science and Technology,
Department of ECE,
India

abhi.021983@gmail.com, ah@nerist.ac.in

Abstract. This paper presents a reduced switch scalable MIN Fat-Tree with new inter-router connections. Unlike the conventional Fat-Tree, consecutive switches are connected with new bidirectional links for faster intergroup communication. In case of traditional Fat-Tree, a packet has to travel upward upto the level called summit from where the downward path to the client is available, most often the summit is at the highest router level of the MIN Fat-Tree. The proposed topology aims at lowering the summit by eliminating the entire topmost switch level, thereby reducing the size of the network significantly without considerable degradation in the network performance and maintaining the scalable property of the MIN Fat-Tree. The results indicate that the proposed network can not only reduce the delay but also reduce the number of switches to a great extent. It was found that around 33.33% of switches were reduced in the proposed network for eight clients and 14.28% for 128 clients.

Keywords. Network-on-chip, Fat-Tree, switch, topology, end-to-end delay, hop count, routing algorithm.

1 Introduction

The inception of network-on-chip (NOC) has created a new paradigm for on-chip communications. The major reasons for this acceptability are its high scalability, better latency, high throughput and less area consumption [1].

However, each of these factors have a large dimension for improvement and research. Among all the topologies designed for NOC, the mesh topology has gained a lot of attention. The popularity was because it could accommodate a good number of clients and also for its seamless scalable property.

Besides, as every switch had at least one path to every other switch the routing was quite simple and easy to simulate. But later researches showed that with the increase in network size, there was a high probability for packet loss [17, 15, 8, 20].

Also, as every switch had only one client connected to it, scalability often resulted into performance degradation [16, 4]. In order to conceal these issues, the routing was improved to a larger extent, but still it always had some loophole due to which an alternative was of utmost need. In 1985, Charles E. Leiserson proposed the Fat-Tree topology which is more like a real tree [18].

Unlike the conventional trees used in computer science, the link size of the Fat-Tree gets thicker as one traverse from leaves towards the root. Charles E. Leiserson in [18] has proved the universality of the Fat-Tree network by offline simulation that for any network size, Fat-Tree gives the best routing and efficient utilization of hardware. Network-on-chip being a real time

scenario, the parallel computation has always been an important factor.

Implementing the Fat-Tree for network on a chip has proved to be a major achievement in addressing issues like scalability, high performance, modular design etc. A. Bouhraoua et al. [14, 10, 9, 11, 12] restructured the Fat-Tree proposed by Leiserson into more efficient one by including new downlinks that could rectify the contentions of the conventional Fat-Tree.

But, this version had more number of links at the bottom level switches due to which additional chip area is required to accommodate all the extra wirings. Hence, as the topology grew, the size of buffers also increased, consuming a large area of the chip. They also presented a heterogeneous architecture [13] of their modified Fat-Tree where the downlinks were of different bandwidths. These architecture was designed to meet the real time scenario of communication where the clients may have different individual bandwidths.

This architecture reduced the wastage of bandwidths and enabled the clients to receive packets through multiple options. For optimal utilization, the packets were broken to smaller segments called flits. However, this only resulted into requirement of a large number of buffers to store these flits. Another improved version of the Fat-Tree was presented by Gomez et al. named as Reduced Unidirectional Fat-Tree (RUFT) with non-minimal routing [23]. This routing scheme forced the packets to travel till the last stage of the network prior to travelling downwards towards the clients. As a result, even if the communication was to be made between adjacent clients, the packet had to travel an unnecessary distance till the last stage resulting unwanted delay.

In this paper, we have presented a further improved version of the Fat-Tree which addresses the previous issues of Bouhraoua and Gomez. Our proposed architecture uses buffer in the form of hierarchical ports to store the flits of a packet to implement flit wise communication. Further, in order to minimize the hardware used to realize the network, new links have been introduced between the switches which connects these clients. With these architectural changes and improvised routing, the proposed Fat-Tree is simulated to see the route ability of the network.

2 Related Work

Out of the various dimensions for research in network-on-chip, topological challenges and routing has taken the centre stage. The topology deals with how to make NOC compatible with the area of the chip by carefully placing the IP cores(Intellectual properties) along with network nodes(routers or switches), the routing deals with ways to gain better performance and throughput with minimum latency and packet loss by carefully choosing the path from source to destination in any given topology. Combined together, the topology and routing makes an architecture.

Guerrier et al. [16] presented an architectural overview of scalable interconnections where they also illustrated how the shared bus architecture would fail to meet the requirements of future interconnections. They proposed a state-of-art for a packet switched interconnection with wormhole routing for better performance. However, during analysis it was seen that there was an increase in the latency with the network load, resulting in performance degradation which was not acceptable. Couple of years later, Shashi Kumar et al. [17] came up with a m x n mesh network where every switch was connected to four other adjacent switch providing better interconnection. But, even this network failed to take network load beyond 50% due to packet loss and latency.

In order to meet the synchronization on the chip, Sheibanyrad et al. [24] came up with a globally synchronous and locally asynchronous (GALS) network using hybrid FIFOs. These FIFOs were used for maximizing throughput and minimizing the probability of metastability. Although there was a large latency, yet the model achieved high throughput. A topology adaptive network was proposed by Bartic et al. [3] which provided scalability and flexibility to choose both network as well as the routing. Based on virtual cut through switching, this design could lower the latency to a good extent.

Relevant to our proposed work, Bouhraoua et al. [14, 10, 9, 11, 12] have made significantly

contributed in restructuring the conventional Fat-Tree network. The original Fat-Tree designed by Leiserson had contentions in the downward links.

The Fat-Tree structure proposed by Bouhraoua et al. [14] doubled the downward links of the original Fat-Tree resulting increase in the number of links from top to bottom. In order to store the large number of packets at the lowest levels, additional FIFOs were introduced. Though this modified Fat-Tree rectified the contentions of the original tree, but due to FIFOs, a large amount of chip area had to be consumed. The large number of downlinks increases the bandwidth, but in real time, only few of this bandwidth was actually used. In order to meet this wastage, Bouhraoua et al. [11] introduced heterogeneous bandwidth in the downlinks and also broke the packets into smalle segments.

Although, this reduced the wastage of bandwidth but also resulted into increase in the FIFOs and hence the constraint on area still prevailed. Later, Bouhraoua et al. [12] improvised their earlier design of modified Fat-Tree along with new routing strategy based on clusters. But hardware utilized remained unchanged. They simulated the network for uniform and non-uniform addresses.

Although, the average latency increased but the network did not reach any saturation level which prevailed in earlier designs. In contrast to mesh topology which saturated at 30% injection rate, their improved modified Fat-Tree did not saturate till 90% injection rate.

But, with their routing strategy, there was always an unwanted delay even when the adjacent client communicated with each other. Similar issue was found in the RUFT network proposed by Gomez et al. [23]. The network defined by them had unidirectional multistage interconnections which rectified the downwards contentions of the original Fat-Tree. But, due to non-minimal routing, each packet has to reach the last stage of the network before moving downwards unlike in [12], where the downward path is available from a level of router which are called summit, which may or may not be the highest level of router in the Fat-Tree topology.

In [21], an extended zoned node is presented which promotes the usage of extra links with an aim to create connected layers and to increase the bisection bandwidth of the network and to provide fault tolerance. The extended zoned nodes does not minimize the hardware used rather is deployed to make super nodes to cater the needs of high throughput and minimized latency.

In [2], a SMBFT is proposed, which minimizes the hardware used in a BFT and has proven to be efficient but at the cost of complicating the router design by adding multiple extra links per router as opposed to the proposed work which only added a single lateral link per router in the network.

A partial group based routing for the modified Fat-Tree was presented by Biswas et al. [7] where the routing did not necessarily travel the last level of the network prior to traveling downwards till the destination clients.

In their work they partitioned the complete network into two equal groups and for reaching the destination client the packets only had to travel to the highest level if the destination client was in a different group. This new routing strategy addressed the unwanted delay which was a major disadvantage in Bouhraoua et al. [14, 10, 9, 11, 12] and Gomez et al. [23].

When compared with the original Fat-Tree, the routing approach of Biswas et al. [7] could reduce the routing activities from 8% to 13%. Later a fully grouped routing strategy appeared in [5] where Biswas et al. extended the number of groups from 2 to network dependent. This means, as the network size increased, the number of groups increased too.

Both these routing strategies in [7] and [5], homogeneous modified Fat-Tree was implemented. Both reduced the unwanted delay for adjacent clients to communicate but it was not eliminated as the packets still had to travel at least to a next higher level before traveling downwards till the destination client.

In an attempt to address all these issues in the modified Fat-Tree, we in the coming section present our proposed Reduced Switch Scalable MIN Fat-Tree topology. For better understanding of the proposed topology next section describes the MIN Fat-Tree and Partial group based routing in homogeneous Fat-Tree topology.

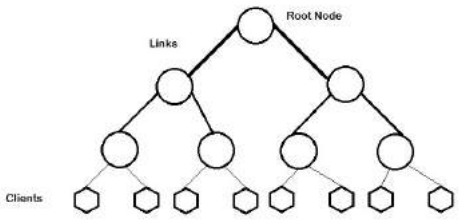


Fig. 1. Fat-Tree Topology: Fat-Tree as proposed by Leiserson

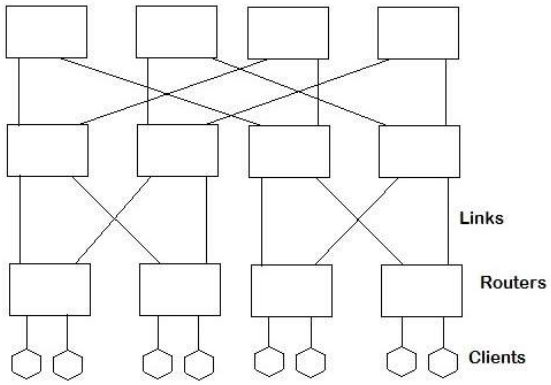


Fig. 2. Fat-Tree Topology: 2-ary 3-tree

3 Fat-Tree Topology

A generalized definition of Fat-Tree states that [22, 25] a Fat-Tree is a collection of vertices connected by edges and is defined recursively as:

(i) A single vertex by itself is a Fat-Tree. This vertex is also the root of the Fat-Tree.

(ii) If $v_1, v_2, ..., v_i$ are vertices and $T_1, T_2, .., T_j$ are fat-trees, with $r_1, r_2, .., r_k$ as roots (j and k need not to be equal), a new fat-tree is built by connecting with edges, in any manner, the vertices $v_1, v_2, .., v_i$ to the roots $r_1, r_2, .., r_k$. The roots of the new Fat-Tree are $v_1, v_2, .., v_i$.

The above definition is very general and covers almost all trees in text. A general Fat-tree as proposed by Leiserson and described above is depicted in Figure 1. Notice that in Figure 2, the links from the lower levels getting thicker and thicker as we climb up in the hierarchy. Due the limitation of a single root Fat-Tree, the attention from this type of network has shifted to another class of Fat-Trees called k-ary n-trees. The general k-ary n-trees have been borrowed from another popular multistage interconnection networks known as k-ary n-fly or popularly known under the text as butterfly networks [5, 6]

3.1 Routing in Fat-Tree topology

As described by Bouhraoua and Elrabba in [14, 10, 9, 11, 12], the Fat-Tree in the above figure 1(b) is characterized the routing of the packets from source to destination which is similar to a binary tree. The packet from the source is routed in the upwards direction until a router is reached which has the downward connection to the destination. This router is termed as “routing summit”.

An n row network has 2^n number of clients. A router in the above depicted network is indexed as (r,c) where r is the row index and c is the column index of the router. The connection of the routers are such that, a router (r,c) is connected to two routers given as [14, 10, 9, 11, 12]

— Router (r+1,c),

— Router (r+1,c-2^r) or (r+1,c+2^r) depending upon the value c/2^r is odd or even.

The address of the clients is in the interval [0,2^n-1] and the relation between a client address and router column is given by $addr = 2 \times c + s$, where s = $\{0,1\}$, i.e., s = 0 for clients attached to the left link of the lowest level router and s = 1 for clients attached to the right link of the router.

The reach range of a router is defined as the number of client addresses it can reach. The network architecture dictates that the reach range of a router (r,c) is divided into two intervals I_L the left interval given by I_L = [2c_L,2c_L+2^r-1] and I_R the right interval given by I_R =[2c_L+2^r,2c_L+2^{r+1}-1], where c_L and c_R is the leftmost column index and right most column index of the groups G_L and G_R respectively to which a router (r,c) is connected to.

A group here is defined as the lower level routers to which a upper level router (r,c) is connected to. The lower level router connected by the left link of

the router (r,c) is called the left group G_L and the lower level router connected by the right link of the router (r,c) is called the right group G_R . It is worth mentioning here that the reach range of topmost routers is the entire client set.

3.2 Partial Group Based Routing for Homogeneous Fat-Tree Topology

A Partial Group Based Routing Algorithm is described here for the MIN Fat-Tree topology [7]. For the sake of scalability the original MIN Fat-Tree structure of Figure 1(b) shown above has not been changed. The routing for the original MIN Fat-Tree presented in [14] is based on the calculation of right and left intervals in every router whenever a packet is received by it for routing decisions. The calculation of the right and left interval is explained above.

The partial group based routing is an attempt of reducing the calculation of these intervals by routers for only half of the time in each direction of the packet flow. This objective is achieved by logically grouping the entire MIN Fat-Tree into two halves. Each half of the topology will contain routers and clients. The topmost routers of the topology do not belong to any group and serve to identify the proper group of a destination client while routing decision takes place.

The routing is designed such that, the original routing of the Fat-Tree remains the same while routing in the same group. Whenever an intergroup routing decision is to be taken the partial group based routing kick starts. The figure below shows the logical grouping of an eight client MIN Fat-Tree topology.

The figure above shows that any MIN Fat-Tree network of n clients will be divided dynamically into two groups with each group having $\frac{n}{2}$ number of clients in it. The groups are numbered as 1 and 2 from left to right. The clients are numbered from 0 to n-1 from left to right, making the first group contain 0 to $\frac{n}{2}$-1 clients and the remaining $\frac{n}{2}$ to n-1 client is contained in the next group.

Whenever a source sends a message from one group to the another, the packets are straight forwardly routed to the topmost level where depending upon the group id of the destination client the packet is sent downwards either by the right link or the left link of the topmost router.

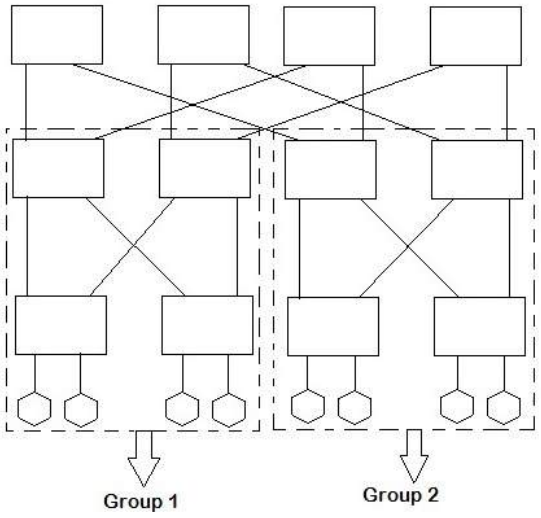


Fig. 3. Partial Group Based Routing in MIN Fat-Tree

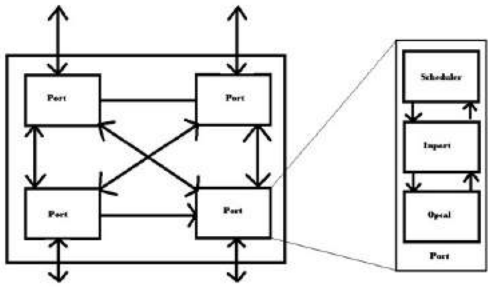


Fig. 4. Port structure of Partial Group Based Routing in MIN Fat-Tree

The general convention of binary tree is followed where, if the destination group id is smaller than the source group id, the packet is forwarded via the left link of the topmost routers otherwise the packet is forwarded by the right link of then topmost routers. Assigning group id dynamically is straight forward and given by:

$$c_{assign\ gid} = \begin{cases} 1, & \text{for c} \in \{0, .., \frac{n}{2} - 1\}, \\ 2, & \text{for c} \in \{\frac{n}{2}, ..., n - 1\}. \end{cases} \quad (1)$$

where c is the client such that 0≤c≤n-1.

For the simulation purpose, a logical router is designed having four hierarchical ports fully connected to each other using bidirectional links. The port is the place within the router where the routing decision takes place. A port within the router contains three parts, a *Scheduler*, an *Inport*, and an *Opcal*. The router structure and a port structure are depicted below for better understanding.

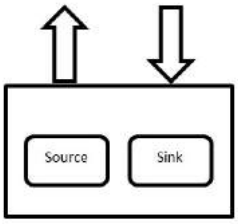


Fig. 5. A logical client router interface: Client structure

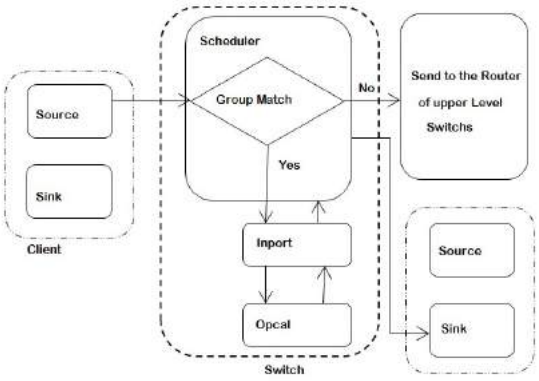


Fig. 6. A logical client router interface: Client router interface

The group id of the router is given such that, any router with in column 0 to $\frac{n}{2}$-1 are group 1 routers and the routers in $\frac{n}{2}$ to n are contained in group 2. Where n is the total number of client. A packet coming from another router first enters the *scheduler* of the router in which it has come.

The *scheduler* in the port determines whether the packet is destined for a client contained within the group as the router is in, if so, the packet is forwarded to *opcal* via *Inport*.

Opcal is the place where routing decision is taken in the port and the packet is routed according to the original routing algorithm of MIN Fat-Tree described in [9], where the left and right range of the router is determined by I_L = [2c_L,2c_L+2^r-1] and I_R =[2c_L+2^r,2c_L+2^{r+1}-1] respectively, c_L represents the address of the leftmost client attached in the group.

If, the packet injected from the source to the router is not destined to any client in the group formed by equation (1), then the scheduler forwards the packet to the higher levels of the topology. The packet eventually reaches the topmost group from where the downward path of the packet starts.

Note here that, the range calculation and routing decision is taking place only in the routers which belong to the same group as the destination client otherwise the packet is forwarded upward until the topmost routers are reached, which as mentioned earlier serves as group identifiers.

The *Inport* module in the router serves to distinguish, packets from *scheduler* and *opcal*. A packet received from *scheduler* by *Inport* signifies that the routing decision for the packet has to be made and the packet is forwarded to the *opcal*, while a packet from an *opcal* received by the *Inport* signifies that the routing decision has already been taken and the packet is forwarded to the *scheduler*. The figure below shows a logical client and router interface.

The results thus collected from the simulation of above networks are presented in section 5 along with the results of the proposed topology for comparative analysis. The next section describes the proposed reduced switch scalable min Fat-Tree topology.

4 Proposed Work: The Reduced Switch Scalable MIN Fat-Tree Topology

In this section we first explain the structure of the individual modules of the switch scalable MIN Fat-Tree. Thereafter a detailed explanation of the improved topology along with routing has been formulated.

4.1 Modules of the Proposed Network

The primary modules for a network are the routers, the clients and the interconnection links. The routers for the improved Fat-Tree network consists of hierarchical ports which are connected to each other. The interconnections among the ports have been so designed that two types of communications are possible, port-port and router-router. The port-port connection enables communication of message flits within the router. This would enable the flits to travel in the distinct links after the routing decision has been made by the router.

Due to hierarchical connections, alternative paths can be chosen whenever, the desired link is busy. Once the port-port communication is over, the flits travels out of the respective router towards the destination client or to other routers in the path. The ports have schedulers incorporated within them for taking the routing decisions. Whenever, the header flit of the message arrives at the port for the first time, the scheduler computes the routing and redirects towards to the respective port towards destination. The entire routing has been explained in the subsequent section.

Similar to the original Fat-Tree, our reduced switch scalable MIN Fat-Tree has all the clients connected to the lowest routers. Each router at the last level has two clients connected to them. Each of the clients has a source and a sink to generate and receive messages. An unique identification number has been assigned to each clients which has been used as destination address for communication.

4.2 The Reduced Switch Scalable MIN Fat-Tree Topology

The topology of new Reduced Switch Scalable MIN Fat-Tree is proposed here which not only reduces the hardware switches but also is implicitly scalable to number of client N, where N $\geq 2^3$. The objective of reducing the number of routers used to realizing the network is achieved by adding an extra link from each router of the network. The topological behavior of MIN Fat-Tree remain unchanged, but the number of routers needed to connect N number of clients is reduced by a factor of N$/2$ where N represents the total number of clients. The figure below describes how a Reduced Switch MIN Fat-Tree is fashioned out from a regular MIN Fat-Tree topology by adding the extra lateral links.

The reduced switch scalable MIN Fat-Tree will have direct links between the routers according to the height of the tree with at least 8 clients. Each router will have only one direct link attached to consecutive router. A tree with m clients can be represented as m= 2^h. Here, the height of the tree will be H = h-1. If the H=2 (the minimum height) then direct links will be between routers R_i and R_i+2 at both levels. If H>2 & H$\leq$4, then direct links at level higher than 2 will be between routers R_i and R_i+4. Similarly, if the H $>$ 4 & H$\leq$6 then direct links at level higher than 2 will be between routers R_i and R_i+8 and so on as it can be generalized using equations (2) and (3) as depicted below. Hence, with every increase in the height of the tree, the distance between the routers having direct links increases by a factor of 2:

$$Connection(r,c)_{Lateral} = (r, c \pm 2^l), \qquad (2)$$

$$\text{w}here, l = \begin{cases} 1, & \text{for lowest level of routers,} \\ r-1, & \text{o}therwise. \end{cases} \qquad (3)$$

Here, (r,c) is the level and column index of each router. The factor of 2 is added or subtracted to the column index shown in equation (2) whenever an odd number or even number of 2^l block of router is encountered in each level of router. Figure below depicts how the blocks are formed in different stages of the network.

4.3 Proposed Routing

The proposed routing for reduced switch scalable MIN Fat-Tree is based on Wormhole Routing [19]. The message packet has been divided into a number of flits (flow control digits) for transmission. Generally, the bits constituting a flit are transmitted in parallel between any two routers. The header flit (or flits) of a packet directs the route. As the header flit advances in a specified route, the remaining flits follow in a pipeline fashion. If the header flit encountered a channel busy, it is blocked until the

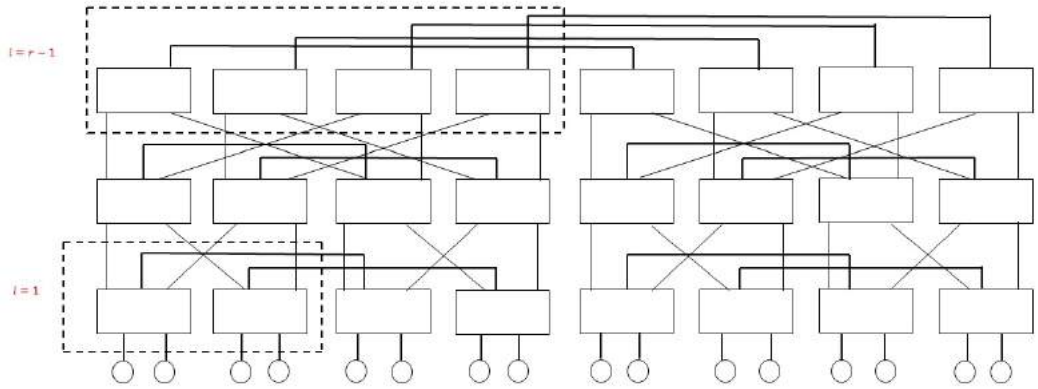


Fig. 7. Figure depicts how the blocks of 2^l routers are counted. At the level 1 and 2 at $l = 1$ and at higher levels $l = r - 1$

channel becomes available. These flits remain in flit buffers along the established route. Once a channel has been acquired by a packet, it is reserved for the packet. The channel is released when the last, or tail flit has been transmitted on the channel.

For ease of anticipation, prior to routing algorithm, we first present an example to demonstrate how the message packet traverses in the improved Fat-Tree. Let us consider an 8 clients network as shown in Fig.5.

Let the routers be numbered from 1-4 in the lowest level and 5-8 in the highest level of the network. The new links introduced are between routers 1 & 3, 2 & 4, 5 & 7 and 6 & 8. Hence, if a message packet needs to travel from a client in router 1 to destination in router 3, it has two links of the conventional Fat-Tree to travel to upper level and a new direct link between routers 1 & 3. Since, the network has been evenly partitioned into two equal partitions (routers 1, 2, 5 & 6 in first part and so on) it turns quite simple to take a routing decision. Whenever the header flit arrives router 1, the routing first fetches the destination address and checks whether it belongs to the same partition or different one.

If the partition is same, it could use the conventional Fat-Tree routing but if the destination is in different partition then first it will forward the packet with those new links of the improved Fat-Tree so that they directly reach the router to which the destination clients are connected. Only if these links are busy then they would follow the path to upper level and take similar routing decisions in those upper level routers.

The topology of the proposed network is free from deadlocks but, is prone to live locks. Here, the routing algorithm is designed so that it avoids livelocks by restricting the movement of flits in the upper part of the network once it has traversed the lateral links i.e. once a packet has traversed the lateral link, it can only be routed in downward direction.

Before presenting the proposed routing technique, we first present some preliminaries for easy anticipation. As each router at the lowest level have two clients connected to them, hence from if the total number of clients are known, the number of columns in the network can be computed by:

$$col = \frac{total\ clients}{2}. \tag{4}$$

The network defines the location of each routing switch with its position in the row and column, which can be fetched through a simple operation. However, the client addressing has been done by just assigning a positive integer n, where n $\in \{1, 2, 3.....\}$ in such a way that all the odd number will represent the left client and vice versa. At any instance to fetch the details of the client a function *finddestcol( )* is called to within routing such that:

$$finddestcol(\ ) = \lceil \frac{destination\ address}{2} \rceil. \tag{5}$$

The routing is primarily focused on three situations, (a) if destination is at the two ends of the network (b) if destination is in the centre of the network and (c) anywhere other than (a) and (b). If the destination client is at either end of the network, then the routing first targets to identify the direct links between the routers, if any exists. If they do, then the messages are directly forwarded by them. Else, they are sent a level up and again direct links are being searched and forwarded if found.

If only no such links exists or these links are busy then the messages are forwarded by conventional Fat-Tree routing till they reach the destination column. Once the packet is in the destination column, the destination is reached by forwarding the packet downwards by the rightmost links of each successive router in the downward path to the client.

Another important aspect of the routing is the level at which the messages reach. As the message travels to the next higher level, the direct links traverse the message to a router by a factor 2^n. Hence, throughout the routing each router first fetches the destination column where it should reach, then checks whether they are the end columns or centre columns or anywhere other than these columns. For each of these situations the routing decision is taken accordingly. We have explained the entire routing in two different algorithms. Algorithm 1 checks if there is a direct link between any two routers and also shows all the possibilities when such links can exists. Algorithm 2 routes the message in the network using Algorithm 1. In algorithm 1, we have restricted the number of rows to 6. This may be increase accordingly if required.

Algorithm 1: Check for direct links

Input: router location, total clients
Output: directlinks

initialization;
rcol ← get column of current router;
rrow ← get row of current router;
maxcol ← $\frac{total\ clients}{2}$;
if *(rrow == 1 || rrow == 2)* **then**
  if *(rcol* $\geq$ *1 && rcol* $\leq \frac{maxcol}{2}$ *)* **then**
    set l = 1;
    if *(count(number of* 2^l *blocks)%2 !=0)* **then**
      rcol+2^l $\rightarrow$ direct Link;
    end
    if *(count(number of* 2^l *blocks)%2 ==0)* **then**
      rcol-2^l $\rightarrow$ direct Link;
    end
  if *(rcol* $> \frac{maxcol}{2} + 1$ *&& rrow* $\leq$ *maxcol)* **then**
    if *(count(number of* 2^l *blocks)%2 !=0)* **then**
      rcol+2^l $\rightarrow$ direct Link;
    end
    if *(count(number of* 2^l *blocks)%2 ==0)* **then**
      rcol-2^l $\rightarrow$ direct Link;
    end
if *(rrow >2)* **then**
  if *(rcol* $\geq$ *1 && rcol* $\leq \frac{maxcol}{2}$ *)* **then**
    set l = r-1;
    if *(count(number of* 2^l *blocks)%2 !=0)* **then**
      rcol+2^l $\rightarrow$ direct Link;
    end
    if *(count(number of* 2^l *blocks)%2 ==0)* **then**
      rcol-2^l $\rightarrow$ direct Link;
    end
  if *(rcol* $> \frac{maxcol}{2} + 1$ *&& rcol* $\leq$ *maxcol)* **then**
    if *(count(number of* 2^l *blocks)%2 !=0)* **then**
      rcol+2^l $\rightarrow$ direct Link;
    end
    if *(count(number of* 2^l *blocks)%2 ==0)* **then**
      rcol-2^l $\rightarrow$ direct Link;
    end

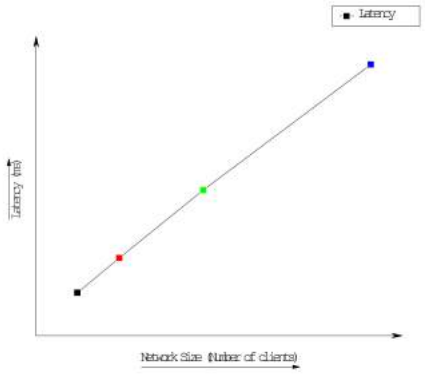
5 Results and Discussion

The simulation of the topologies mentioned in previous sections are done by using a data packet of 4 bytes for all the topologies and a generation function of injecting one packet per cycle is utilized. The time period of a cycle is fixed for 2ns.

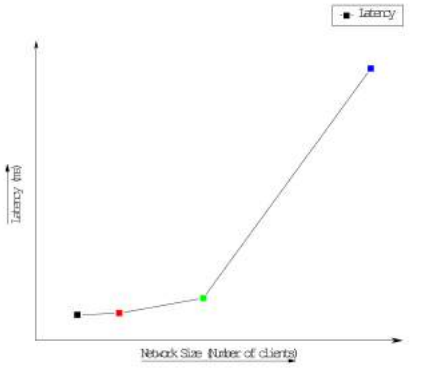
The OMNeT++ environment is installed in Windows 10 operating system. Two destination address generation schemes are utilized, one is a uniform address generation scheme which uses the intuniform() library function present in OMNeT++ header file.

This function generates a uniform destination address for the entire list of client present in the system.

For another type of destination address generation, a static address is chosen, which always forwards a packet from all the clients to a particular client, there by modeling the worst case performance of the topologies.

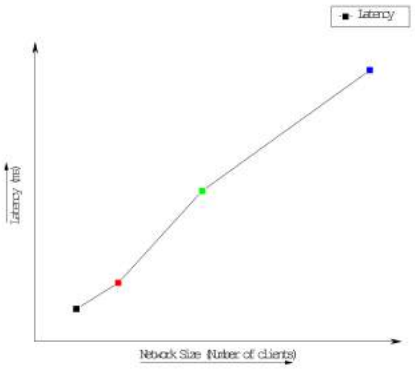


(a) Latency plot for Partial Group Based routing in MIN Fat-Tree for Uniform Adress

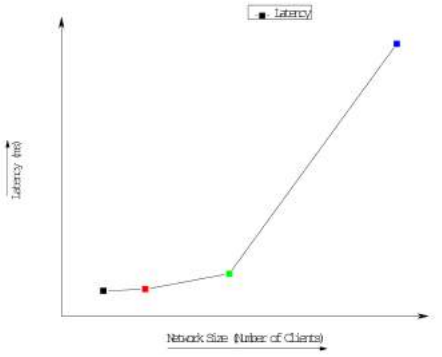


(b) Latency plot for Partial Group Based routing in MIN Fat-Tree for Static Adress

Fig. 8. Latency plot for Partial group based MIN Fat-Tree



(a) Latency plot for MIN Fat-Tree for Uniform Address



(b) Latency plot for MIN Fat-Tree for Static Address

Fig. 9. Latency plot for MIN Fat-Tree

5.1 Message Generation

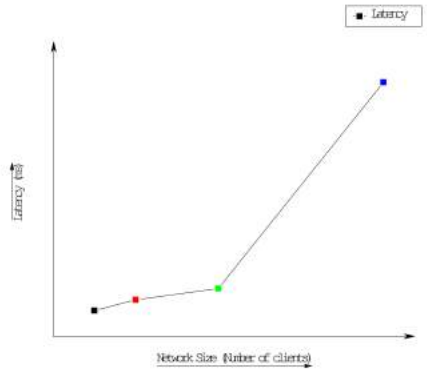
The message has been generated at the source of the client module in the network. Every single message is broken up into flits in such a way that the header flit hops into router for routing decision and all other follows till destination is reached. Buffers are used in the form of ports. The destination address in the packet decides the amount of traffic in the network during one simulation. Two different kinds of destination address generation techniques have been taken into consideration. The function used is:

$$d = x + \text{intuniform(1,y)}. \quad (6)$$

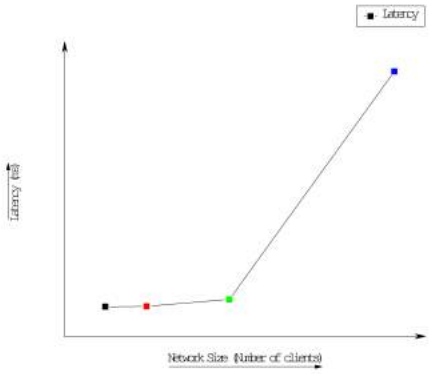
where x and y are have values such that x + y $<$ total number of clients in the network. The function generates random destination address by adding any value from the range 1 to y with x. This method implies that the clients were placed without any consideration of their communication patterns. The uniform generation is good as it will stress out the network as it will cause more packet hops.

5.2 Network Analysis

In order to analyze the proposed reduced switch MIN Fat-Tree network, two parameters have been considered. The latency measures the delay in delivery of a message generated from a source.

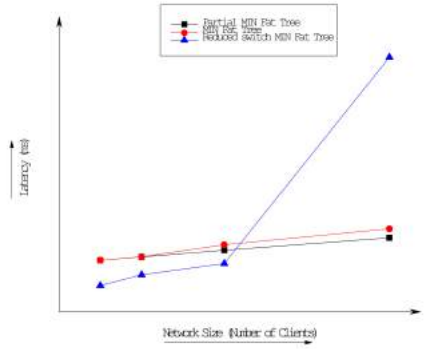


(a) Latency plot for Reduced Switch Scalable MIN Fat-Tree for Uniform Address

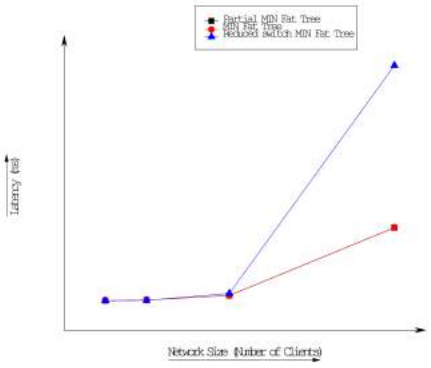


(b) Latency plot for Reduced Switch Scalable MIN Fat-Tree for Static Address

Fig. 10. Latency plot for Reduced Switch Scalable MIN Fat-Tree



(a) Latency comparison for all the three networks for Uniform Address



(b) Latency comparison for all the three networks for Static Address

Fig. 11. Latency plot for Partial group based MIN Fat-Tree

The hop count measures the number of hops the message had to make from source to destination. The network has been simulated for 8, 16, 32, 64 and 128 clients along with same number of clients for MIN Fat-Tree and with 64 clients of Partially Grouped MIN Fat-Tree. The results have been shown below.

5.2.1 Latency and Hop Count

The latency in our network has been computed as the time taken for the first flit to start from the source and till the last flit of the message reaches into the destination client. The Latency graph have been plotted for all the networks and shown in Fig. 7, 8 and 9, finally, the comparison graph is shown in the Fig. 10. The results have been shown as 10^{-3} seconds.

Another marker for analyzing simulated network is the hop count, which is, number of hops made by a flit to reach the destination from the source gives the hop count. Also table 1 enlists the maximum hop count for all the tree networks.

From the figures above it is evident that, partial group based routing in MIN Fat-Trees produces almost similar latency counts for smaller network

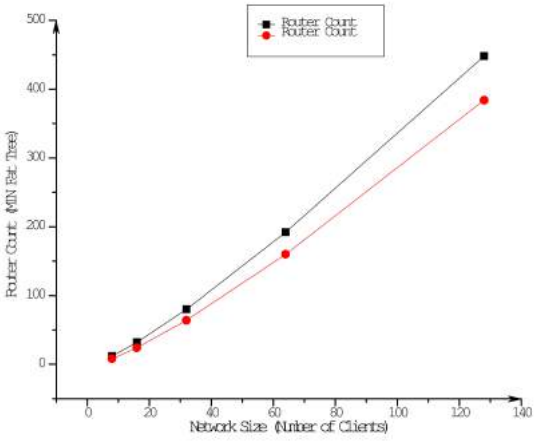


Fig. 12. Comparison of Number of Switches in MIN Fat-Tree and the Proposed Fat-Tree

sizes viz. 8 clients and 16 clients in uniform address scheme, but shows some improvement in higher network sizes viz. 32 and 64 clients as compared to traditional MIN Fat-Tree routing for similar network sizes.

For static address generation scheme partial group based routing and traditional MIN Fat-Tree routing produces almost similar graph.

On the other hand, the proposed reduced switch scalable MIN Fat-Tree topology shows relatively improved results for 8, 16 and 32 client size networks under uniform address generation scheme as compared to the other two networks, but shows high latency in bigger networks viz. 64 clients and above.

For static address generation scheme, the reduced switch scalable MIN Fat-Tree shows very high latency count for all network sizes as compared to other two networks.

It can be seen from the table above that, the proposed reduced switch scalable MIN Fat-Tree reduces the hop count by one hop accross all sizes of networks. This is because of the removal of a layer of switches/routers from the top level of the designed networks.

Algorithm 2: Routing

```
   Input: router column,destination column, router row
   Output: traverse up, down or directlink
1  initialization;
2  rowmax ← maximum height of the network;
3  destcol←finddestcol(Destination address);
4  rcol ← get column of current router;
5  rrow ← get row of current router;
6  while (! rowmax) do
7      if (directlink to destcol is available) then
8          if (directlink != busy) then
9              traverse directlink to destcol;
10         end
11         if (directlink == busy) then
12             traverse up by available link;
13         end
14     if (directlink unavailable) then
15         if (destcol is in reach_ range of rrow) then
16             traverse down left or right accordingly to
                 reach destination router in destcol;
17         end
18     if (directlink already traversed) then
19         traverse down left or right accordingly to reach
             destination router in destcol;
20     end
21     if destination router reached then
22         deliver packet to destination;
23         break;
24     end
25 end
26 if (rowmax) then
27     if (directlink traversed || destcol in reach_ Range)
         then traverse down left or right accordingly to
         reach destination router in destcol ;
28     else traverse directlink ;
```

5.2.2 Hardware Size

Another factor of analysis for the proposed reduced switch scalable MIN Fat-Tree was reduction in the number of switches/routers in the network.

For this purpose, we compared the number of routers in our proposed network with the conventional Fat-Tree used by Bouhraoua et al. [9-11].

It was found that our proposed network could reduce number of routers from 33.33% percent to 14.28% of the routers. For this analysis, we scaled the network size till 128 clients. In Table 3, we have showed in details the number of routers reduced with graphical analysis in Fig. 11.

Table 1. Comparison of the Hop Count for the three different networks

Network Size	Maximum HOP count		
	Reduce switch Scalable MIN Fat-Tree	Partially Grouped Routing For MIN Fat Tree	MIN Fat-Tree
8	4	5	5
16	6	7	7
32	8	9	9
64	10	11	11
128	12	13	13

Table 2. Comparison of the Hop Count for the three different networks

Network Size	Router Count		
	MIN Fat-Tree	Proposed MIN Fat Tree	% Reduced
8	12	8	33.33%
16	32	24	25%
32	80	64	20%
64	192	160	16.66%
128	448	384	14.28%

6 Conclusion and Future Work

In this paper we have proposed a reduced switch scalable MIN Fat-Tree for network on chip designs. The proposed network addresses the issue of reducing the size of original MIN Fat-Tree . The proposed network and its associated routing algorithm is simulated along with MIN Fat-Tree having Partial group based routing and the traditional routing. On careful analysis of the results thus obtained from the simulations, it was found that, the proposed network shows comparable results for network sizes of 8,16 and 32 clients in uniform addressing scheme.

However, the reduction in the switches made in the proposed network is compensated by a higher latency count in all the network sizes for static address generation scheme and for higher network sizes viz. 64 and above clients for uniform address generation scheme. Proposed network in inherently scalable and also reduces the hardware utilized by 33.33% to 14.28%. Although, further reduction in the said network could be achieved, but that is at the cost of scalability of the network. Further, the routing algorithm for the proposed network is free from livelocks and deadlocks.

The present paper attempted to reduce the height of the MIN Fat-Tree by eliminating the topmost row switches, It can be further investigated to see if the width of the network can also be reduced maintaining the scalable property of the network.

References

1. **Agarwal, A., Shankar, R. (2009).** Survey of network on chip (noc) architectures & contributions. Journal of Engineering, Computing and Architecture, Vol. 3.
2. **Anjum, S., Khan, I., Anwar, W., Munir, E., Nazir, B. (2012).** A scalable and minimized butterfly fat tree (SMBFT) switching network for on-chip communication. Research Journal of Applied Sciences, Engineering and Technology, Vol. 4, pp. 1997–2002.
3. **Bartic, T., Mignolet, J.-Y., Nollet, V., Marescaux, T., Verkest, D., Vernalde, S., Lauwereins, R. (2005).** Topology adaptive network-on-chip design and implementation. Computers and Digital Techniques, Vol. 152, pp. 467 – 472.
4. **Benini, L., De Micheli, G. (2002).** Networks on chips: a new SoC paradigm. Computer, Vol. 35, No. 1, pp. 70–78.

5. **Biswas, A., Hussain, M. A. (2016).** A fully grouped routing for homogeneous FAT tree network-on-chip topology. 2016 International Conference on Computing, Communication and Automation (ICCCA), pp. 1510–1514.
6. **Biswas, A., Hussain, M. A. (2019).** A survey of FAT-tree network-on-chip topology. International Journal Of Scientific & Technology Research, Vol. 08, No. 11, pp. 984–990.
7. **Biswas, A., Mahanta, H. J., Hussain, M. A. (2014).** Implementing a partial group based routing for homogeneous fat tree network on chip architecture. 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, pp. 364–370.
8. **Bjerregaard, T., Mahadevan, S. (2006).** A survey of research and practices of network-on-chip. ACM Comput. Surv., Vol. 38, No. 1, pp. 1–es.
9. **Bouhraoua, A., Elrabaa, E. (2006).** A high-throughput network-on-chip architecture for systems-on-chip interconnect. pp. 1–4.
10. **Bouhraoua, A., Elrabaa, M. (2007).** An efficient network-on-chip architecture based on the fat-tree (FT) topology. volume 32, pp. 28–31.
11. **Bouhraoua, A., Elrabaa, M. (2010).** A new client interface architecture for the modified fat tree (MFT) network-on-chip (NOC) topology. Proceedings of the 5th International Workshop on Reconfigurable Communication-Centric Systems on Chip 2010, pp. 169–172.
12. **Bouhraoua, A., Elrabaa, M. E. (2006).** An efficient network-on-chip architecture based on the fat-tree (FT) topology. 2006 International Conference on Microelectronics, pp. 28–31.
13. **Bouhraoua, A., Elrabaa, M. E. S. (2008).** Addressing heterogeneous bandwidth requirements in modified fat-tree networks-on-chips. 4th IEEE International Symposium on Electronic Design, Test and Applications (delta 2008), pp. 486–490.
14. **BOUHRAOUA, A., ELRABAA, M. E. S. (2011).** Improved modified fat-tree topology network-on-chip. Journal of Circuits, Systems and Computers, Vol. 20, No. 04, pp. 757–780.
15. **Dally, W. J., Towles, B. (2001).** Route packets, not wires: on-chip interconnection networks. Proceedings of the 38th Design Automation Conference (IEEE Cat. No.01CH37232), pp. 684–689.
16. **Guerrier, P., Greiner, A. (2000).** A generic architecture for on-chip packet-switched interconnections. DATE'00.
17. **Kumar, S., Jantsch, A., Soininen, J., Forsell, M., Millberg, M., Oberg, J., Tiensyrja, K., Hemani, A. (2002).** A network on chip architecture and design methodology. Proceedings IEEE Computer Society Annual Symposium on VLSI. New Paradigms for VLSI Systems Design. ISVLSI 2002, pp. 117–124.
18. **Leiserson, C. E. (1985).** Fat-trees: Universal networks for hardware-efficient supercomputing. IEEE Transactions on Computers, Vol. C-34, No. 10, pp. 892–901.
19. **Ni, L. M., McKinley, P. K. (1993).** A survey of wormhole routing techniques in direct networks. Computer, Vol. 26, No. 2, pp. 62–76.
20. **Pasricha, S., Dutt, N. (2008).** Networks-on-chip. In **Pasricha, S., Dutt, N.**, editors, On-Chip Communication Architectures, Systems on Silicon. Morgan Kaufmann, Burlington, pp. 439–471.
21. **Peratikou, A. (2014).** An optimised and generalised node for fat tree classes. Ph.D. thesis, School of Computing University of Portsmouth.
22. **Petrini, F., Vanneschi, M. (1997).** k-ary n-trees: high performance networks for massively parallel architectures. Proceedings 11th International Parallel Processing Symposium, pp. 87–93.
23. **Requena, C. G., Villamón, F. G., Requena, M. E. G., Rodríguez, P. J. L., Marín, J. D. (2008).** RUFT: Simplifying the fat-tree topology. 2008 14th IEEE International Conference on Parallel and Distributed Systems, pp. 153–160.
24. **Sheibanyrad, A., Greiner, A. (2007).** Hybrid-timing FIFOs to use on networks-on-chip in GALS architectures. ESA International Conference on Embedded Systems and Applications, CSREA Press, Las Vegas, Nevada, United States, pp. 27–33.
25. **Valerio, M., Moser, L. E., Melliar-Smith, P. M. (1994).** Recursively scalable fat-trees as interconnection networks. Proceeding of 13th IEEE Annual International Phoenix Conference on Computers and Communications.

Article received on 21/02/2021; accepted on 28/12/2021.
Corresponding author is Abhijit Biswas.

ISSN 2007-9737

Ride Sharing Using Dynamic Rebalancing with PSO Clustring: A Case Study of NYC

Moustafa Maaskri[1], Mohamed Hamou Reda[2], Adil Tomouh[1]

[1] Djillali Liabes University,
Computer Science Department, EEDIS Laboratory,
Algeria

[2] Dr. Tahar Moulay University,
Computer Science Department, GeCoDe Laboratory,
Algeria

moustafa.maaskri@univ-sba.dz, hamoureda@yahoo.fr, toumouh@gmail.com

Abstract. The shared vehicle can improve the efficiency of urban mobility by reducing car ownership and parking demand. Existing rebalancing research divides the system coverage area into defined geographical zones, but this is achieved statically at system design time, limiting the system's adaptability to evolve. In the current study, a method has been proposed for rebalancing unoccupied vehicles in real-time while considering travel requests, using a bio-inspired method known as Particle Swarm Optimization clustering (PSO-Clustering). The solution was examined using data on taxi usage in New York City, first looking at the traditional system (no ride sharing, no rebalancing), then carpooling, and finally of both ride sharing and rebalancing.

Keywords. Ride sharing, PSO, rebalancer, clustering, simulation.

1 Introduction

Traffic congestion on city roads has become a significant issue that must be addressed in urban development due to urbanization and the rapid rise in vehicles The acceleration of urbanization and the rapid increase within the number of vehicles for travel have made traffic congestion on urban roads a serious problem that must be addressed in urban development. As a result, several researchers suggested the concept of "carpooling". Experimental results have shown that this idea demonstrates the effectiveness of policies in reducing urban traffic congestion.

In March 2013, researchers at the Massachusetts Institute of Technology (MIT) analyzed a week of taxis in Manhattan, New York [1]. Approximately 10,000 of New York's 13,600 taxis were used during the hour. To meet its 98% transportation requirement, Manhattan only needs 3,000 shared taxis. This study found that an effective carpooling system reduces traffic congestion in cities and improves the speed of passenger transportation for in-service vehicles and drivers' operating benefit.

Furthermore, energy consumption and environmental pollution should be reduced [9]. Accordingly, implementing carpooling is an effective way to increase the quality of urban traffic [10, 12]. Some cities have introduced and incorporated taxi ride sharing as a means of reducing taxi traffic congestion. As a result, carpooling has sparked the attention of many researchers as an intriguing subject of urban transportation science. In 2011, Agatz investigated the issue of driver and passenger assignment in a competitive environment and suggested a method for maximizing vehicle mileage and individual travel costs [2].

Shinde presented a multi-objective optimization-assisted carpool path matching genetic algorithm.

The algorithm reduces computational complexity and time intervals while also improving the carpool effect [19]. In 2015, Pelzer proposed a dynamic decision algorithm that supported the partition of the network [19]. The algorithm divides and numbers the road network and uses the spatial route search algorithm for matching passengers and vehicles.

In 2015, Jiau used a genetic algorithm to implement carpool path matching in a short amount of time, resulting in a carpool path matching scheme with low complexity and memory [13].

In 2015, Huang suggested a fuzzy control genetics-based carpooling algorithm that combines a genetic algorithm and a fuzzy control system to optimize the route and balance driver assignments and demands in an intelligent carpooling system [11].

Cheng developed a multi-dynamic taxi ride sharing model in 2013, using the genetic algorithm to solve the carpool problem to benefit travelers and drivers [6].

In 2013, Ma proposed a large-scale carpooling service; it responds efficiently to real-time requests sent by taxi users and generates carpooling schedules that significantly reduce the total distance of the trip [15].

Xiao et al. developed a membership function based on three factors: driving directions, driving time, and the number of passengers in 2014 to achieve fuzzy carpool grouping and identification of passengers and taxis [23].

In 2017, Zhang proposed the first systematic work, named CallCab, based on a data-driven methodology to design a single recommendation system for daily and ride-sharing services. This recommendation system was designed to help passengers find the best taxi with carpooling [25].

As shown in Figure 1, which was generated by extracting requests from New York Taxi data over two separate periods, mobility requests develop over time, and the distribution of requests is uneven [20].

It can result in an unbalanced distribution of drivers for RS systems, as seen in **??**, where the majority of demand is concentrated in the upper region, and most vehicles are located on the opposite side after their last journey, where fewer new customers request a ride [4].

2 Related Work

Taxi rebalancing can be categorized into approaches based on static zones [8, 7, 3, 22, 21] and dynamic zones [5, 14]. In the generation of static rebalance zones, the relocation zones' geographic coverage is predefined at design time. For example, the New York Manhattan area is divided into predefined areas that do not change over time [8]. Each vehicle, using cross-learning, learns and decides at each time step whether to move to one of the neighboring areas or to stay in its current area. Austin's rebalancing zones are established by dividing the city into 2 square mile square blocks in [7].

For each location, a block weight is calculated to account for the excess or deficit of vehicles in the block in the sense of expected travel supply and demand. The forecast travel demand is determined from historical data and current demand, and blocks with a low weight aim to collect vehicles from the community where there is a surplus. The zones were identified using a fine-grained grid but also static [3]. If a vehicle is idling, it will rebalance itself using its local knowledge: it determines whether or not to rebalance towards a neighboring region based on the distribution of demand in the surrounding areas [8]. According to the route of the road network, the works presented in [7] divide the region into rebalancing areas.

The zones are defined so that for each region Ri, a zone makes it possible to reach Ri in the time allotted. Idling vehicles are rebalanced to prevent excess vehicles in the same area, taking into account travel time to reduce empty journeys and potential demand, calculated from the current order. However, the zones do not adjust based on traffic conditions or the number of taxis once they have been established. The majority of approaches only allow rebalancing for idling cars. In contrast, rebalancing was combined with carpool assignment, allowing carpool pick-up from zones neighbors, reducing passenger waiting time but increasing travel time [3, 8].

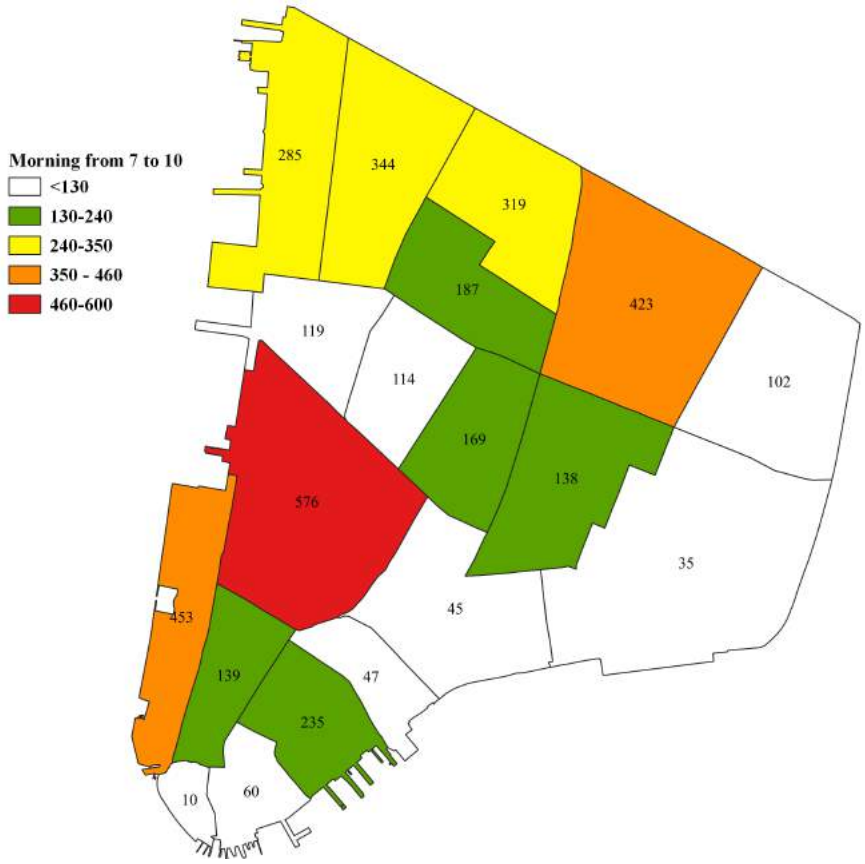


Fig. 1. Observed demand imbalance in NY Taxi dataset [20] trips between morning (7-10 am) and evening (6-9 pm) in the south part of Manhattan on Tuesday, July 7, 2016 (morning)

A dynamic zone generation algorithm was used for balancing; rebalancing zones were calculated using a clustering algorithm. On the queries produced by a distribution specified on historical data, k-means clustering is applied [5].

Consequently, the coverage and size of the zones can change, but the total number of zones remains constant. the second approach uses EM clustering and allows to create a different number of zones according to different densities of requests[14]. Furthermore, the approach uses real-time data rather than historical data to better respond to dynamic demand.

Both approaches are similar to the current, in which a particle swarm optimization (PSO) clustering algorithm was used to generate dynamic zones for balancing unoccupied vehicles, with a fixed number of zones based on real-time data.

3 Background

This section presents the basic information needed to understand the design and implementation of our approach: the particle swarm optimization (PSO) algorithm and the latter's clustering strategy for car rebalancing.

3.1 Particle Swarm Optimization Algorithm (PSO)

The living world initially inspires this algorithm. It is based on a model created by Craig Reynolds at the end of the 1980s to simulate the flight of a flock

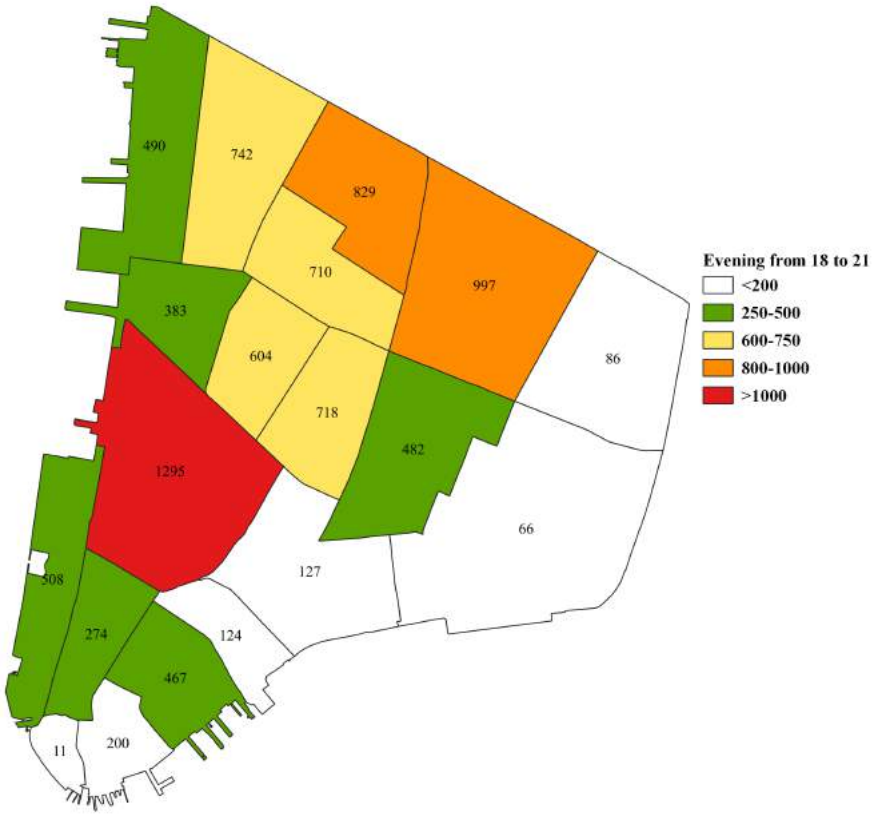


Fig. 2. Observed demand imbalance in NY Taxi dataset [20] trips between morning (7-10 am) and evening (6-9 pm) in the south part of Manhattan on Tuesday, July 7, 2016 (evening)

of birds. The mathematical description of PSO [18, 17, 24] is as follows:

We assume that the size of the population is N, each particle is treated as a point in D dimensional space. The i^th particle is represented by $x_i = (x_{i1}, x_{i2}, ..., x_{id}, ..., x_{iD}), x_i$ is a latent solution of the optimized question. The rate of the particle i is represented as v_i, $v_i = (v_{i1}, v_{i2}, ..., v_{id}, ..., v_{iD})$, it is a position change quantity of particle in an iteration. The particles are manipulated according to the following equation:

$$v_{id} = \omega v_{id} + c_1 rand()_1 (p_{id} - x_{id}) + c_2 rand()_2 (p_{gd} - x_{id}), \quad (1)$$

$$\begin{cases} v_{id} = v_{max} & if v_{id} > v_{max}, \\ v_{id} = -v_{max} & if v_{id} < -v_{max}, \end{cases} \quad (2)$$

$$x_{id} = v_{id}. \quad (3)$$

In the equation (1), the historical best position of all the particles in the population is represented by pgd, the historical best position of the current particle is represented by p_{id}, the particle's new velocity is calculated according to its previous velocity and the distances of its current position from its own historical best position and the group's historical best position.

Variable ω is the Inertia weight,c_1 and c_2 are positive constants, $rand_1()$ and $rand_2()$ functions in the range [0,1]. In equation (2), particles' velocities in each dimension are limited to a maximum velocity v_{max}, with v_{max} determining the search precision of particles in solution space. If it's too large, the particles will fly the best solution; if it's too small, the particles will fall into the local search space and have no way of moving to the global search.

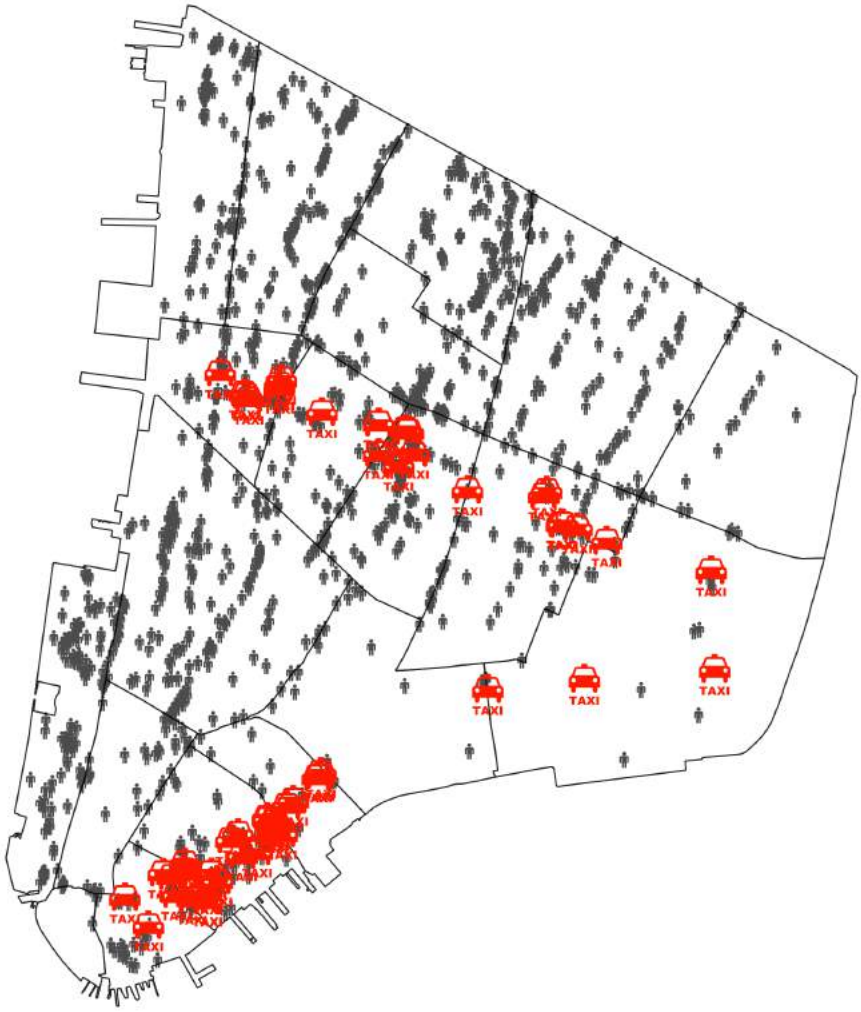


Fig. 3. Example of an unbalanced fleet distribution in 2 min

The particle's new position is determined using its current position and new velocity in equation (3), and the performance of each particle is then evaluated using a predefined fitness function, leading to the best solution to the research issue.

3.2 The Clustering Algorithm based on PSO

In the clustering algorithm based on the Particle Swarm Optimization algorithm[16], each particle $Y_i = (y_1, y_2, ..., y_K)$ represents centers of the K classes, while $y_j(j = 1, 2, ..., K)$ represents the central point's coordinates vector of the j_{th} class in the i^{th} particle (the dimension of y_j is decided according to the actual situation).

The particle swarm constitutes many candidate classified plans. We know it is a key of clustering which use an optimization algorithm to evaluate the quality of classification plan, so the authors propose an adaptability function f as follows:

$$f(y_i) = \frac{max(\bar{d_1}(y_i))}{max(d_2(y_i))}. \quad (4)$$

where $max(\bar{d}_1(y_i))$ is the maximum value of mean values of distances within the same classes in the classification plan, which is expressed by particle Y_i, while $max(d_2(y_i))$ is the minimum value of distances between classes in the classification plan, which is expressed by particle Y_i: $max(\bar{d}_1(y_i)) = \max_{j=1,2,...,k} \left(\sum_{\forall x_i \in y_i} \frac{d(x_i, y_i)}{|y_j|} \right)$, $|y_j|$ is the element number in the j^th class. $min(d_2(y_i)) = \min_{\forall i,j,i \neq j}(d(y_i, y_j))$; $i, j = 1, 2, ..., k$.

If the minimum value of the adaptability function (4) simultaneously satisfies a small distance within the same class and a considerable distance within classes, the classification strategy is stronger.

The clustering algorithm based on the Particle Swarm Optimization algorithm consists of the following steps:

1. In the n dimension space, we set the population size m, acceleration coefficient c_1 and c_2, hypothesis biggest iterative times num, clustering number K, and a given point set with N points, etc. Set the historical best position of each particle p_{best} equal to the initial position and set the global best position of particle swarm p_{best} equal to the best of all p_best in a population of particles with random positions and velocities (the position and velocity vectors are constituted by K vectors of n dimension space).

2. For each particle Y_i, recalculate distances between the set $\{x_1, x_2, ..., x_N\}$ and K centers and divide the set $\{x_1, x_2, ..., x_N\}$ according to the distance regulation of the K-means algorithm.

3. For each particle Y_i, Calculate the fitness evaluation according to the expression $f(Y_i)$.

4. Compare and reset the historical best position p_{best} and the best fitness evaluation of each particle, as well as, compare and reset the global best position g_{best} and the best fitness evaluation of particle swarm.

5. Change the velocity and position of particles according to equations (1) and (3) and limit them according to equations (2) and (5):

$$\begin{cases} x_{id} = x_{max} & if\ x_{id} > x_{max}, \\ x_{id} = -x_{max} & if\ x_{id} < -x_{max}. \end{cases} \quad (5)$$

   In the expression (5), we select the maximum value of each dimension in all points as x_{max}.

6. Inspect termination condition (the algorithm has achieved the hypothesis biggest iterative times); if it is met, the algorithm should be terminated; otherwise, return to step (2).

7. Output classification result.

4 Dynamic Rebalancing based on Demand

This section describes the generation of zones based on requests for real-time vehicle rebalancing (DRBD) in RS fleets. We introduce an RS system first and then our proposed rebalancing.

4.1 Ride Sharing System

We have designed a ride sharing algorithm applied to a fleet of 4-seater vehicles for the carpooling system. This model is designed to work in any city around the world. Each vehicle perceives the order from the dispatcher for pick up or drop off passengers or rebalance. This cycle is described in the algorithm 1.

The internal state of the vehicle is composed of its position, represented by the latitude and longitude coordinates, its destination, and the number of seats empty. For an empty vehicle, the destination is zero, and if a vehicle responds to one or more requests, its destination corresponds to that of the request R_i, which can be served fastest. The vehicle's location on the road network is updated whenever a new position is reached, whether it is the destination or the pick-up point.

A request R is available for a vehicle V if there are enough empty seats to accommodate the number of passengers associated with the request (ranging from 1 to 4), and the total waiting time for

R, (i.e. the time between the creation of the request and the estimated time of passenger pick-up is less than the maximum time allowed; considered to be 15 minutes). All customers who have waited more than 15 minutes leave the system unserved, and the request is recorded as unserved. $S_r^i = [r_{pos}^i, r_{dest}^i, r_{passengers}^i]$ represents the state of the i^{th} request received by a vehicle. Each request includes a pick-up location, destination, and the number of passengers.

Algorithm 1 Controller Vehicle V

Parameters: V Vehicle, R Request
Result: Given To The V The Best Action

1: **function** DO_ACTION(V)
2: **if** V is idle **then** ▷ //V is out of R and no R to pick up
3: Rebalance(V); ▷ //Algo 2
4: **else** ▷ //A is drive to destination
5: **if** existe R matching A and A has enough space **then**
6: $V.PickUp(R)$; ▷ //Doing ride sharing
7: **end if**
8: **end if**
9: **end function**

Vehicles can choose between 3 actions, which are organized into two categories: (1) drop-off, in which a vehicle responds to a request by driving the passenger (s) to their destination, (2) pickup, in which a vehicle goes to a pickup point of the selected request, and (3) when the vehicle perception is empty, and it does not respond to any demand, it is activated to rebalance as shown in line 3 of the algorithm 1.

4.2 Rebalancer - DRBD

Rebalancing can be used for various carpooling systems; however, the researchers use the Expectation-Maximization approach for the clustering in their Deep RL carpooling request attribution strategy [5]. We only used DRBD rebalancing in our case, which is activated when a vehicle receives no requests and no more requests to serve in its region. The DRBD aims to allocate vehicles efficiently and dynamically based on current demand, thus avoiding fleet imbalance, leading to longer waiting times for passengers or a high number of requests not processed. DRBD deduces the travel zones and calculates their associated probabilities for a vehicle to rebalance Eq. 6:

$$p_r(v, z_i) = \frac{|R_i|}{|R|}, \tag{6}$$

where z_i is the i^{th} zone, R_i is the set of pending requests within the current zone, and R is the set of pending requests across all zones.

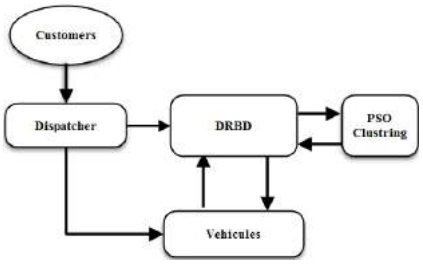


Fig. 4. Rebalancing whith DRBD

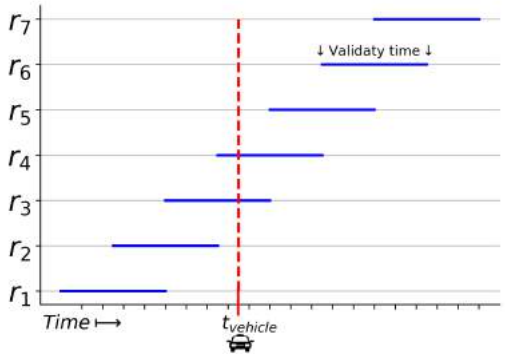


Fig. 5. Requests active at time t_v, only (r3; r4) are taken into account when rebalancing

The principle of the method BRBD is illustrated in Figure 4, representing the rebalancing module. The demand dispatcher has a dual role: it selects the demands for vehicles according to time and seat availability and filters the demand for rebalancing, depending on the period used.

DRBD takes as input the pending requests available at the moment. As shown in Figure 5, only the validated requests are considered, not any

previous fulfilled or rejected requests (estimated or planned).

The procedure applied to move an idling vehicle to a new position is described in Algorithm 2. First of all, DRBD generates new clusters based on pending (unserved) requests, according to Algorithm 3.

Algorithm 2 Rebalancing idle vehicles

Parameters: V_{idle} **Idle Vehicle**
Result: Rebalancing a vehicle V to a new position

procedure REBALANCE (V_{idle})
  $C, relocatingProb \leftarrow FindingClusters(reqsAvailable, K)$ ▷ Algo 3
for each: $(V \in V_{idle})$
  $rnd \leftarrow generate\_random\_value \in [0, 1]$
  $i \leftarrow 0$
  while $(i \leq size(C))$ **do**
    if $relocatingProb[i] \geq rnd$ **then**
      $V.distination \leftarrow C[i].position$
      $V.MoveToDistination$
      $Break$ ▷ exit while loop
    end if
    $i++$
  end while
end procedure

Algorithm 3 Definition relocating zones for rebalancing

Parameters: $reqsAvailable$ **validity Requests**
Result: $Clusters$**,** $relocatingProb$

function FINDING CLUSTERS ($reqsAvailable, K$) ▷ //K is the number of clusters
  $Clusters, C \leftarrow PSO\_Clustring(reqsAvailable, K)$ ▷ //C is the centroid
  $prob \leftarrow 0$
for each: $(i \in Clusters)$
  $prob = \frac{size(i)}{size(reqsAvailable)}$
  $relocatingProb[i] \leftarrow prob$
  return $C, relocatingProb$
end function

By applying optimization swarm particulates for clustering, a total of K clusters and their centroid are generated.

The vehicle is then moved to a featured area selected by a weighted random selection based on the probability of each category calculated in Equation 6 (lines 4-5. We preferred a weighted random selection approach over the others because it allows vehicles to explore different areas when rebalancing, preventing all vehicles from rebalancing in the same area.

5 Experimental Setup

The requests are generated using the Open New York Taxi Dataset [20]. It describes the recorded trips of yellow taxis in the Manhattan area. We extracted trips from three consecutive Tuesdays of July 2015 to represent typical weekday demand patterns.

We use a fixed fleet size of 200 shared vehicles to observe cases where the activation of carpooling is necessary to satisfy all requests (peak hours from 7:00 to 10:00). Each vehicle has a capacity of 4 passengers. Each request includes the time when the user requested the trip(ptime), the number of passengers(npass), the longitude and latitude of origin point (olng,olat), and the destination point (dlng, dlat) (See Table 1).

We used a simplified traffic simulation to focus only on rebalancing and carpooling strategies without taking congestion into account. Vehicles drive themselves to their current destination (e.g., driver pick-up or drop-off point or relocation area).

Travel time is calculated in the same way as in a grid network, and we assume a speed of 35 Km/h for peak hours. The evaluation is based on morning peak traffic data (7:00 to 10:00) and includes 10,000 requests. The number of Passengers by request is distributed during evening peak time (See Figure 6).

The demand satisfaction rate was less than 100%. Looking at Figure 6, it is apparent that few requests have 5 and 6 passengers necessitating the use of vehicles with 6 seats available together. All the vehicles used in the simulation had a maximum capacity of four passengers.

Table 1. Data set head

N	ptime	olng	olat	dlng	dlat	npass
1	$2015/07/07-07:00:01$	-74.015488	40.715603	-74.010475	40.721542	1
2	$2015/07/07-07:00:05$	-73.985352	40.722023	-73.999344	40.733822	1
3	$2015/07/07-07:00:11$	-73.996910	40.725388	-74.011169	40.709332	2

Table 2. All values refer to 10,000 requests served by a fleet composed by 200 vehicles of 4 seats

Scenarios	Requests served	Requests served (%)	WT (min)	DS(Km)
Base (no RS,no RB)	8388	83.88	5.14	124.22
Ride charing only	9010	90.10	4.03	135.26
DRBD (RS and RB)	9025	90.25	3.59	136.38

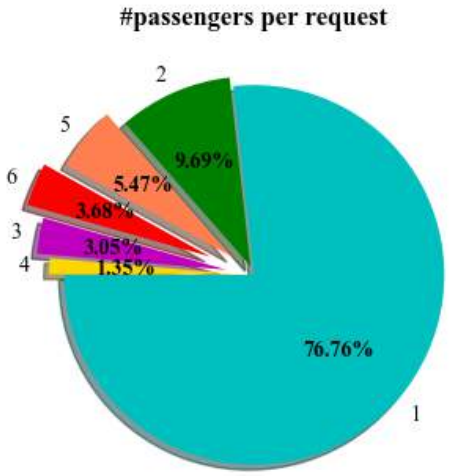


Fig. 6. Number of passengers per requests (the first 10,000 requests in dataset)

6 Results and Analysis

6.1 General Considerations

The DRBD rebalancing was compared with the following strategies to evaluate the performance of our approach:

— Base: a central dispatcher assigns the nearest vehicle to the request with the highest waiting time.

— Ride sharing: if the new request origin and destination are in zones on the current route, a vehicle can pick up more demands before it reaches full occupancy.

— Ride sharing and rebalancing: the vehicle drives towards the center of the zone to which it was randomly assigned.

For all the simulation scenarios, we have assumed that each vehicle has a capacity of 4 passengers, so requests with more than 4 passengers are ignored by the vehicles. Based on the data collection, the fleet will serve 90.85% of the 10,000 requests at its maximum level of service.

6.2 Evaluation Metrics

To evaluate DRBD, we use the set of the most commonly used indicators in related work:

— Number and percentage of served requests.

— Number and percentage of timed-out requests (RR):The maximum waiting time per request is limited to 15 minutes and after this the request is discarded from the system and flagged as unserved.

— Waiting time (WT): the time between the user request generation and the pick-up time.

— Total Vehicle distance traveled per vehicle in service (DS).

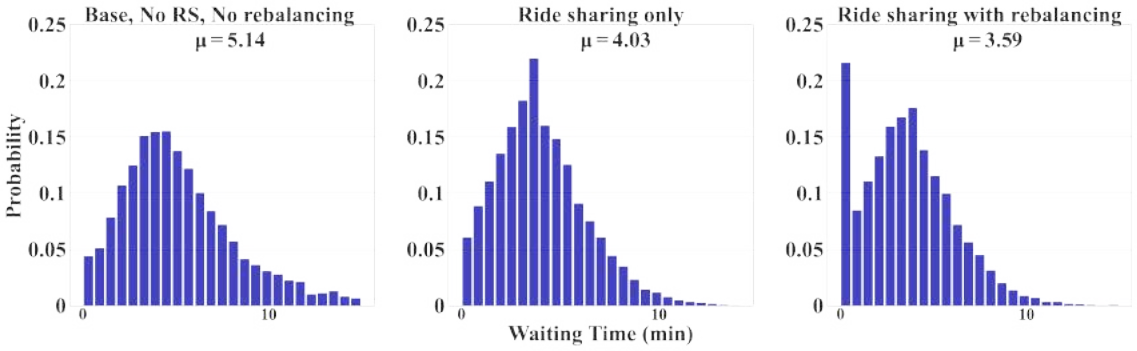


Fig. 7. Comparison of the implemented scenarios: waiting time

— Number of requests served per vehicle ($N_{requests}$): we recorded the distribution of the number of passengers per vehicle in the fleet and computed its variance.

6.3 Simulation Results

Each scenario shows a different level of service according to the indicators mentioned above.

— Served requests: Scenario Base (no RB, no RS), which models a standard taxi service, serves about 83.88% of requests, and ride sharing serves 90.10% of requests. However, the DRBD serves 90.25% of requests.

— Waiting time (WT): Passenger waiting times for each scenario were shown in Figure 7. We observed a significant reduction in waiting time by enabling ride sharing in the base scenario (No RS and No RB). RS only indicates waiting times (4.02 minutes); but, when we use PSO's DBRB clustering, we see another reduction in WT (3.59 min).

— Requests Distribution: As shown in Figure 8, several vehicles in the Base are traveling with just a few passengers in Base (no RB, no RS). Once serving one or few requests, these vehicles may end up in an area of the network that is empty of any further request. Enabling rebalancing or ride sharing can prevent them from staying idle and help the vehicle to find new requests. It can be seen in situations DRBD and ride sharing only, where allowing ride-sharing and rebalancing results in further improvements since all vehicles serve the same number of passengers.

— Distance traveled: Figure 9 shows the distance traveled by vehicle for each scenario. Base scenario (no RB, no RS) shows that around one-fourth of the vehicles are traveling only a few kilometers, confirming they only serve a few requests and then stay idle in an area with no further demand. Since the number of serving requests varies by scenario, an essential difference in distance traveled was recorded. From Table 2, we can confirm that enabling rebalancing adds additional travel distance in service for vehicles.

6.4 Discussion

According to simulation results, allowing flight sharing and rebalancing has shown very positive results in satisfying most possible service requests. We showed that DRBD with PSO Clustering improves all vehicles' average and individual performance when used in conjunction with ride sharing compared to Base (no RD, no RS).

Passenger wait time for DRBD has decreased to nearly 40% compared to classic taxi service (Scenario Base - no RB, no RS). However, the main observed advantage of DRBD (with RB and RS) is that each vehicle's workload seems to better

ISSN 2007-9737

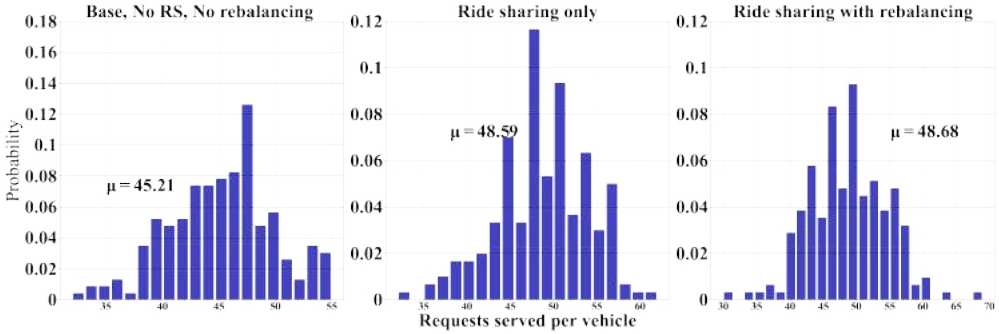


Fig. 8. Comparison of the implemented scenarios: Requests Distribution

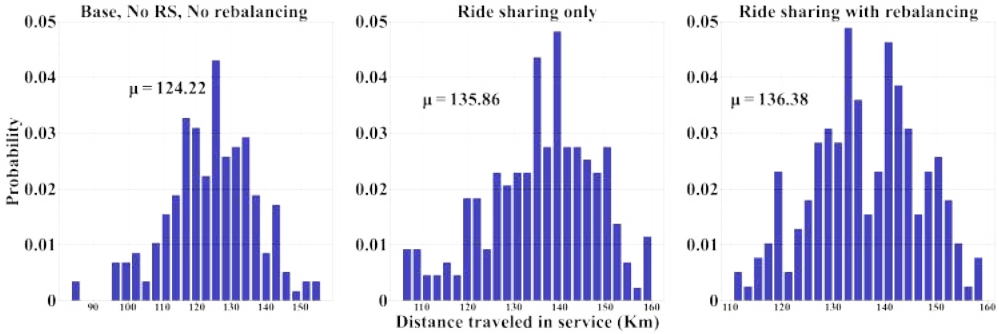


Fig. 9. Comparison of the implemented scenarios: Distance traveled

converge to a global average value, resulting in fairer workload distribution. It has been found that the rebalancing approach (RS and RB with DRBD scenarios) generates additional traveled distance, resulting in a slight increase in overall distance in operation.

We show a snapshot of the number, size, and shape of the clusters it created for idle vehicle V at time t to illustrate how DRBD by PSO clustering differs from fixed zone clustering in terms of zone outcomes (See Figure 10). DRBD was used to calculate ten relocation zones in this case. The stars represent cluster centers, and the number of the related trips.

7 Conclusion and Future Work

This paper presents a Dynamic Rebalancing Based on Demand (DRBD), a vehicle rebalancing algorithm for ride sharing in sharing vehicle systems. Unlike existing approaches which use fixed geographical zone to relocate empty vehicles, DRBD uses PSO clustering to generate zones dynamically. DRBD enables zones to be dynamic in terms of position by their centroid.

First, rebalancing areas are identified by analyzing pending requests in real-time at each time step. The zone to which an unoccupied vehicle rebalances is then determined using a probability distribution defined on the zones, which is calculated by dividing the number of requests in the zone by the total number of requests in all zones.

The effectiveness of the BRBD is simulated by integrating it with 200 carpool vehicles, which respond to 10,000 carpool requests in the lower Manhattan area. In terms of the DRBD technique, the workload is distributed more evenly throughout the fleet, suggesting a more accurate rebalancing strategy with no loss of efficiency while respecting waiting times and passenger distribution by vehicle.

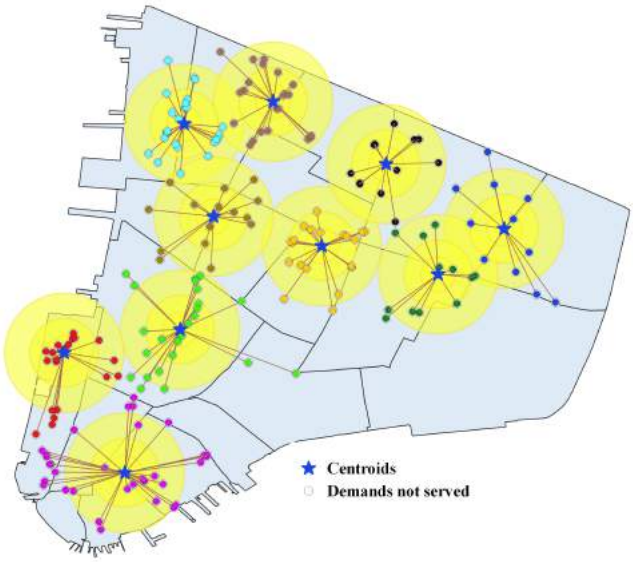


Fig. 10. Example of PSO clustering in instance t with k=10

This work can be extended in several directions. Check for general applicability should be combined with other ride sharing methods and tested using other maps and data sets of road networks. In terms of learning, the vehicles could be activated using a reinforcement learning model to fine-tune their behaviors in response to new demand models that emerge. Rebalancing could be further improved by considering real-time traffic congestion when deciding which cluster to move.

References

1. **Afian, A., Odoni, A., Rus, D. (2015).** Inferring unmet demand from taxi probe data. volume 10.
2. **Agatz, N. A., Erera, A. L., Savelsbergh, M. W., Wang, X. (2011).** Dynamic ride-sharing: A simulation study in metro atlanta. Transportation Research Part B: Methodological, Vol. 45.
3. **Alabbasi, A., Ghosh, A., Aggarwal, V. (2019).** Deeppool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning.
4. **Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., Rus, D. (2017).** On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proceedings of the National Academy of Sciences of the United States of America, Vol. 114.
5. **Castagna, A., Guériau, M., Vizzari, G., Dusparic, I. (2021).** Demand-responsive rebalancing zone generation for reinforcement learning-based on-demand mobility. AI Communications, Vol. 34.
6. **Cheng, J., Tang, Z., Liu, J., Zhong, L. (2013).** Research on dynamic taxipooling model based on genetic algorithm. Wuhan Ligong Daxue Xuebao (Jiaotong Kexue Yu Gongcheng Ban)/Journal of Wuhan University of Technology (Transportation Science and Engineering), Vol. 37.
7. **Fagnant, D. J., Kockelman, K. M. (2018).** Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in austin, texas. Transportation, Vol. 45.
8. **Gueriau, M., Dusparic, I. (2018).** Samod: Shared autonomous mobility-on-demand using

decentralized reinforcement learning. volume 2018-November.

9. **He, W., Hwang, K., Li, D. (2014).** Intelligent carpool routing for urban ridesharing by mining gps trajectories. IEEE Transactions on Intelligent Transportation Systems, Vol. 15.
10. **Huang, S. C., Jiau, M. K., Lin, C. H. (2015).** A genetic-algorithm-based approach to solve carpool service problems in cloud computing. IEEE Transactions on Intelligent Transportation Systems, Vol. 16.
11. **Huang, S. C., Jiau, M. K., Lin, C. H. (2015).** Optimization of the carpool service problem via a fuzzy-controlled genetic algorithm. IEEE Transactions on Fuzzy Systems, Vol. 23.
12. **Huang, S. C., Jiau, M. K., Liu, Y. P. (2019).** An ant path-oriented carpooling allocation approach to optimize the carpool service problem with time windows. IEEE Systems Journal, Vol. 13.
13. **Jiau, M. K., Huang, S. C. (2015).** Services-oriented computing using the compact genetic algorithm for solving the carpool services problem. IEEE Transactions on Intelligent Transportation Systems, Vol. 16.
14. **Liu, Y., Samaranayake, S. (2019).** Proactive rebalancing and speed-up techniques for on-demand high capacity vehicle pooling.
15. **Ma, S., Zheng, Y., Wolfson, O. (2013).** T-share: A large-scale dynamic taxi ridesharing service.
16. **Pei, Z., Hua, X., Han, J. (2008).** The clustering algorithm based on particle swarm optimization algorithm. Intelligent Computation Technology and Automation, International Conference on, Vol. 1, pp. 148–151.
17. **Shi, Y., Obaiahnahatti, B. G. (1998).** A modified particle swarm optimizer. volume 6, pp. 69–73.
18. **Shi, Y., Obaiahnahatti, B. G. (2001).** Fuzzy adaptive particle swarm optimization. volume 1, pp. 101–106.
19. **Shinde, T., Thombre, B. (2015).** An effective approach for solving carpool service problems using genetic algorithm approach in cloud computing. International Journal of Advance Research in Computer Science and Management Studies, Vol. 3.
20. **TLC (2016).** Tlc trip record data.
21. **Wallar, A., Zee, M. V. D., Alonso-Mora, J., Rus, D. (2018).** Vehicle rebalancing for mobility-on-demand systems with ride-sharing.
22. **Wen, J., Zhao, J., Jaillet, P. (2018).** Rebalancing shared mobility-on-demand systems: A reinforcement learning approach. volume 2018-March.
23. **Xiao, Q., He, R.-C., Zhang, W., Ma, C. (2014).** Algorithm research of taxi carpooling based on fuzzy clustering and fuzzy recognition. Jiaotong Yunshu Xitong Gongcheng Yu Xinxi/Journal of Transportation Systems Engineering and Information Technology, Vol. 14, pp. 119–125.
24. **Yao, Y. Z., Xu, Y. R. (2007).** Parameter analysis of particle swarm optimization algorithm. Harbin Gongcheng Daxue Xuebao/Journal of Harbin Engineering University, Vol. 28.
25. **Zhang, D., He, T., Liu, Y., Lin, S., Stankovic, J. A. (2014).** A carpooling recommendation system for taxicab services. IEEE Transactions on Emerging Topics in Computing, Vol. 2.

Article received on 16/04/2021; accepted on 21/12/2021.
Corresponding author is Moustafa Maaskri.

ISSN 2007-9737

A Combination of Sentiment Analysis Systems for the Study of Online Travel Reviews: Many Heads are Better than One

Miguel Á. Álvarez-Carmona[1,2*], Ramón Aranda[1,2],
Rafael Guerrero-Rodríguez[3], Ansel Y. Rodríguez-González[1,2],
A. Pastor López-Monroy[4]

[1] Consejo Nacional de Ciencia y Tecnología (CONACYT),
Mexico

[2] Centro de Investigación Científica y de Educación Superior de Ensenada,
Unidad de Transferencia Tecnológica,
Mexico

[3] Universidad de Guanajuato,
Mexico

[4] Centro de Investigación en Matemáticas,
Mexico

malvarez@cicese.edu.mx

Abstract. This study presents an analysis of the Rest-Mex forum task 2021, which is the first international evaluation event using tourism-related (Online Travels Reviews - OTRs) data from Mexico. In that forum, 14 specialized sentiment analysis systems were presented. The main contribution of this research is a method to successfully combine those 14 systems specialized on sentiment analysis systems for OTRs. The outputs of those 14 systems were used to evaluate the proposed combination schemes. The systems were trained and tested with 7,413 OTRs from the city of Guanajuato, Mexico, a well-known cultural destination. All of them were collected from TripAdvisor. We propose three schemes to combine the systems to predict the polarity of OTRs efficiently. The combination based on deep learning improves significantly each of the results obtained in the sentiment analysis systems at the individual level. Also, the results were improved for 4 out of the 5 polarity classes in the collection. To the best of our knowledge, this is the first paper that reports results from the combination of different specialized systems in sentiment analysis for OTRs.

Keywords. Sentiment analysis, OTRs, merge systems, deep learning, Mexican tourism.

1 Introduction

Tourism is a social, cultural, and economic phenomenon related to people's movement to places outside their usual place of residence for personal or business/professional reasons [13]. This activity is vital in various countries, including Mexico, where tourism represents 8.7% of the national GDP, generating around 4.5 million direct jobs [9, 14].

With the pandemic generated by the SARS-COV-2 virus, which spread out in Mexico in mid-March 2020, tourism was one of the most affected sectors [18]. This situation forced several economic sectors to pause their activities, with tourism being one of the most affected causing a disruption at different levels in activities such as accommodation, food services, transport, commerce, among others [12, 14].

Natural Language Processing (NLP) is an artificial intelligence area that has the potential to help in the recovery process of tourism by

generating mechanisms for detecting problems derived from the analysis of data from tourists shared on specialized digital platforms such as the case of Online Travels Reviews (OTRs). In this way, the tourism sector and the tourists themselves could be benefited by the NLP [6].

Sentiment analysis tasks in OTRs have gained relevance in the last decade [2, 10, 16, 17]. A significant goal of sentiment analysis is to classify and analyze the polarities of reviews related to products and experiences such as accommodation, online booking sites, e-commerce, social media, among others [25]. However, as with NLP, the most significant attention of scientific communication efforts have focused on the English language mainly. Although it is true that some studies have been conducted on Spanish language, only a few of them address data outside from the country of Spain.

One of the NLP specialized forums that have arisen due to the need to solve tasks related to the analysis of OTRs in Spanish language was Rest-Mex 2021 [3]. Rest-Mex 2021 edition was an international evaluation forum where one of the main tasks proposed by the organizers was to explore sentiment analysis on OTRs from the TripAdvisor website for tourist attractions in the Mexican city of Guanajuato. Among the places under study were the Museum of the Mummies, the University of Guanajuato, the Juarez Theater, the Basilica, the Hidalgo Market, the Union Square, the Alhóndiga de Granaditas and the Kissing Alley.

For this purpose, the organizers collected 7,413 opinions, where 5,197 of them were for the training phase. During this phase, the organizers released these labeled opinions to the scientific community. The participants of this event had the opportunity to build classification models based on machine learning using these 5,197 instances to train learning algorithms in order to predict the polarity of the OTRs in Spanish for the selected tourist attractions. The 2,216 remaining opinions were selected for the test process. This sub-collection was released unlabeled to the participants. The participants classified the instances with their respective models. Finally, participants sent their results obtained to the organizers to be evaluated.

Seven participating teams proposed fourteen different sentiment analysis systems to solve the task. The organizers analyzed the results for each of the systems.

The exciting thing about these results was that the systems are highly complementary. Theoretically, the combination of these systems reaches around 96.8% effectiveness, 56.7% being the best individual result. However, when taking advantage of the information from these systems, their combined result reaches 57.6%. Clearly this is not a substantial improvement considering that multiple systems come together. Therefore, the organizers themselves considered the possibility of finding fusion strategies of the participating results in order to get closer to the best theoretical result.

In this paper, we explore different ways of merging the different participating systems to improve the individual results of different sentiment analysis systems in OTRs in Guanajuato, Mexico.

The aim of this work is to answer the following research questions:

1. Is it possible to merge the Rest-Mex participants systems to improve the results of sentiment analysis for OTRs?
2. Which systems are most important to merge, and what are their characteristics?
3. What tourist attractions can benefit the most from this merging process at the practical level, and what are their characteristics?

The remainder of this paper is organized as follows: Section 2 describes the different international forums specialized in Mexican data and the phenomenon of completeness. Section 3 summarizes and classifies the participants' systems of the Rest-Mex 2021. Section 4 presents the proposal of this work, the methodology, the corpus, the performance measures, and the baselines. Section 5 describe the results obtained, and research questions are answered. Finally, Section 6 presents the conclusions and proposes directions for the future work derived from this study.

2 The Phenomenon of Completeness over Mexican Text Classification Tracks

For many years, NLP leading research focused on the English language considering the scientific community's available data. In order to generate data collections in Spanish language, some organizations such as CLEF, IberEval, or IberLef have organized campaigns to generate evaluation tasks with two main purposes [8,21]: (i) generating data in Spanish for different tasks and (ii) that the scientific community may propose specialized solutions for the Spanish language that can take advantage of this data.

Some of these tasks were proposed exclusively for Mexican Spanish. Among the most important is the Mex-A3t, which proposed to solve the task of author profiling and aggressiveness detection [7], FakeDeS [8] where the fake news detection task is performed, and more recently Rest-Mex [3] being the first to propose a task exclusively to analyze tourism-related data.

The Mex-A3t forum was the first study where the efficacy of the collective result of all participants was measured [4]. This measure was called theoretically Perfect Assembly. During all its editions, the Mex-A3t has reported that this measure gives a result higher than 95%. However, when trying to merge the participating systems, a result is obtained well below the theoretical perfect assembly where they do not even obtain different results to the best participating system. For the Rest-Mex 2021 edition, the same result was also reached. The organizers then proposed a fusion based on vote: first, taking into account all the systems, the best eight systems, the best five, and finally, the best three. However, the best vote result obtained an error of 0.47, where is the same result of the best individual system too.

These evaluation forums' results show that the different participating systems are highly complementary and that the upper bound is the perfect theoretical assembly. However, they have not been able to implement any method so that the fusion of the results surpasses the best-positioned result of each task. This may be because the voting approaches used in [3,4,7,8] are simple. There are more complex fusion methods based on stacking, which is a supervised learning method that learns from the mistakes and successes of the different systems and makes better decisions [5].

This is why this work proposes to generate a fusion of systems based on stacking in such a way that it surpasses at least the best individual results of the Rest-Mex forum when performs the task of sentiment analysis for OTRs.

3 Rest-Mex 2021 Participant Systems on Sentiment Analysis for OTRs

For this study, we propose to merge the solutions of fourteen systems from seven participating teams. This section summarizes and classifies their approaches.

Three different groups of systems were detected. The Transformers-based systems, which apply this type of deep learning architecture. The Bag of Words (BoW) based systems, and Meta-Features based systems.

Table 1 shows the descriptions and types of the participants systems.

4 Methodology

This section describes the proposed fusion, the Rest-Mex corpus characteristics, the baselines, and finally it shows the performance measures.

4.1 Merge methods proposed

There are three different ways of merging the different systems outputs:

1. Weighted vote,
2. Classic Stacking,
3. Deep learning Stacking.

4.1.1 Weighted Vote

For this non supervised method, each system results are merged giving a different level of importance to each system. This level of importance is awarded according to the ranking obtained in Rest-Mex forum. The weighted vote is defined in the following equation 1:

$$Wvote(i) = Max_c \left(\sum_{x=1}^{s} \frac{1}{Rank(S_x)} S_x(i) \right), \quad (1)$$

where s is the number of different systems in the Rest-Mex forum. i is the instance to classify, $S_x(i)$ is the class c returned by the system S_x in the instance i. $Rank(S_x)$ is the ranking obtained by the system S_x in the Rest Mex forum.

In this way, the class chosen, for some instance i will be the one that obtains the majority vote but giving more importance to the systems that obtained the best result.

4.1.2 Classic Stacking

For this method, we propose the application of a supervised approach. This approach takes the different systems' outputs class to generate an array as follows:.

$$arrayStacking(i) =< S_1(i), S_2(i), ..., S_{s-1}(i), S_s(i) > . \quad (2)$$

In this way, an *arrayStacking* can be generated for each instance within the test collection.

Once all the *arrayStacking* are built, it is possible to generate training models that learn from the errors and successes of each of the dimensions in the collection, where each dimension represents a competing system. In this way, classical classification algorithms would determine which systems to consider to obtain the final class. The algorithms that it is proposed to use are SVM, KNN[1], Decision Tree (DT), Random Forest (RF), and Naive Bayes (NB). It is also proposed apply a 10-cross validation scheme for their evaluation.

[1] with $k \in \{1, 3, 5, 7\}$

4.1.3 Deep Learning Stacking

This approach is very similar to classic stacking since *arrayStacking* is generated in the same way as in the equation 2. However, for this variant, it is proposed to use classification based on deep learning. In particular, a neural network with ten hidden layers to ensure that the best relationship is found between the outputs of the participating systems and the real class of each instance. Table 2 summarizes the principal characteristics of the Deep Learning algorithm proposed.

Just like the previous section, it is also proposed to apply a 10-cross validation scheme for their evaluation.

4.2 Rest-Mex Sentiment Analysis Corpus

For the Rest-Mex, the idea was to analyze Online Travel Reviews issued by tourists who visited the most representative tourist attractions in Guanajuato, Mexico. This collection was obtained from the tourists who shared their opinions on TripAdvisor between 2002 and 2020 [3]. Each opinion's class is an integer between [1, 5].

The corpus consists of **7,413 OTRs** shared by tourists. The organizers use a 70/30 partition to divide into train and test. This means that we used 5,197 labeled instances for the train partition, while 2,216 were used as unlabeled instances for the test partition [2].

Table 3 shows the distribution of the instances for the sentiment analysis task for the train and test partitions.

Table 4 shows the different attraction in the collection and their polarity average. Also, an OTR example is included per attraction in the original language for illustration purposes. It is possible to observe that there are places that have better rating by tourists. On the other hand, attractions such as Kissing Alley, Hidalgo Market, or the Museum of Mummies are the worst rated. It is also important to mention that the average is 4.27, which is consistent with the class imbalance since negative polarity appears infrequently in the

[2] The corpus is available and can be requested at https://sites.google.com/cicese.edu.mx/rest-mex-2021/corpus-request

Table 1. Systems description

Team	Type	Description
Minería UNAM	Transformers	They apply two Bert-based approaches for classification. The first approach consists of fine-tuning BETO, a Bert-like model pre-trained in Spanish. The second approach focuses on combining Bert embeddings with the feature vectors weighted with TF-IDF [23].
UCT-UA	Transformers	The team proposes two methods. The results in their primary submission were obtained from the model BETO. The secondary method has a better result for this team. This method consists of a cascade of binary classifiers based again on BETO. [1]
DCI-UG	Transformers	The proposed method is based on a modified Spanish BERT-base architecture model. The BERT-Base architecture was modified by removing the last layer of the network. Then, the last two layers of the modified BERT architecture were concatenated to be used as the input to a dense layer with a swish activation function. As a final layer, a dense layer was used with five outputs (one for each class) using softmax as activation function [24].
Labsemco UAEM	BoW	The team proposes an unsupervised method for keyword extraction in order to construct a list of prototypical words conveying a sentiment weight. Secondly, They emphasize the match of the scores of prototypical words with the labels of the texts where they appear. An SVM does the classification task applied to vector representations of text entities. [22]
Techkatl	BoW	For this system, the model development and experiments were carried out on the RapidMiner platform. The author proposes filtered stemming words as pre-processing. Their representation is based on TF-IDF. Also, the author applies several classification algorithms. Bayesian Methods obtain the best result [19].
Arandanito Team	Meta-Features	The team proposes a simple method based on naive features, which consist of extracting simple measures such as number of words, number of digits, empty words, among others. They test various classifiers and finally propose a weighting scheme to determine the best classification algorithm; for its representation, it was KNN with $k = 7$. [11]
The last	Meta-Features	The proposal of this team consists of calculating the Jaccard distance between each instance in the test participation with the average of each of the 5 classes in the train. Jaccard's distance is weighted by the number of repetitions of each word in each class. Finally, the KNN algorithm is used to determine the class of each instance in the test. [20]

collection. This makes the worst-rated places also the hardest to get good ranking results.

4.3 Performance Measures

Systems are evaluated using standard evaluation metrics, including Accuracy (equation 3), F-measure (equation 5) and MAE (equation 11). All equations involved to measuring the performance of an S_x system are described as follows:

$$Accuracy(S_x) = 100 * \frac{\sum_{i=1}^{n} correct(S_x(i))}{n}, \quad (3)$$

$$correct(S_x(i)) = \begin{cases} 1 \quad If \quad T(i) = S_x(i), \\ \\ 0 \quad Else \end{cases}, \quad (4)$$

$$F - measure(S_x) = \frac{1}{5}\sum_{c=1}^{5} F(S_x, c), \quad (5)$$

$$F(S_x, c) = 2 * \frac{Precision(S_x, c) * Recall(S_x, c)}{Precision(S_x, c) + Recall(S_x, c)}, \quad (6)$$

Table 2. Characteristics of the applied Deep Learning algorithm

Hidden Layers	10
Neurons per layer	1000
Activation function	Relu
Neurons of the final layer	5
Final layer	Softmax
Loss function	Categorical Cross Entropy
Optimizer	Adam
Epochs	50

$$Precision(S_x, c) = \frac{\sum_{j=1}^{n} p_c(j)}{\sum_{i=1}^{|c|} correct(S_x(i))}, \quad (7)$$

$$p_c(j) = \begin{cases} 1 & If \quad T(j) = c, \\ 0 & Else \quad , \end{cases} \quad (8)$$

$$Recall(S_x, c) = \frac{\sum_{j=1}^{n} r_c(S_x(j))}{\sum_{i=1}^{|c|} correct(S_x(i))}, \quad (9)$$

$$r_c(S_x(j)) = \begin{cases} 1 & If \quad S_x(j) = c, \\ 0 & Else \quad , \end{cases} \quad (10)$$

$$MAE(S_x) = \frac{1}{n} \sum_{i=1}^{n} |T(i) - S_x(i)|, \quad (11)$$

where S_x is a participating system x, $T(i)$ is the result of the instance i according to the Ground Truth, and $S_x(i)$ is the output of the participant system x for instance i. C is the classes set and $c \in C$. Finally, n is the number of instances in the collection.

Table 3. OTRs instances distribution for the Rest-Mex corpus

Class	Polarity	Train instances	Test instances
1	Very negative	80	35
2	Negative	145	63
3	Neutral	686	295
4	Positive	1596	685
5	Very positive	2690	1138
Σ		5197	2216

4.4 Baselines

As baselines, we propose to use the vote schemes applied by the Rest-Mex organizers. Also, they proposed the majority class as baseline. Finally, the most crucial baseline for our work is the best-ranked result for the forum., This result is obtained by *Minería Unam* team. It is important to mention that the measure the organizers proposed to rank the systems was MAE. For this reason, we will also use it to rank the final results.

5 Experimental Results

Table 5 shows a summary of the results obtained by each team for the sentiment analysis task [3]. For systems with *B* are the baselines, with *P* are the fusion methods proposed; others are normal participants systems of the forum.

The worst results obtained by the proposed methods are those of classic stacking since the best result is obtained by SVM with 0.49 of MAE, 0.34 of F-measure, and 58.03 of Accuracy. Those results are below the majority of baselines and mainly below the best individual result.

The weighted average obtains a better result than SVM. However, it also is below the best individual result.

Finally, only the proposal based on Deep Learning improves all baselines for all metrics, and it is the closest system to Perfect Assembly. For MAE, it is obtained 0.41, 0.58 for F-measure, and 62.59 for Accuracy.

5.1 Analysis of the Results

In this section, we aim to provide answers to the research questions proposed in this study.

[3]* The authors did not send the system's description to the organizers' forum.

5.1.1 Is it Possible to Merge the Rest-Mex Participants Systems to Improve the Results of Sentiment Analysis for Mexican OTRs?

It is possible to merge the participating systems and obtain a considerably better result. However, it is not a straightforward task; it was necessary to resort to one of the approaches that have had the best results in recent years in artificial intelligence: Deep Learning, since other approaches that are also complex were not able to improve the best individual result obtained.

Table 6 shows the best F-measure results by class. It can be seen that the improvement of the merge has a more significant impact on the minority classes. Since for class 5 (very positive), although there is an improvement, it is smaller than for classes 1 (very negative), 2 (negative), and 3 (neutral). Class 4 (positive) was the only one where there was no improvement, and the result of the Minería UNAM team achieved a better result.

5.1.2 Which Systems are Most Important to Merge, and What are their Characteristics?

Table 5 also shows the Information Gain (IG) of each system for the array stacking representation. It is possible to see that, as might be expected, the best systems also have the highest information gain values. This means that those results are the most valuable for the merger. It is even possible to see an inverse correlation between the information gain and MAE of -0.62, a direct correlation of 0.70 with Accuracy, and a very strong direct correlation of 0.93 with F-measure.

It is also clear to see that the transformer-based systems were the most successful both in their individual result and in contributing to the merge results. This could indicate that using only this type of system could help reduce noise and obtain better results.

However, it was not the case. Table 5 also shows the result obtained by the *Deep Learning T* system, which is the same scheme used by the method based on Deep Learning that obtained the best result but only using the systems based on transformers and failed to obtain a result above the baselines. This indicates that although the systems based on BoW and Meta-Features do not have much impact on the merge, their contribution is also essential, and with them, it is possible to surpass all the individual results and baselines. Therefore, it is concluded that although not all systems are equally important, they all provide valuable information.

5.1.3 What Tourist Attractions can Benefit the Most from this Merging Process at the Practical Level, and What are their Characteristics?

The tourist attractions in Guanajuato that obtained a significant improvement when classified using the proposed merge method are Hidalgo Market, the Kissing Alley and the Museum of the Mummies. In particular, the OTRs that have a negative polarity (class 1 and 2), which shows an improvement from 38% of opinions well classified by the best system up to 60% with the proposed method.

This was expected since, as seen in Table 6, the main improvements occurred within these classes. This coincides with the fact that these attractions are among the worst rated by the travelers on TripAdvisor as Table 4 shows.

The correlation coefficient among the average polarity and improved ranking by place is 0.69, which means that while the more negative the overall rating given by tourists to a tourist attraction, the proposed system will have better results than the best individual result.

This supposes clear practical advantages for destinations since the opinions and themes surrounding tourist attractions with negative evaluations from tourists can be detected more efficiently and solutions can be designed accordingly in a shorter period of time. These can range from decision-making by tourism service providers to public policies involving all tourism stakeholders. For more details over themes and problems identified in the collection and these particular tourist attractions in Guanajuato [15].

Table 4. Example instances corpus and polarity average per attraction

Attraction	Polarity Avg	Example
Juarez Theater	4.70	Tomar un tour por este lugar es muy impresionante y bello, la arquitectura, los acabados y toda la historia de lugar es muy interesante
University of Guanajuato	4.60	Me gustó bastante este lugar, solo la pude contemplar desde afuera y me gusto mucho. Definitivamente un Must Visit en Guanajuato.
Union Square	4.59	Hay lugares donde comer muy rico!, de noche es muy bonito y romantico, escuchas a las estudiantinas tocar
Básilica de Guadalupe	4.50	Esta venerable iglesia, ahora basílica de Nuestra Seõra de Guanajuato, es uno de los mejores ejemplos de arquitectura barroca del siglo XVII. En su interior, uno puede admirar la antigua figura de la virgen, Patrona de Guanajuato.
Alhóndiga de Granaditas	4.45	Tiene una variadísima colección de piezas que abarcan siglos, desde antes de la conquista hasta nuestros días. Vale la pena dedicarle tiempo ma apreciar sus murales y las diferentes salas de exhibición.
Pipila Monument	4.27	Nos aconsejaron no ir caminando, es peligroso por los robos a turistas en el trayecto. Se recomienda en visitas guíadas o vehículos
Diego Rivera Museum	4.24	Este museo de tres pisos se vende como sede de muchas obras de Diego Rivera, sin embargo, después de recorrer todo el museo, y ante la frustración de no encontrar más que dibujos y bocetos, decidí preguntarle a uno de los guardas, aquí me aclaró que las obras de dos pisos completos se encuentran en restauración, y en otra exhibición en Japón. No dejando así al público ni una sola hora de pintura para apreciar.
Kissing Alley	3.95	Es un lugar público y como tal se congrega mucha gente solo para tener la oportunidad de tomarse una foto en el famoso callejón, lamentablemente es un caudal de gente que es casi imposible.
Hidalgo Market	3.94	Si vas corto de tiempo, no te molestes en incluir este punto en tu recorrido. Lo que encuentras son playeras con el nombre de la cuidad, artesanías de barro y dulces (que igual los encuentras por toda la ciudad y en cada paseo).
Museum of the Mummies	3.60	Grotesco espectáculo después de dos horas de cola! Vean mejor la película del Santo contra las momias.
Average	**4.27**	

6 Conclusions and Future Work

This work proposed three schemes to merge different systems specialized in sentiment analysis for OTRs. The weighted vote and classic stacking approaches failed to improve the proposed baselines. However, the proposed method based on deep learning surpasses the individual results and the other merge methods.

There is evidence that it is possible to take advantage of the collective information to improve each system. However, given the results, it is

Table 5. Performance of all systems in Sentiment Analysis for OTRs

IG	System	MAE	F-measure	Accuracy	Type
-	*Perfect Assembly*	*0.06*	*0.94*	*96.84*	-
-	Deep Learning*(P)*	**0.41**	**0.58**	**62.59**	-
-	3 best results*(B)*	0.47	0.47	57.67	-
0.307	Minería UNAM$_{Run1}$*(B)*	0.47	0.42	56.72	Transformers
-	Deep Learning T(P)	0.48	0.41	58.43	-
-	Weighted vote*(P)*	0.49	0.39	58.03	-
-	5 best results*(B)*	0.49	0.39	57.89	-
-	SVM*(P)*	0.49	0.34	58.03	-
-	8 best results*(B)*	0.50	0.33	57.53	-
-	KNN-7*(P)*	0.52	0.37	56.00	-
-	NB*(P)*	0.53	0.39	55.68	-
-	RF*(P)*	0.53	0.38	53.70	-
0.299	UCT-UA$_{Run2}$	0.54	0.45	53.24	Transformers
-	KNN-5*(P)*	0.54	0.38	54.10	-
0.272	UCT-UA$_{Run1}$	0.56	0.40	53.83	Transformers
0.149	DCI-UG$_{Run1}$	0.56	0.28	53.33	Transformers
0.125	Minería UNAM$_{Run2}$	0.58	0.24	54.78	Transformers
0.130	DCI-UG$_{Run1}$	0.60	0.25	53.70	Transformers
-	KNN-3*(P)*	0.61	0.34	50.27	-
-	DT*(P)*	0.63	0.33	47.42	-
0.143	Labsemco-UAEM$_{Run1}$	0.64	0.30	49.05	BoW
-	KNN-1*(P)*	0.65	0.29	46.25	-
0.061	TechkatI$_{Run1}$	0.66	0.27	50.18	BoW
-	Majority class*(B)*	0.72	0.13	51.35	-
0.006	Arandanito Team	0.76	0.16	45.71	Meta-Features
0.002	TextMin-UCLV*$_{Run1}$	0.78	0.17	36.23	-
0.008	TechkatI$_{Run2}$	0.81	0.21	44.76	BoW
0.028	Labsemco-UAEM$_{Run2}$	0.91	0.24	36.50	BoW
0.002	TextMin-UCLV*$_{Run2}$	1.00	0.18	38.31	-
0.091	The last	1.26	0.21	36.95	Meta-Features
Correlation	-	-0.62	0.93	0.70	-
Type	Avg IG	Avg MAE	Avg F	Avg Acc	-
Transformers	0.213	0.55	0.34	54.26	-
BoW	0.060	0.75	0.25	45.12	-
Meta-Features	0.048	1.01	0.18	41.33	-

Table 6. Performance per class

F-measure class	Team	Team result	Deep learning result	Improvement
1	UCT-UA$_{Run2}$	0.37	**0.72**	48.62%
2	UCT-UA$_{Run2}$	0.39	**0.54**	27.78%
3	Minería UNAM$_{Run1}$	0.47	**0.51**	7.84%
4	Minería UNAM$_{Run1}$	**0.44**	0.37	-15.9%
5	Minería UNAM$_{Run2}$	0.71	**0.76**	6.57%

concluded that it is not a trivial task and that there is still a wide margin for improvement since the perfect assembly obtains an error result of 0.06 while the approach proposed in this work obtained 0.41.

Although the transformer-based approaches are the most valuable to the mix, all the others also provide valuable information; it is recommended to use all available systems to face this type of task. Collective intelligence is clearly better; in other words ”many heads are better than one”.

The classes that have a significant improvement are the minority. In the sentiment analysis task, the minority classes are usually those with negative polarity (1 and 2), so indirectly, the classification of

tourist places that have the most negative OTRs in the collection benefited the most.

The system proposed in this work was applied exclusively to the domain of OTRs related to tourist attractions. However, we consider that it is possible to apply it to multiple types of tourist domains, where OTRs are fundamental such as accommodation and food experiences, service satisfaction, travel and purchase decision-making, destination management, evaluation of destination image and so on. As stated previously in this paper, this type of solution can help to gain a better understanding of the traveling experience directly from the voice of the travelers themselves.

Future work can be directed towards the design and implementation of more complex deep learning architectures so as to get even closer to the result of the perfect assembly. It is considered also essential to explore multilingual collections since these types of mixes are independent of language.

References

1. **Abreu, J., Mirabal, P. (2021).** Cascade of biased two-class classifiers for multi-class sentiment analysis. Proceedings of the Third Workshop for Iberian Languages Evaluation Forum (IberLEF 2021), CEUR WS Proceedings.
2. **Alaei, A. R., Becken, S., Stantic, B. (2019).** Sentiment analysis in tourism: Capitalizing on big data. Journal of Travel Research, Vol. 58, No. 2, pp. 175–191.
3. **Álvarez-Carmona, M. Á., Aranda, R., Arce-Cárdenas, S., Fajardo-Delgado, D., Guerrero-Rodríguez, R., López-Monroy, A. P., Martínez-Miranda, J., Pérez-Espinosa, H., Rodríguez-González, A. (2021).** Overview of Rest-Mex at IberLEF 2021: Recommendation system for text Mexican tourism. Procesamiento del Lenguaje Natural, Vol. 67.
4. **Álvarez-Carmona, M. Á., Guzmán-Falcón, E., Montes-y Gómez, M., Escalante, H. J., Villaseñor-Pineda, L., Reyes-Meza, V., Rico-Sulayes, A. (2018).** Overview of MEX-A3T at IberEval 2018: Authorship and aggressiveness analysis in Mexican Spanish tweets. Notebook papers of 3rd sepln workshop on evaluation of human language technologies for iberian languages (ibereval), seville, spain, volume 6.
5. **Álvarez Carmona, M. Á., Villatoro Tello, E., Montes y Gómez, M., Villaseñor-Pineda, L. (2020).** Author profiling in social media with multimodal information. Computación y Sistemas, Vol. 24, No. 3, pp. 1289–1304.
6. **Anis, S., Saad, S., Aref, M. (2020).** A survey on sentiment analysis in tourism. International Journal of Intelligent Computing and Information Sciences, pp. 1–20.
7. **Aragón, M. E., Álvarez-Carmona, M. A., Montes-y Gómez, M., Escalante, H. J., Villaseñor-Pineda, L., Moctezuma, D. (2019).** Overview of MEX-A3T at IberLEF 2019: Authorship and aggressiveness analysis in Mexican Spanish tweets. IberLEF@ SEPLN, pp. 478–494.
8. **Aragón, M. E., Jarquín-Vásquez, H. J., Montes-Y-Gómez, M., Escalante, H. J., Villaseñor-Pineda, L., Gómez-Adorno, H., Posadas-Durán, J. P., Bel-Enguix, G. (2020).** Overview of MEX-A3T at IberLEF 2020: Fake news and aggressiveness analysis in Mexican Spanish. IberLEF@ SEPLN, pp. 222–235.
9. **Arce-Cardenas, S., Fajardo-Delgado, D., Álvarez-Carmona, M. Á., Ramírez-Silva, J. P. (2021).** A tourist recommendation system: A study case in Mexico. Mexican International Conference on Artificial Intelligence, Springer, pp. 184–195.
10. **Brahimi, B., Touahria, M., Tari, A. (2020).** Improving Arabic sentiment classification using a combined approach. Computación y Sistemas, Vol. 24, No. 4.
11. **Carmona-Sánchez, G., Carmona, A., Álvarez-Carmona, M. A. (2021).** Naive features for sentiment analysis on Mexican touristic opinions texts. Proceedings of the Third Workshop for Iberian Languages Evaluation Forum (IberLEF 2021), CEUR WS Proceedings.
12. **Crick, J. M., Crick, D. (2020).** Coopetition and covid-19: Collaborative business-to-business marketing strategies in a pandemic crisis. Industrial Marketing Management, Vol. 88, pp. 206–213.
13. **Di-Bella, M. G. (2019).** Introducción al turismo.
14. **Elorza, S. R. (2020).** Turismo y sars-cov-2 en México. Perspectivas hacia la nueva normalidad. Desarrollo, economía y sociedad, Vol. 9, No. 1, pp. 93–98.

15. **Guerrero-Rodriguez, R., Álvarez-Carmona, M. Á., Aranda, R., López-Monroy, A. P. (2021).** Studying online travel reviews related to tourist attractions using NLP methods: The case of Guanajuato, Mexico. Current Issues in Tourism, pp. 1–16.

16. **Maitama, J. Z., Idris, N., Abdi, A., Bimba, A. T. (2021).** Aspect extraction in sentiment analysis based on emotional affect using supervised approach. 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD), IEEE, pp. 372–376.

17. **Masmoudi, A., Hamdi, J., Belguith, L. H. (2021).** Deep learning for sentiment analysis of Tunisian dialect. Computación y Sistemas, Vol. 25, No. 1, pp. 129–148.

18. **Rivas Díaz, J. P., Callejas Cárcamo, R., Nava Velázquez, D. (2020).** Perspectivas del turismo en el marco de la pandemia covid-19.

19. **Roldán Reyes, E. (2021).** Techkatl: A sentiment analysis model to identify the polarity of Mexican's tourism opinions. Proceedings of the Third Workshop for Iberian Languages Evaluation Forum (IberLEF 2021), CEUR WS Proceedings.

20. **Romero-Cantón, A., Aranda, R. (2021).** Sentiment classification for Mexican tourist reviews based on K-NN and Jaccard distance. Proceedings of the Third Workshop for Iberian Languages Evaluation Forum (IberLEF 2021), CEUR WS Proceedings.

21. **Stamatatos, E., Potthast, M., Rangel, F., Rosso, P., Stein, B. (2015).** Overview of the PAN/CLEF 2015 evaluation lab. International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, pp. 518–538.

22. **Toledo-Acosta, M., Martńez-Zaldivar, B., Ehrlich-López, A., Morales-González, E., Torres-Moreno, D., Hermosillo-Valadez, J. (2021).** Semantic representations of words and automatic keywords extraction for sentiment analysis of tourism reviews. Proceedings of the Third Workshop for Iberian Languages Evaluation Forum (IberLEF 2021), CEUR WS Proceedings.

23. **Vásquez, J., Gómez-Adorno, H., Bel-Enguix, G. (2021).** Bert-based approach for sentiment analysis of Spanish reviews from tripadvisor. Proceedings of the Third Workshop for Iberian Languages Evaluation Forum (IberLEF 2021), CEUR WS Proceedings.

24. **Velazquez Medina, G., Hernández Farías, D. I. (2021).** DCI-UG participation at Rest-Mex 2021: A transfer learning approach for sentiment analysis in Spanish. Proceedings of the Third Workshop for Iberian Languages Evaluation Forum (IberLEF 2021), CEUR WS Proceedings.

25. **Yadav, A., Vishwakarma, D. K. (2020).** Sentiment analysis using deep learning architectures: A review. Artificial Intelligence Review, Vol. 53, No. 6, pp. 4335–4385.

Article received on 03/10/2021; accepted on 19/10/2022.
Corresponding author is Miguel Á. Álvarez-Carmona.

ISSN 2007-9737

Identification of POS Tags for the Khasi Language based on Brill's Transformation Rule-Based Tagger

Sunita Warjri[1], Partha Pakray[2], Saralin A. Lyngdoh[3], Arnab Kumar Maji[1]

[1] North-Eastern Hill University,
Department of Information Technology,
India

[2] National Institute of Technology, Assam,
Department of Computer science and Engineering,
India

[3] North-Eastern Hill University,
Department of Linguistics,
India

{sunitawarjri, parthapakray, saralyngdoh, arnab.maji}@gmail.com

Abstract. Khasi is a Mon-Khmer language that belongs to the Austro-Asiatic language family. Khasi language is spoken by the indigenous people of the state Meghalaya in the North-Eastern part of India. The main purpose of this paper is to develop Part-of-Speech (PoS) tagger for the Khasi language using a Rule-based approach. To work on POS tagging, one needs a grammatically tagged corpus. However, the Khasi language does not have a standard corpus for PoS tagging. Therefore, another aim or purpose of this paper is to develop a Khasi lexicon or POS corpus and using the Rule-Based Brill's Transformation to automatically tag the given Khasi text. While anticipating the challenges in building such a corpus, this paper has brought out an analysis based on the Khasi corpus of around 1,03,998 words in its initial phase. We also show in this paper how the Khasi corpus is created. By using Brill's Transformation rule-based learning on the created corpus in this preliminary study, accuracies of 97.73% and 95.52% were obtained on validating data and testing data respectively. This work is the first attempt to investigate POS tagging using the rule-based model with the designed Khasi POS corpus.

Keywords. Natural language processing (NLP), computational linguistic, part-of-speech (PoS), PoS tagging, Khasi language, Khasi corpus, lexical morphology, transformation rule-based tagging.

1 Introduction

Natural Language Processing (NLP) is a zone of Machine Learning, where Natural Languages are made to interact with computer systems. This process of interaction involves linguistic analysis that combines with the computational power of Artificial Intelligence (AI) and Computer Science (CS), i.e. NLP involves the power of natural language theory and its applied component. To make the computer automatically understand the natural language either in the form of text or speech, many fields of NLP is involved. One such field is Part-of-Speech (PoS) Tagging or PoST.

PoS tagging means assigning grammatical class information to the words of a given sentence, according to its context. A system that results in identifying tags for the given input text is called PoS Tagger. POS tagging is also known as grammatical tagging or word category disambiguation. Part-of-Speech (PoS) tagging system is a very strong backbone of NLP as it can be used for other fields of NLP such as Named Entity Recognition (NER), Information

Extraction, Translation of language, and other NLP applications.

To proceed towards PoS tagging, one needs to study in detail the nature of the preferred language and to understand its structure and its grammatical flow. For PoS tagging, tagsets are needed. Tagset is a set that consists of tags or labels. These tags or labels describe the classes of grammatical part-of-speech, which can be used for annotating words of a particular language. For example, tags are basically labels such as NN, ADJ, which represent Nouns and Adjectives respectively.

One of the PoS taggers was built by E. Brill [6]. This PoS tagger was developed by using the rule-based algorithm. The rule-based tagging approach is still commonly used today for tagging for languages. The rule-based tagging is also called a Transformation based tagging method. This transformation-based tagging automatically tags the given input words and produces the word along with its belonging tag as an output. Contextual rules and regular expression rules are the main components for developing the transformation rule-based algorithm. This algorithm is the oldest method towards checking on the context rules.

Other tools that perform POS tagging include Stanford Log-linear Part-of-Speech Tagger [25], Tree Tagger [21], Hidden Markov Model (HMM) based Tagger [4], and etc. Though, there are other tools for the PoS tagging method, Brill's transformation rule-based method has been opted and used in this paper.

As each language differs in terms of the utterance, spelling system of a language, writing, and also the flow and formation of the sentences. This makes the grammatical rules to be different in different languages. Therefore, in this paper, we perform research and analyze the Khasi language using this transformation rule-based approach. In Brill's transformation rule-based method, one does not have to design the complex detailed rules for a particular language. The transformation ruled based learning has the property of complex crossing with the manually generated rules and the machine-learned rules from the corpus.

Transformation rule-based tagger is said to be more accurate than Hidden Markov Model (HMM) based Tagger [4, 13] in terms of producing and generating high accuracy and also to tag the given words correctly. However, some related research works show HMM to be more accurate than the rule-based method [11, 12, 18]. This is indicative of the fact that shows that different languages exhibit different flow on different systems.

Research works or literature on Khasi from Computational Linguistics (CL) perspectives are inadequate. PoS tagging being the most important initial task towards other fields of NLP, we believe that this research is going to contribute to the realm of NLP study of Khasi language. It will help this indigenous language of India to develop and get recognition in the world as well as in the country.

The paper is organized as follows: Section 2 describes the existing related works on POS Tagging; Section 3 describes methodology used in Khasi Part-of-speech Tagging (KPOST); Section 4 describes some of challenges for Khasi corpus building; Section 5 shows the experimental results ; Whereas Section 7 consists of Conclusions and some future perspectives of the work.

2 Existing Literature on Rule-Based Tagger

The rule-based approach has been used extensively in many languages. In this section, we will have a brief discussion on some of the NLP related works, that have used the rule-based methodology.

In paper [15], the rule-based and statistical approach has been introduced in the study of the Arabic language. Pre-processing of the lexicon has been done before tagging automatically, as the Arabic language is morphologically rich. In this paper, the corpus has been created manually using the tag set comprising of 131 tags. Data were taken from newspaper and published paper for building the corpus: from "AlJazirah" newspaper 59,040 words, from "Al-Ahram" newspaper 3,104 words, from "Al-Bayan" newspaper 5,811 words, and from "Al-Mishkat" published paper in social science 17,204 words were taken. After disambiguation of the ambiguous words, the accuracy of around 90% is achieved from this statistical tagger.

ISSN 2007-9737

In paper [1], tagging on Modern Arabic text has been carried out using the Transformation-based learning technique. The Arabic Tree Bank (ATB) [10] has been used as a corpus in this paper. The corpus consists of 770k words. The annotated corpus includes the syntactic trees and morphological analysis also. Experiments are conducted twice in this research work. The training accuracy of 98.50% is achieved from experiment 1 and 97.90% from experiment 2. Evaluation of testing data has also been carried out and the accuracy 96.90% and 96.15% are obtained for experiment 1 and experiment 2 respectively.

In paper [9], a discussion is made on the rule-based technique for Part-of-Speech (PoS) tagging and Named Entity Recognition (NER), for the Arabic language. For PoS tagging, the lexicon phase and morphological phase have been used. The POS tagger had been tested on 793 words, out of which 679 words are correctly tagged. For NER tagging, 480 words are tagged correctly from 490 different words.

In paper [16], authors present POS tagging about the Sindhi language, where the Rule-based approach has been applied. In this work, rules are framed to disambiguate the words and a lexicon is developed for the Sindhi language. Tags set consisted of 67 different tags, which are used for designing supervised corpora. The training corpus, which is used in this paper, has 26366 words and a testing corpus of 6783 words are used. An accuracy of 96.28% is achieved for the Sindhi language. A Sindhi linguist has verified the data analyzed in this paper.

Several Indian languages are also explored using this methodology. Some of them are described in this section. In paper [23], POS tagging for Manipuri language has been discussed using the Rule-based method. The author has segmented the affixes from the roots by using the stripping technique. Fewer corpus resources are used in this work. Using some POS rules, the system is able to achieve an accuracy of 50% for 100 words with 5 rules, 77% for 500 words with 15 rules and 85% for 1000 words with 25 rules.

In paper [10], POS tagging using a rule-based approach with layered tagging has been introduced for Bangla language. Morphological analysis is performed for Nouns and Verbs. Based upon the postfix of words, morphological analysis is generated. With regard to morphological features, the words were classified to post positions, numbers, and classifiers. The verbs are classified according to the morpho-syntactic formation of the root and then classified to honorific and persons. A four-level layered architecture of the work has been discussed in this work. Assignment of tags to words at level 1, rules to disambiguate at level 2, multiple word categories of the verb at level 3 and in level 4, chunking of words are performed.

In paper [12], Rule-Based Part-of-Speech Tagger has been used to analyze the Hindi language. Corpus for this Tagger consists of 26,149 tagging words with 30 tags. The data are collected for the corpus consists of short stories, news, and essays. Using a rule-based tagger, the system achieved 87.55% as overall accuracy on the testing data.

Also, Recall, Precision, and F-Measure parameters are used for the result computation. The Recall score, that has been achieved in this work, is 92.84% for news, 87.32% for essays and 88.99% for short stories. The precision score is 89.94% for news, 81.36% for essay 85.11% for short stories. An F- score of 91.37% for news, 84.23% for essay and 87.06% for short stories, is achieved in this work.

In paper [14] also, PoS tagging for Bangla language using the rule-based approach has been explored. This tagging system is based on words suffix rules and stemming technique. Corpus having 45,000 words along with their corresponding tags is used in this system. Using some ruleset with a verb dataset, the accuracy of 93.7% is achieved, which is significantly improved accuracy than the other existing POS tagging system for Bangla.

In paper [22], POS tagging and morphological analysis for the Tamil language has been discussed. The alignment and projection techniques have been used to project the POS tags. Lemmatization (i.e. to analyze vocabulary and morphological words correctly) and induction methods are also employed for getting the root words from English to Tamil. During the testing phase, 85.56% accuracy is achieved for Bible

corpus and accuracy of 83% for CIIL corpus. An improvement of the system is also proposed, which obtained the accuracy of 92.48% for Bible test corpus.

The achieved accuracy is having an improvement of 7% in comparison to the previous accuracy of 85.56%.

In the paper [19], Part-of-Speech tagging for the Indonesian language is presented using rule-based approach. In this work, the Indonesian large dictionary or KBBI is utilized with some morphological rules for POS tagging. Using PAN Localization corpus in 4 parts for Indonesian an average accuracy rate of 87.4% is achieved.

The paper [20], describes part-of-speech tagging for the Indonesian language using a rule-based approach. The manually tagged corpus is used in this work which consists of roughly 250.000 tokens. Using the corpus the system yields an accuracy of 79%.

Some PoS tagging work on Khasi language has also been done. In paper [24], Khasi POS tagging based on HMM tagger had been discussed. The corpus consists of 86,087 tokens with 5,313-word types. NLTK tool tagger has also been applied to the same corpus in [24]. Accuracy of 86.76%, 88.23%, 88.64%, 89.7%, 95.68% is obtained for Baseline Tagger, NLTK Bigram Tagger, NLTK Trigram tagger, NLTK Tagger, and HMM POS Tagger respectively.

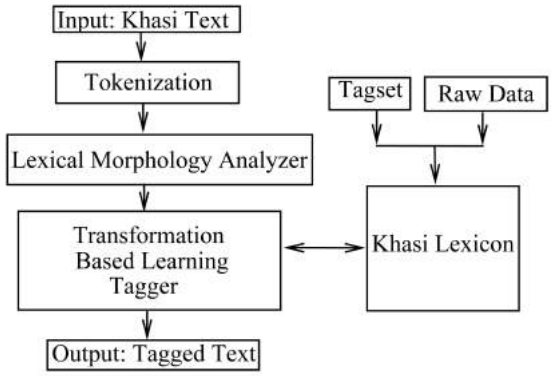


Fig. 1. Architecture for Khasi Part-of-Speech tagging (KPOST)

In paper [28], the Khasi POS tagger has been developed based on the HMM method. The lexicon consists of around 7,500 words for training and 312 words for testing data. Using the manually tagged Khasi lexicon on the HMM-based POS tagger, an accuracy of 76.70% was obtained. Also POS tagging using CRF for Khasi had been discussed in paper [29, 30].

Concerning Rule-based, it is also used in different fields of NLP such as Machine translation. In paper [2], noun phrase translation using a rule-based approach has been discussed. This automatic translation was done from the Punjabi language to the English language.

For this purpose, many steps are taken place such as: 1. Pre-processing, 2. Tagging, 3. Ambiguity Resolution, 4. Translation, and, 5. Synthesis. Around 2000 phrases are used for training the system and 500 sentences are used for testing. The overall translation accuracy, which is achieved by the system is around 85%.

3 Methodology for Khasi Part-of-Speech Tagging (KPOST)

In this paper, the POS tagger for Khasi language, based on Brill's Transformation and Khasi morphological rules, have been employed. In the subsections below, there is a brief discussion on the methods that have been carried out in this work. Architecture for KPOST is shown in Figure 1.

3.1 Tokenizer

This is the first step for POS tagging. It is used for separating the required words, symbols, and punctuation of the given text by assigning space between words. In our designed corpus, we have split some of the words for more clarifications. Therefore, for the given input text, split or replacement of words, is also done accordingly as per requirement based on our designed corpus.

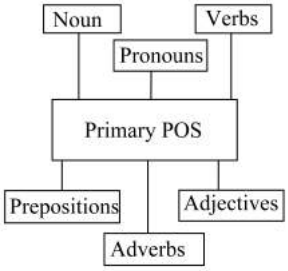


Fig. 2. Classes of Primary Parts-of-Speech

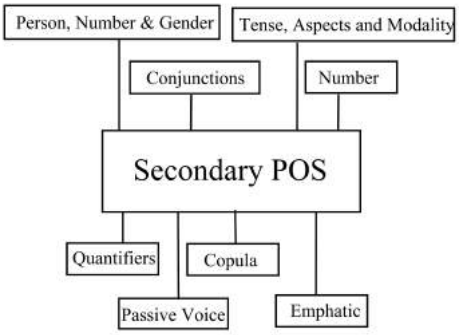


Fig. 3. Classes of Secondary Parts-of-Speech

3.2 Khasi Lexicon

Lexicon is a large collection of vocabulary data, along with its corresponding information, which is used for analyzing computational linguistic. In this work, for building the Khasi lexicon, we have used tag sets comprising of 53 tags. Out of 54 tags discussed in [27], we have used only 53 tags. The tag-sets that we have used in our proposed work can be found in Table 1. Simple graphical representations of the proposed grammatical classes used in the designed tag sets are shown below in Figures 2 and 3.

Figures 2 and 3 represent the visual art of classes of Part-of-Speech (POS), used for categorizing Khasi text in this work. Figure 2 represents the Class of Primary Part-of-Speech and Figure 3 represents the Class of Secondary Parts of Speech. More details regarding the grammatical classes used for POS tagging, designed tag-sets of Khasi language can be found in [27].

The Khasi corpus has been built manually, by tagging the raw text or words to its corresponding tags by using the designed tagsets. The manually tagged data represent the POS information of each Khasi word in the corpus. The raw data for corpus have been collected form Khasi newspapers [17], which are available online. The raw data that are collected mainly comprises of the political news and article news. Prepossessing of the corpus context, such as splitting and orthography correction are done manually.

However, the information maintained in the Khasi corpus or lexicon which we have discussed in paper [28] differs from the corpus data used in this paper. The corpus size is more in this present work than the earlier [28] as well as changes are made is few tagged words.

In this work, we have to build a Khasi corpus manually, which consists of around 1,03,998 words, with 6645 numbers of distinct words. For example, some Tagged Khasi sentences from the corpus are shown in Table 1.

These manually tagged Khasi words in the lexicon corpus are created under the observation and verification of a linguistic expert from the Department of Linguistics, North-Eastern Hill University, Shillong, India.

As this research work is an initial work towards computational Linguistic for Khasi, as a remark, We are releasing some of the designed Khasi POS corpus data. The corpus is available online at [26].

3.3 Lexical Analyzer

The Lexical analyzer is used to analyze the given input words, according to the predefined lexical rules. Lexical rules are used as foundation for analyzing formation of words. Lexical rules are the basic special compositions or building blocks of a word vocabulary. Lexical rules are recorded to generate a productive resource in computational linguistic. There are three types of Lexical rules [3, 5]. They are 1. Inflectional (lexeme to word), 2. Derivational (lexeme to lexeme), and 3. Post-Inflectional (word to word). To analyze morphological processes of Khasi words, Derivational method is widely used in this work. Some prefixes of the words are

Table 1. Khasi Lexicon: Manually tagged dataset for training

Sentences	Khasi Lexicon
1	Shi/QNT sngi/CMN hadien/ADP ba/COM la/VST dep/ITV ka/3PSF jingïathep/ABN vote/FR ba/COM n/VFT jied/TRV ïa/IN ki/3PPG MDC/FR jong/POP ka/3PSF KHADC/FR bad/COC JHADC/FR ,/SYM ka/3PSF lyer/CMN ka/3PSF sdang/TRV ba/COM n/VFT beh/TRV ba/COM ka/3PSF jingïakhun/ABN ka/3PPG n/VFT long/CO hi/EM hapdeng/ADP ka/3PSF Congress/FR ,/SYM ka/3PSF UDP/FR bad/COC ka/3PSF NPP/FR ha/IN KHADC/FR bad/COC ha/IN JHADC/FR ka/3PSF jingïakhun/ABN ka/3PPG n/VFT long/CO hapdeng/ADP ka/3PSF UDP/FR bad/COC ka/3PSF NPP/FR ./.
2	Ki/3PPG ba/COM bun/QNT ki/3PPG riew/CMN sngap/ITV lyer/CMN ki/3PPG la/VST sdang/TRV ba/COM n/VFT ïathuh/TRV lypa/AD ruh/COC ne/CRC ba/COM n/VFT ‘/SYM predict/FR ’/SYM ba/COM na/IN kata/DMP ka/3PSF Constituency/FR yn/VFT jop/ITV uta/DMP ,/SYM yn/VFT jop/ITV kata/DMP ./.
3	Ki/3PPG n/VFT don/CO napdeng/ADP kine/DMP ki/3PPG ba/COM lah/VSP ba/COM n/VFT dei/CO bad/COC ki/3PPG n/VFT don/CO ruh/COC ki/3PPG ba/COM n/VFT lait/ITV ./.

also analyzed, by using lexical rules to identify the POS categories of the given words. The morpho-syntactic processes, that are engaged for the analysis of lexical resource, are nominalization, agentivization, and causation. In Khasi, a series of prefixes indicate the existence of double morphological rules. For instance, nominalization, followed by causativization and by root word, as in *“Nongpynsum"* which means “one who causes/give a bath to somebody”.

NONG+PYN+SUM $\rightarrow$ NONG is Agentive, PYN is Causation and SUM is Verb.

Some words are followed by single morpho-syntactic rules such as nominalization or causativization, words like *“Nongsum"* and *“Pynsum"*. Word *“Nongsum"* mean “one who bathes" and word *“Pynsum"* means “to cause/give a bath to somebody".

NONG+SUM $\rightarrow$ NONG is Agentive, and SUM is Verb.

PYN+SUM $\rightarrow$ PYN is Causation, and SUM is Verb.

Table 2, shows some examples of Khasi prefixed words used in generating morphological lexical rules.

3.4 Brill’s Transformation-based Learning (TBL)

Transformation-based learning (TBL) is an algorithm that uses the rule-based technique to categorize the part-of-speech classes and tag the given words automatically. The transformation-based Rule for POS tagging was first introduced by E. Brill [7, 8]. Using the Lexicon or Gold corpus, it will learn automatically the actual information that is linguistically hidden in sentences of the languages. This complex property maintains the flow of the grammatical class orders of sentences. This learned information helps the system to tag the words appropriately, according to their corresponding meanings.

In the present paper, analyzes accounted are based on the Transmission Rule-Based method of identification of POS tagging introduced by Brill. This model, at the initial stage, uses the un-annotated Corpus (or Raw data), which consists of input sentences that are not tagged or labeled. These sentences are then passed to the Initial State Annotator. Here, each word of the input text is annotated i.e. tags are assigned. To generate the possible tags for each word, this method makes use of a dictionary or lexicon (train data). The Temporary Corpus is the output that comes out from the Initial State Annotator. The Gold Corpus consists of the texts which are

Table 2. Some of Khasi Lexical morphology rules

Sl. No.	**Lexical morphology**	**Khasi Words**	**Meaning**	**POS Tags**
1	Nomilization	Jingpule Jingbatai	Studies Instructions	Jing->ABN
		Jinglum jingkhuid	Collections Cleanliness	jing->ABN
2	Causation	pynkulmar pynshai	To make trouble To make clear	Pyn->CAV
		pyntreikam pynkhuid	To make it work To make it clean	pyn->CAV
3	Agentive	Nongjop Nongthoh	Winner Writer	Nong->CMN
		Nongaïlam Nongrem	Leader Loser	nong->CMN

already tagged manually. This Khasi Lexicon is used to compare with the Temporary Corpus using some rules.

The tagger used two types of transformation processes which consist of rules: the Lexical and the Contextual Learner. In the Lexical learner module, the most frequent tag is used to assign for the known words, for the unknown words this module uses some system-generated rules to tag. Basically, in this architecture, the words are tagged by following or looking at the Khasi Lexicon or corpus. For particular words, that are not present in the corpus and the generated rules are unable to tag it, then that word is tagged as UNK (unknown).

In the Contextual learner module, Khasi Lexicon and some rules are used, to achieve high accuracy in reading and representation of the tagger. These rules are based on the flow of the context. As a word may belong to more than one tag, these rules act as a very important process, as they are used to identify the correct tags. These rules are automatically created by the Transformation-Based Error-Driven Learning (TEL). The rules are used to change the incorrect tags by applying the rules repeatedly.

Thus the Brill's transformation-based method uses some remarkable predefined set of rules as well as automatically generated rules that are learned by the system while training the data. Simple steps are given below which is followed in building the Khasi POS tagger:

— **Step 1.** Load the Corpus file.

— **Step 2.** Converting data from "word/tag" pair to a list containing tuple. i.e,[(word1,tag),(word2,tag)].

— **Step 3.** Set some predefined rules or lexical rules.

— **Step 4.** Feed the corpus data to BrillTagger-Trainer() for training data.

— **Step 5.** Evaluating and print the accuracy.

— **Step 6.** Printing the confusion matrix of standard tags with the test tags.

— **Step 7.** Calculate the Precision, Recall, F1-score.

In this experiment our **Input** to the system is the Khasi POS corpus (training data, validating data, test data) and raw data(Un-tagged words). The **Output** achieved from the system are Accuracy and Tagged words. Some of the predefine rules are as discussed in Sub-section 3.3 and also shown in Table 2. The achieved results whit the data are shown and discussed in Section 5.

4 Challenges on Corpus Building

In this section, we discuss some of the challenges, we met while building the corpus. These problems are (i) problems of spellings and orthography, and

(ii) problems of Ambiguity. As we have collected the raw data from online Khasi newspapers, we have found out that words which have the same meanings are spelled differently in different newspapers and in different articles. This is due to the reason that standardization of spellings and orthography of Khasi is still to be established. Khasi orthography has 23 alphabets and 6 vowels, the vowels are: “a e i ï o u". The Khasi alphabets are shown in Table 3.

Throughout our corpus building process, we found the same words are spelled differently by different writers. Such as, in some words, an alphabet is placed with *“ i "* instead of *“ ï "* and similarly, for the alphabets *“ n "* or *“ ñ "*. In the Khasi newspapers context, orthography problems are found to a large extent. Some such words are shown in Table 4.

Another technical problem that we have encountered is the presence of ambiguous words based on context. Some of such ambiguous words in Khasi are shown in Table 5, with their corresponding tags which are tag using the designed tagsets For instance, some words that are spelled alike but pronounced differently and have distinct meanings. The word like *"hadien"* which means *"after"* can be the Preposition or *"hadien"* which also mean *"behind"* can be a grammatical class for Adverb of Place.

Table 3. Khasi alphabet

a	b	k	d	e	g	ng	h	i	ï	j	l
m	n	ñ	o	p	r	s	t	u	w	y	

Table 4. Orthographic words in Khasi Language

Words	Orthography
ïing	ïing **or** ïing
ïathuh	ïathuh **or** iathuh
pynïoh	pynïoh **or** pynioh
Hynñiew	Hynñiew **or** Hynniew
Hynñiewtrep	Hynñiewtrep **or** Hynniewtrep
Rilum	Rilum **or** Ri lum
Kin	Kin **or** Ki yn

Table 5. Ambiguous words in Khasi Language

Khasi word	Meaning	Corresponding POS tags
hadien	after	IN
hadien	behind	ADP
shitom	Sick	CMN
shitom	tough/hard/difficult	ADJ
rong	color	CMN
rong	carry/blown away	TRV
mar	as soon as	AD
mar	material	MTN
mar	distribute	AD

Table 6. Most common words count

Sl.No.	Khasi words	Count in Khasi Corpus
1	ka	12461
2	ki	6888
3	ba	5286
4	u	4397
5	ïa	4124
6	la	2965
7	ban	2771
8	ha	2231
9	bad	1638
10	jong	1457
11	ngi	1006
12	na	793
13	dei	667
14	ruh	612
15	kane	598

5 Experimental Results

In the subsequent subsections, a brief discussion is presented based on the experimental work conducted on the Khasi lexicon or corpus along with the achieved results and its analysis.

5.1 Results

In this work, the method of Transformation-based learning (TBL) has been used to identify POS classes for the Khasi language. In comparison to the Khasi language, this is the first attempt to investigate POS tagging using the rule-based model with the designed Khasi POS corpus. The designed Khasi POS corpus consists of 4,580 sentences.

Table 6, presents the 15 most common words count in the corpus. We also present the

Table 7. Distribution of PoS Tags in the Designed Khasi corpus

Sl.No.	Tags	Desctription	Count in Khasi Corpus
1	QNT	Quantifiers	2056
2	CMN	Common nouns	9101
3	ADP	Adverb of Place	1048
4	COM	Complementizer	8381
5	VST	Verb, past tense	2945
6	ITV	Intransitive verb	1672
7	3PSF	3rd Person singular Feminine	12461
8	ABN	Abstract nouns	3166
9	FR	Foreign words	3516
10	VFT	Verb, future tense	4143
11	TRV	Transitive verb	2867
12	IN	Preposition	5134
13	3PPG	3rd Person plural common	6888
14	POP	Possessive Pronoun	1881
15	COC	Coordinating conjunction	1418
16	SYM	Symbols	4636
17	CO	Copula	5286
18	EM	Emphatic	712
19	AD	Adverb	4322
20	CRC	Correlative conjunction	304
21	DMP	Demonstrative Pronouns	1464
22	VSP	Verb, past perfective participle	203
23	ADD	Adverb of degree	591
24	NEG	Negation	2007
25	ON	Ordinal number	74
26	SUC	Subordinating conjunction	919
27	SPA	Superlative Adjective marker	752
28	CN	Cardinal Number	1127
29	RFP	Reflexive Pronouns	103
30	CAV	Causative Verb	1116
31	ADJ	Adjective	1575
32	ADF	Adverb of frequency	71
33	3PSM	3^{rd} Person singular Masculine gender	4397
34	PPN	Proper nouns	2713
35	MOD	Modalities	421
36	CLF	Classifier	369
37	ADT	Adverb of Time	921
38	DTV	Ditransitive verb	583
39	RLP	Relative Pronouns	93
40	CLN	Collective nouns	377
41	1PSG	1^{st} Person singular common gender	229
42	VPP	Verb, present progressive participle	539
43	3PSG	3^{rd} Person singular common Gender	199
44	PAV	Passive Voice	178
45	1PPG	1^{st} Person plural common gender	1009
46	CMA	Comparative Adjective marker	199
47	INP	Interrogative Pronouns	275
48	2PG	2^{nd} Person singular/plural common gender	145
49	MTN	Material nouns	114
50	DTV	Ditransitive verb	45
51	ADM	Adverb of Manner	7
52	2PF	2^{nd} Person singular/plural Feminine	7
53	2PM	2^{nd} Person singular/plural Masculine gender	5

distribution of tokens concerning POS-tags in the designed Khasi corpus along with its specifications as shown in Table 7. For more details regarding the tagsets, it can be found in [27].

From the designed Khasi corpus 84,972 manually tagged words are used as training data for the system, out of which 6,645 are the distinct Khasi words. For validating data 14,686 tagged words are used and 4340 words are used as testing data.

Using the validated data alongside the training data on the system an accuracy of 97.73% is yielded as performance. Due to the non-availability of lexicon or corpus, therefore we have created

Table 8. Validation result of our proposed Khasi POS tagger

Sl. No.	Khasi Lexicon Size	Rules Generated	Validation Accuracy
1	Training data (20,280 tokens) Validating data (5,323 tokens)	15005	86.79%
2	Training data (30,580 tokens) Validating data (5,323 tokens)	20102	88.84%
3	Training data (40,920 tokens) Validating data (5,323 tokens)	26017	91.83%
4	Training data (84,972 tokens) Validating data (14,686 tokens)	53,577	97.73%

Table 9. Validating result achieved using state-of-art and proposed TBL system for the Khasi Lexicon

Sl. No.	Khasi Corpus	Technique	Accuracy
1.	Training data (84,972 tokens) Validating data (14,686 tokens)	NLTK Bi-gram	88.88%
2.		NLTK Tri-gram	88.15%
3.		combining (Bi-gram+Trigram)	92.55%
4.		TBL **(Proposed work)**	97.73%

a Khasi lexicon or corpus. It is expected that the accuracy will increase further if more data are added to the corpus.

Table 8 represents the different validation results of the Transformation rule-based learning method for Khasi POS tagging using the designed Khasi lexicon. The result shows that as the data in the lexicon increase and fed to the model, the validation accuracy and generated rules are also increased.

Table 10. Some sample rules generated by the system

TBL Generated rules
11 11 0 0 | COM->None if Word:ba@[0] & Word:@[1] & Word:n@[2]
5 5 0 0 | COM->3PSF if Word:@[0] & Word:ka@[-1]
5 5 0 0 | 3PSF->None if Word:ka@[0] & Word:@[1]
4 5 1 0 | ITV->TRV if Word:bynrap@[0] & Word:shuh@[1] & Word:shuh@[2]
4 4 0 0 | TRV->ITV if Word:ong@[0] & Word:.@[1] & Word:“@[2]
4 4 0 0 | TRV->ITV if Word:nonghikai@[1,2,3]
4 4 0 0 | .->SYM if Word:.@[-3,-2,-1]
3 3 0 0 | 3PPG->None if Word:ki@[0] & Word:@[1] & Word:ba@[2]
3 3 0 0 | CMN->TRV if Word:pule@[0] & Word:-@[1] & Word:puthi@[2]

We have also compared the achieved results of the proposed TBL system with some state-of-the-art techniques shown in Table 9. From Table 9, we can observe that the TBL system has outperformed the state-of-the-art method.

Table 10 represents some sample sets of rules generated by the TBL system for Khasi which is generated from the trained Khasi lexicon. A comparison of our achieved result from the designed Khasi POS corpus using rule-based, with other related work on POS tagging that uses a rule-based approach is presented in Table 11.

Table 11 describes the corpus size used during the experiment, the different languages, and the accuracy achieved in percentage. From the table, we can observe that the proposed Khasi POS tagging which is experimented with the manually designed Khasi corpus give accurate result in comparison to other languages. We strongly suspect that more accuracy may be obtained, if more data are added to the training corpus.

The precision, recall, F-score are calculated from the confusion matrix so that the false positives, true positives and false negatives values can be accessed. The confusion matrix for training data with 40,920 tokens and validating data with 5,623 tokens is shown in Table 12.

Table 11. Comparison with some existing work on Khasi language for rule-based tagging

Sl. No.	Corpus	Generated Rules	Accuracy	Language References
1	Corpus data of around 85,159 tokens	-	90%	Arabic [15]
2	corpus consists of around 770k words	experiment 1 - 255 and experiment 2 - 1500	Training experiment 1 - 98.50% Training experiment 2 - 97.90% Testing experiment 1 - 96.9% Testing experiment 2 - 96.15%	Arabic [1]
3	Training data (26366 words) Testing data (6783 words)	-	96.28%	Sindhi [16]
4	corpus consists of 1000 words	For 100 words - 7 For 500 words - 15 and For 1000 words - 25	For 100 words - 50% For 500 words - 77% and For 1000 words - 85%	Manipuri [23]
5	Corpus of 26,149 words	-	87.55%	Hindi [12]
6	Corpus of 45,000 words	-	93.7%	Bangla [14]
7	KBBI	-	87.4%	Indonesian [19]
8	250,000 tokens	-	79%	Indonesian [20]
9	training data (84,972 tokens) validating data (14,686 tokens) testing data (4,340 tokens)	53528	validating $\rightarrow 97.73\%$ testing $\rightarrow 95.52\%$	Khasi (Our proposed KPOST)

In Table 12, Rows represent the actual values and column represent the predicted values. The true positive value can be identified as the intersection of row and column. Such as 1PSG in Row and 1PSG in the column the true positive value is 8.

From the table, the wrongly tagged result which is counted as false positive and false negative can be found. Such as, from the actual values, we can see there are 3 wrongly tagged outputs for 3PSF as 3PSM, which is counted as false positive for 3PSM. Similarly, there are 3 false negatives for 3PSF.

Again to have a close look at results we present the confusion matrix for training data with 84,972 tokens and validating data with 14,686 tokens in Table 13.

From the confusion matrix, one can calculate the recall, F-score, and precision of the validating data after training the model. The formula for calculating the recall, F-score, and precision can be express as below:

$$precision = \frac{TP}{TP + FP}, \tag{1}$$

$$recall = \frac{TP}{TP + FN}, \tag{2}$$

$$fscore = \frac{2 * (precision * recall)}{precision + recall}, \tag{3}$$

Table 12. Confusion matrix table of validating data 5,623 tokens when trained with 40,920 tokens

Tags	.	1PPG	1PSG	3PPG	3PSF	3PSG	3PSM	ABN	AD	ADD	ADJ	ADP	ADT	CAV	CLF	CLN	CMA	CMN	CN	CO	COC	COM	CRC	DMP	DTV	EM	FR	IN	ITV	MOD	MTN	NEG	ON	PAV	POP	PPN	QNT	RFP	RLP	SPA	SUC	SYM	TRV	VFT	VPP	VSP	VST	XX
.	324																																																	
1PPG		8																																																
1PSG			8																																															
3PPG				247																																														
3PSF					650		3											1																																	
3PSG						6	1																																										
3PSM					31		340											1																																	
ABN								145										2																																	
AD									1305	6								5																																	
ADD									5	36								2					3																											
ADJ								2	12	2	76	2						21																																	
ADP									1			13						3											1																					
ADT									1		1		48					3											1																					
CAV									1					75																																			
CLF															9	3																																	
CLN																22					3																												
CMA																	2																																
CMN							1		1		7	1						602		2				4		1		1																				
CN																			56										1																				
CO																				97																													
COC																					160							3																				
COM																						339																										
CRC																							2																									
DMP																		1						71																								
DTV																		1							15			1																				
EM																										39																							
FR																		78	1								2391																					
IN									9			10						1							1			434																				
ITV						1												6										48																				
MOD																													8																			
MTN																														0																		
NEG																															33																	
ON																																0																
PAV																																	4															
POP									1																									95														
PPN																		65		3				7											179													
QNT										1								2	2																	63												
RFP																																				6											
RLP																																			1	0										
SPA																																					20									
SUC									1									1		6																	25		1	1					
SYM																		1																				196								
TRV									8		2			1				28						2				26										1	2242							
VFT																																							144							
VPP									8																														1	9						
VSP																																								0						
VST																																	2								173				
XX																				1																							1				0	

where *TP is true positives*, *FP is false positives* and *FN is false negatives*.

We present Table 14, which shows the recall, f-score, and precision of each tag of validating data produced by the system when used the training data of 84,972 tokens and validating data of 14,686 tokens. Where we can observe the performance of the system as the data in the corpus is increased.

To have a good view of the system for the designed Khasi corpus we present the graphical view Figure 4 and 5. Figure 4, is the graphical representation of the F1-score of each tag using 40,920 tokens as training data in the proposed

Table 13. Confusion matrix table for validating data 14,686 tokens with 84,972 tokens as training data

Tags	.	1PPG	1PSG	2PG	3PPG	3PSF	3PSG	3PSM	ABN	AD	ADD	ADF	ADJ	ADP	ADT	CAV	CLF	CLN	CMA	CMN	CN	CO	COC	COM	CRC	DMP	DTV	EM	FR	IN	INP	ITV	MOD	NEG	ON	PAV	POP	PPN	QNT	RFP	RLP	SPA	SUC	SYM	TRV	VFT	VPP	VSP	VST	XX
.	623																																																	
1PPG	44																																																	
1PSG		12																																																
2PG			7																																															
3PPG				7306																																														
3PSF				22	1703	9												1										1																							
3PSG						7	1																																											
3PSM					31		902																																											
ABN								327										2																																
AD									3854	3								2																																
ADD									3	122								1						1																										
ADF										1																																							
ADJ									8	2291							5										1	1																						
ADP									5		56																	8																					
ADT									2	1		113																2																					
CAV													182																																				
CLF														32	2																																		
CLN															55			3																															
CMA																15																																	
CMN							1			1		6	3			1		1434		1						4	1																						
CN																		201																															
CO																			2791																														
COC									1											389																														
COM																			1	3	910																												
CRC																						23																											
DMP																					1		198																										
DTV																								29		1																						
EM																									87																							
FR																		14								1021																						
IN									1			2									2			2		1163																						
INP																							1		11																						
ITV						1						1															231																					
MOD																												26																				
NEG																											1		97																			
ON																														5																	
PAV																															14																
POP									2																							233															10	
PPN																		2								4							549														
QNT									1		1							1																146	1												
RFP																																		12													
RLP																																			2												
SPA																																				46											
SUC									1											10																	93										
SYM						1																				1												640								
TRV										4		1						4									6												1	6921							
VFT																				2									1										1	419	1					
VPP																																									1		30				
VSP																												1																3			
VST																													1		1													1	429	
XX																																															5

tagging system for Khasi. Figure 5, is the graphical representation of the F1-score of each tag using 84,972 tokens as training data in the proposed tagging system for Khasi.

From Figure 4 and 5 we can easily observe the tagging system it improve as we increased the numbers of data in the Khasi corpus.

5.2 Performance Analysis

To check the performance of the system, testing data is used. The testing data are the raw data which are un-tagged. These raw data are given to the system to tag them automatically by learning the rules and tags from the given Khasi lexicon

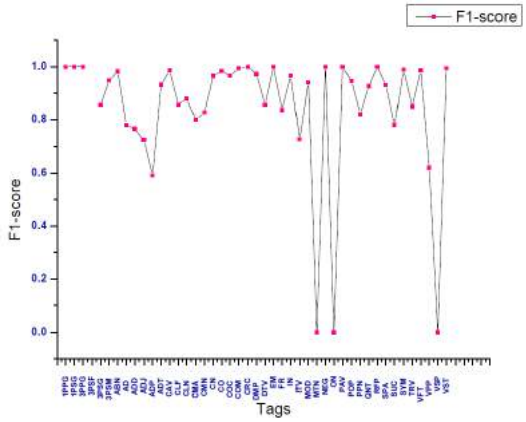


Fig. 4. Graphical representation of f1-score on individual POS tags using 40,920 tokens in the proposed system

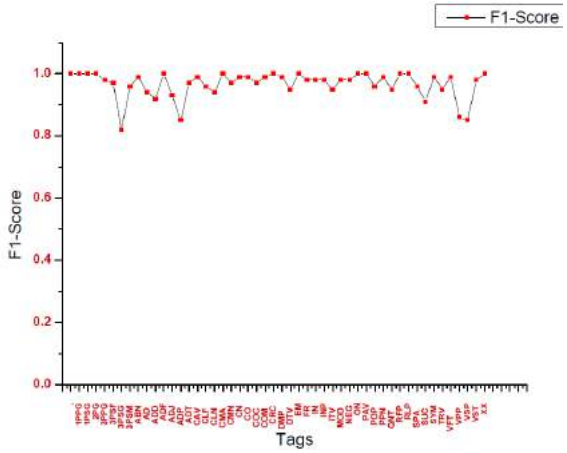


Fig. 5. Graphical representation of f1-score on individual POS tags using 84,972 tokens tokens in the proposed system

that act as gold corpus. As an output for the input raw Khasi data, the system produces the tags as an output for each word. Table 15 shows the sample resulting tags for particular words, which are given as input Khasi text. Table 15 shows the input raw Khasi text in the first row, and the second row represents its corresponding tags with its particular words. The output results are shown and verified by the Linguistic expert and found to be satisfactory.

The testing accuracy is shown in Table 16, using the proposed Transformation based learning (TBL) approach on the developed Khasi lexicon. We have checked the accuracy of our work in a phase-wise manner, where the Input data and output data are shown in Table 15. With the training data of 20,280 tokens and testing data of 970 tokens, the system yields an accuracy of 70.39%.

Whereas, with 30,580 tokens used for training and 1409 tokens used for testing, the accuracy of 79.60% is achieved, again for training data of 40,920 tokens and testing data of 3,122 tokens 90.97% accuracy is achieved. The accuracy increases gradually with the corpus size as shown in Table 16.

It is observed that 95.52% accuracy is achieved, in case of training data that consists of 84,972 tokens and testing data that consists of 6,643 tokens.

6 Discussion

This paper presents a brief discussion on the design Khasi POS corpus. Using the manually designed POS corpus several experiments are executed by dividing the corpus into small sections. The experiment shows an improvement in the performance regarding the identification of tags and it also increases in accuracy results, when the corpus size is increased.

This work is the first instance that uses the Transformation-based learning (TBL) approach for PoS tagging in the Khasi language. This work is indeed a resource work building for Khasi language towards the NLP perspective.

7 Conclusion and Future Works

In this paper, the transformation based method for POS tagging on the designed POS corpus for the Khasi language is presented. The designed Khasi corpus with some challenges and other related POS tagging issues of computing Khasi are also addressed in this paper. This paper presents the first application of Brill Tagger for POS in Khasi.

The accuracy results of validating data and testing data are shown in the tables and the different scores for validating data are also presented. We have also compared the accuracy

Table 14. The tags precision, recall and f-score for training data of 84,972 tokens and validating data of 14,686 tokens

Sl.no.	Tags	Precision	Recall	F1-score
1	.	1.0	1.0	1.0
2	1PPG	1.0	1.0	1.0
3	1PSG	1.0	1.0	1.0
4	2PG	1.0	1.0	1.0
5	3PPG	0.97	0.99	0.98
6	3PSF	0.97	0.98	0.97
7	3PSG	0.77	0.87	0.82
8	3PSM	0.98	0.95	0.96
9	ABN	1.0	0.99	0.99
10	AD	0.94	0.94	0.94
11	ADD	0.96	0.89	0.92
12	ADF	1.0	1.0	1.0
13	ADJ	0.93	0.93	0.93
14	ADP	0.90	0.81	0.85
15	ADT	1.0	0.94	0.97
16	CAV	0.99	1.0	0.99
17	CLF	1.0	0.94	0.96
18	CLN	0.94	0.945	0.94
19	CMA	1.0	1.0	1.0
20	CMN	0.97	0.97	0.97
21	CN	0.99	1.0	0.99
22	CO	0.99	0.99	0.99
23	COC	0.96	0.99	0.97
24	COM	0.98	0.99	0.99
25	CRC	1.0	1.0	1.0
26	DMP	0.99	0.99	0.99
27	DTV	0.93	0.96	0.95
28	EM	1.0	1.0	1.0
29	FR	0.99	0.98	0.98
30	IN	0.98	0.99	0.98
31	INP	1.0	0.91	0.95
32	ITV	0.96	0.84	0.90
33	MOD	0.96	1.0	0.98
34	NEG	0.97	0.98	0.989
35	ON	1.0	1.0	1.0
36	PAV	1.0	1.0	1.0
37	POP	0.97	0.95	0.96
38	PPN	0.99	0.98	0.99
39	QNT	0.93	0.97	0.95
40	RFP	1.0	1.0	1.0
41	RLP	1.0	1.0	1.0
42	SPA	0.93	1.0	0.96
43	SUC	0.93	0.89	0.91
44	SYM	0.99	0.99	0.99
45	TRV	0.93	0.97	0.95
46	VFT	0.99	0.98	0.99
47	VPP	0.78	0.96	0.86
48	VSP	1.0	0.75	0.85
49	VST	0.975	0.99	0.98
50	XX	1.0	1.0	1.0

Table 15. Input Text and the output Khasi text with the tags

Input: Khasi Text	'Ka jylla nalor ka long kaba la die kynrei ne pathar bha ïa ka kyïad, bluit wan ban mih paw pat sa ka jingïoh ban die pathar ki nongkhaïi sa ïa u drok, uba lehse ba long uwei na ki jait jingdih, uba ki khun samla kin hap ban long kiba peitngor ym tang ba ki ngat ïalade, hynrei ba kin shim khia naka bynta ban peitngor pat ïala ki para samla, ba kin nym shah shong kulai sa ha une uwei pat u jait jingdih pynbuaid uba ka pyrthei ruh ka dum buit ban tem ïaka'
Output	Ka/3PSF jylla/CMN nalor/ADD ka/3PSF long/CO ka/3PSF ba/COM la/VST die/TRV kynrei/AD ne/CRC pathar/ADJ bha/AD ïa/IN ka/3PSF kyïad/CMN (/SYM bluit/AD wan/ITV ba/COM n/VFT mih/ITV paw/AD pat/AD sa/VFT ka/3PSF jingïoh/ABN ba/COM n/VFT die/TRV pathar/AD ki/3PPG nongkhaïi/CMN sa/VFT ïa/IN u/3PSM drok/FR ,/SYM u/3PSM ba/COM lehse/AD ba/COM long/CO u/3PSM wei/CN na/IN ki/3PPG jait/ADJ jingdih/ABN ,/SYM u/3PSM ba/COM ki/3PPG khun/CMN samla/CMN ki/3PPG n/VFT hap/TRV ba/COM n/VFT long/CO ki/3PPG ba/COM peitngor/TRV ym/NEG tang/ADD ba/COM ki/3PPG ngat/TRV ïa/IN lade/RFP (/SYM hynrei/SUC ba/COM ki/3PPG n/VFT shim/TRV khia/AD na/IN ka/3PSF bynta/CMN ba/COM n/VFT peitngor/TRV pat/AD ïa/IN la/POP ki/3PPG para/CMN samla/CMN ,/SYM ba/COM ki/3PPG n/VFT nym/NEG shah/PAV shong/TRV kulai/CMN sa/VFT ha/IN une/DMP u/3PSM wei/CN pat/AD u/3PSM jait/ADJ jingdih/ABN pynbuaid/CAV ba/COM ka/3PSF pyrthei/CMN ruh/COC ka/3PSF dum/ADJ buit/CMN ba/COM n/VFT tem/TRV ïa/IN ka/3PSF

results of our work and other research work concerning rule-based pos tagging. From the comparison table, the proposed approach for Khasi POS tagging shown in this paper can be claimed that it generates more accurate POS tag sets, and produce higher accuracy result than the existing works.

The performance analysis of the output produced by the tagger are also discussed. Therefore, in the future, more tokens on the Khasi lexicon or corpus for POS tagging will be introduced. A thorough investigation of the corpus to account for the problems of ambiguities, and orthography encountered in this research will be addressed extensively.

This research paper will eventually contribute to the development of standard Khasi gold corpus.

Table 16. Testing result of Our proposed Khasi POS Tagger

Sl. No.	Khasi Lexicon Size	Rules Generated	Testing Accuracy
1	training data (30,580 tokens) testing data (1409 tokens)	20102	79.60%
2	training data (40,920 tokens) testing data (3122 tokens)	26017	90.97%
3	training data (84972 tokens) testing data (4340 tokens)	53528	95.52%

Acknowledgments

The authors would like to thanks the Government of India, Ministry of Science & Technology, Department of Science & Technology (DST), KIRAN Division, Technology Bhavan, New Delhi, for their supports and financial assistance (Grant: DST/WOS-B/2018/1216/ETD/Sunita(G)) during the study.

References

1. **AlGahtani, S., Black, W., McNaught, J. (2009).** Arabic part-of-speech tagging using transformation-based learning. Proceedings of the Second International Conference on Arabic Language Resources and Tools, 2001, MEDAR, pp. 66–70.
2. **Batra, K. K., Lehal, G. (2010).** Rule based machine translation of noun phrases from punjabi to english. International Journal of Computer Science Issues (IJCSI), Vol. 7, No. 5, pp. 409.
3. **Beard, R. (1987).** Lexical stock expansion. Rules and the Lexicon: Studies in Word Formation. Lublin: Redakcja Wydawnictw KUL, pp. 23–41.
4. **Bhatt, P. M., Ganatra, A. (2009).** Analyzing & enhancing accuracy of part of speech tagger with the usage of mixed approaches for gujarati. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277, Vol. 3878.
5. **Bredenkamp, A., Markantonatou, S., Sadler, L. (1996).** Lexical rules: what are they? Proceedings of the 16th conference on Computational linguistics-Volume 1, Association for Computational Linguistics, pp. 163–168.
6. **Brill, E. (1992).** A simple rule-based part of speech tagger. Proceedings of the third conference on Applied natural language processing, Association for Computational Linguistics, pp. 152–155.
7. **Brill, E. (1994).** Some advances in transformation-based part of speech tagging. arXiv preprint cmp-lg/9406010.
8. **Brill, E. (1995).** Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. Computational Linguistics, Vol. 21, No. 4, pp. 543–565.
9. **Btoush, M. H., Alarabeyyat, A., Olab, I. (2016).** Rule based approach for arabic part of speech tagging and name entity recognition. Int. J. Adv. Comput. Sci. Appl.(IJACSA), Vol. 7, No. 6, pp. 331–335.
10. **Chakrabarti, D., CDAC, P. (2011).** Layered parts of speech tagging for bangla. Language in India, www. languageinindia. com, Special Volume: Problems of Parsing in Indian Languages.
11. **Ekbal, A., Mondal, S., Bandyopadhyay, S. (2007).** Pos tagging using hmm and rule-based chunking. The Proceedings of SPSAL, Vol. 8, No. 1, pp. 25–28.
12. **Garg, N., Goyal, V., Preet, S. (2012).** Rule based hindi part of speech tagger. Proceedings of COLING 2012: Demonstration Papers, Association for Computational Linguistics, pp. 163–174.
13. **Hasan, F. M. (2006).** Comparison of different POS tagging techniques for some South Asian languages. Ph.D. thesis, BRAC University, Dhaka, Bangladesh.
14. **Hoque, M. N., Seddiqui, M. H. (2015).** Bangla parts-of-speech tagging using bangla stemmer and rule based analyzer. 2015 18th International Conference on Computer and Information Technology (ICCIT), IEEE, pp. 440–444.
15. **Khoja, S. (2001).** Apt: Arabic part-of-speech tagger. Proceedings of the Student Workshop at NAACL, NAACL, pp. 20–25.
16. **Mahar, J. A., Memon, G. Q. (2010).** Rule based part of speech tagging of sindhi language. 2010 International Conference on Signal Acquisition and Processing, IEEE, pp. 101–106.

17. **Mawphor (2017).** Mawphor. https://www.mawphor.com/index.php/. [Online; accessed Nov-2017 to June-2019].
18. **Nisheeth, J., Hemant, D., Iti, M. (2013).** Hmm based pos tagger for hindi. Proceeding of 2013 International Conference on Artificial Intelligence and Soft Computing, Springer, pp. 341–349.
19. **Purnamasari, K., Suwardi, I. (2018).** Rule-based part of speech tagger for indonesian language. IOP Conference Series: Materials Science and Engineering, volume 407, IOP Publishing, pp. 012151.
20. **Rashel, F., Luthfi, A., Dinakaramani, A., Manurung, R. (2014).** Building an indonesian rule-based part-of-speech tagger. 2014 International Conference on Asian Language Processing (IALP), IEEE, pp. 70–73.
21. **Schmid, H. (1999).** Improvements in part-of-speech tagging with an application to german. In Natural language processing using very large corpora. Springer, pp. 13–25.
22. **Selvam, M., Natarajan, A. (2009).** Improvement of rule based morphological analysis and pos tagging in tamil language via projection and induction techniques. International journal of computers, Vol. 3, No. 4, pp. 357–367.
23. **Singha, K. R., Purkayastha, B. S., Singha, K. D. (2012).** Part of speech tagging in manipuri: A rule based approach. International Journal of Computer Applications, Vol. 51, No. 14.
24. **Tham, M. J. (2018).** Challenges and issues in developing an annotated corpus and HMM POS tagger for Khasi. The 15th International Conference on Natural Language Processing, Association for Computational Linguistics, pp. 10–19.
25. **Toutanova, K., Klein, D., Manning, C. D., Singer, Y. (2003).** Feature-rich part-of-speech tagging with a cyclic dependency network. Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology-volume 1, Association for Computational Linguistics, pp. 173–180.
26. **Warjri, S. (2020).** Khasi-corpus. `https://github.com/sunitawarjri/Khasi-Corpus/blob/master/Khasi%20Corpus.txt`.
27. **Warjri, S., Pakray, P., Lyngdoh, S., Kumar Maji, A. (2018).** Khasi language as dominant part-of-speech (POS) ascendant in NLP. International Journal of Computational Intelligence & IoT, Vol. 1, No. 1.
28. **Warjri, S., Pakray, P., Lyngdoh, S., Maji, A. K. (2019).** Identification of POS tag for Khasi language based on Hidden Markov Model POS tagger. Computación y Sistemas, Vol. 23, No. 3.
29. **Warjri, S., Pakray, P., Lyngdoh, S., Maji, A. K. (2021).** Adopting conditional random field (CRF) for Khasi part-of-speech tagging (KPOST). Proceedings of the International Conference on Computing and Communication Systems, Springer, pp. 75–84.
30. **Warjri, S., Pakray, P., Lyngdoh, S. A., Maji, A. K. (2021).** Part-of-speech (POS) tagging using conditional random field (CRF) model for Khasi corpora. International Journal of Speech Technology, pp. 1–12.

Article received on 04/10/2021; accepted on 30/11/2021.
Corresponding author is Partha Pakray.

ISSN 2007-9737

Automatic Hate Speech Detection Using Deep Neural Networks and Word Embedding

Olumide Ebenezer Ojo[1], Thang-Hoang Ta[1,2], Alexander Gelbukh[1], Hiram Calvo[1], Grigori Sidorov[1], Olaronke Oluwayemisi Adebanji[1]

[1] Instituto Politécnico Nacional, Centro de Investigacion en Computación, Mexico

[2] Dalat University, Lam Dong, Vietnam

olumideoea@gmail.com, thangth@dlu.edu.vn, gelbukh@gelbukh.com, hcalvo@cic.ipn.mx, sidorov@cic.ipn.mx, olaronke.oluwayemisi@gmail.com

Abstract. Hatred spreading through the use of language on social media platforms and in online groups is becoming a well-known phenomenon. By comparing two text representations: bag of words (BoW) and pre-trained word embedding using GloVe, we used a binary classification approach to automatically process user contents to detect hate speech. The Naive Bayes Algorithm (NBA), Logistic Regression Model (LRM), Support Vector Machines (SVM), Random Forest Classifier (RFC) and the one-dimensional Convolutional Neural Networks (1D-CNN) are the models proposed. With a weighted macro-F1 score of 0.66 and a 0.90 accuracy, the performance of the 1D-CNN and GloVe embeddings was best among all the models.

Keywords. Hate speech, gloVe, 1D-CNN.

1 Introduction

The development of social media and other networking sites enables people to discuss and express themselves. This can be highly beneficial and can also pose tremendous danger to the peace of the society. Analysing sentiments in social media text is an important field of natural language processing and an integral part of many applications. Text from social media sites can contain many risks such as hate speech, fake news, violence, intimidation, racism and can also be life-threatening at times [23, 2, 11, 3, 1, 4, 17].

Individuals often share various opinions that can lead to unhealthy and unequal debate. Fostering a healthy dialogue is difficult for various social media sites, with these platforms being forced either to restrict or shut down users' comments. This study focuses on developing a model for detecting hate speech in English text on an internet forum site.

Humans discriminate against others based on their affiliations, classifying them as belonging or not belonging to a shared identity. The freedom to post online has prompted users to communicate differently, which can often lead to controversial outtakes on other individuals or on specific issues. Occupying different positions in politics and business is making communication play different roles in various events. Communication sites have made several efforts to moderate, but have restricted capacity. Needless to say, these moderators must put their time, resources and effort into managing their platforms against any form of negativity. The goal of this research is to detect hate speech in expressions in an efficient

and accurate manner, as well as to assist in the interpretation of text views.

Different machine and deep learning models was used to classify sentences and to access opinions in Ojo et.al [18]. This offers the edge to examine the perspectives of people on significant economic activities by studying their characters. It is really interesting to analyze these opinions from people about events and various issues as a way of knowing what they are thinking of, planning to do or engaged with at a particular time. In times like this, when decisions and responses are generated and modified in seconds, detecting hate speech in text is extremely necessary. The classification of text on social media platforms can be done using text classification tools [22, 12, 10, 15, 19].

In this paper, we used a deep learning approach to recognize different types of hatred in text and we focused on dataset from posts on a white supremacist forum, Stormfront [5]. We tried out different classification methods known as Naive Bayes Algorithm (NBA), Logistic Regression Model (LRM), Support Vector Machines (SVM), Random Forest Classifier (RFC) and the one-dimensional Convolutional Neural Networks (1D-CNN). The Bag-of-Words (BoW), term frequency–inverse document frequency (TF-IDF), and Global Vectors (GloVe) word embeddings were employed as feature representations.

The sequence classification approach is based on a neural network architecture that, through the representation of words and characters, benefits from the combination of word embeddings and 1D-CNN [3]. Machine learning approaches was also used to learn from the data and to perform the classification task. We used datasets tagged with Hate and No-Hate labels from [5] that was categorized and annotated at the sentence level. We were able to determine which classifier is best to detect hate speech based on the accuracy rate from the models.

2 Literature Review

2.1 General Concept

Deep learning analysis involves the rigorous study of data and requires systematic data analysis. During the analysis, deep connections are established between already existing concepts and new concepts are being developed, allowing long-term retention of ideas so that they can be used in new contexts to solve problems [7]. The architecture of a one dimensional Convolutional Neural Network Models (1D-CNN) and parameters represents a deep learning neural network.

Deep learning approaches like 1D-CNN, have recently been shown to achieve state-of-the-art performance on difficult classification problems [3]. Kernel slides along one dimension in 1D-CNN, which is mostly employed on text and 1D signals. 1D-CNN uses its internal state to process sequential data in order to memorize feature representations, perform classification and prediction tasks, and thus has no conceptual understanding of the data. It combines input vector with state vector to generate a new state vector with a learned function. It allows the measurement of fixed-size vector representations for arbitrary word sequences. We will implement a Convolutional Neural Network type algorithm called 1D-CNN for processing sequential data in this task.

2.2 State of the Arts

Several research works have been conducted using different methods to study and tackle the issue of hate, or toxicity detection in text [20, 9, 14, 1, 2, 5, 3, 4]. Various machine learning approaches, mostly defined by the type of network and training methods, have been used to classify text of this kind. According to [3], hate statements, otherwise known as violent threats, can be likened to a violent crime which affects the individuals or groups targeted. The researchers categorized the threatening comments into those that target an individual or group, and identified the threats of violence. They used a binary classification approach in their work to predict violence threats.

Convolutional Neural Networks (CNN) have also been applied to the text classification task for both distributed and discrete embedding of words [9, 22]. While representations derived from convolutional networks offer some sensitivity to word order, their order sensitivity is restricted to

mostly local patterns, and disregards the order of patterns that are far apart in the sequence. Although some word order sensitivity is given by representations derived from convolutional networks, there is limitation to their order sensitivity which ignores the order of patterns that are far apart in the sequence.

The use of 1D-CNN in [3] yielded better performance in the classification of text. In [18], multiple classifiers such as decision tree classifier, random forest classifier, vector supporting machines, logistic regression model, and others including a deep neural network, were used with n-grams approach to classify the text polarity.

In relation to this work, hate is another term being researched in the scientific community which results to bullying, unrest, embarrassment, and can even cause racism through the use of social media platforms.

Another research carried out on a dataset in [6] have also used the idea of transfer learning during their training process with respect to classification of text from various online communication channels. These networks maintain a state that can retain and reflect data from an indefinitely long text. The network stores several stable vectors in what is known to be memories, which the network remembers when similar vectors are presented to the network memory.

Word embeddings in [8] was used as a machine-learning method to depict each English word as a vector, with these vectors capturing semantic relationships between the associated words. The study looked at how the architecture of word embeddings varies over time and correlates with empirical demographic changes in terms of gender and ethnic stereotypes.

The use of hate speech [5], exist in many similar terms, which includes violent threat [3], offensive behavior [4], language of aggression or abuse [1, 2], and toxicity [14, 9, 20]. We are interested in distinguishing between text that is Hate Speech or not, and the method we proposed is our driving factor in carrying out this study. We will explain our approach in the following sections and provide information about the obtained results.

3 Experimental Analysis

3.1 Data

Gilbert et al. [5] generated a publicly available dataset annotated at the sentence level on Internet forum posts in English. The data were read and individually annotated into two different classes which are Hate and No-Hate. The texts were pre-processed and the embedded terms in text sequence were used as input into the models and the context summarized with a vector representation. We experimented with a number of representations, including BoW and word embeddings with GloVe, and concentrated on machine learning models and a deep-learning classifier. Each of the words in the sentence is then mapped to a (pre-trained) word embedding.

3.2 Methods of Analysis

Machine learning algorithms that were applied to the dataset include NBA, SVM, RFC and LRM. A one dimensional convolutional neural network, 1D-CNN, was also used to learn from the data. We used BoW for the machine learning algorithms and GloVe word embedding techniques for the 1D-CNN. Before performing hate detection classification on the binary label social media text, unnecessary features were removed to increase the efficiency and performance of the classification algorithms.

The data, which had already been labeled, was utilized to determine whether or not a specific text contained hate or no-hate assertions. 70% of the whole dataset was used for training, while the remaining 30% was used for testing. Word vector representations are extremely useful for capturing semantic information. Word2Vec [16] and GloVe [21] are the two most recently described methods for creating word embedding models.

GloVe is more efficient than Word2Vec, according to Pennington et al. [21]. GloVe means Global Vectors, with global referring to global corpus statistics and vectors referring to word representations. To obtain the inputs to the deep learning network, we used the GloVe pre-trained model.

Fig. 1. Confusion Matrix of the NBA Predictions on the Test Set

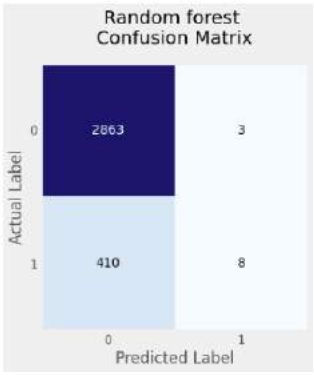


Fig. 2. Confusion Matrix of the RFC Predictions on the Test Set

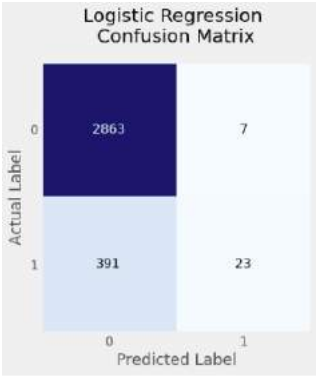


Fig. 3. Confusion Matrix of the LRM Predictions on the Test Set

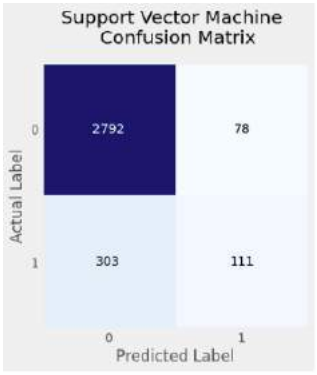


Fig. 4. Confusion Matrix of the SVM Predictions on the Test Set

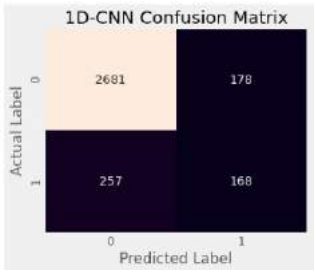


Fig. 5. Confusion Matrix of the 1D-CNN Predictions on the Test Set

A massive corpus of 2B tweets was used to train the GloVe pre-trained model. The machine learning algorithms used the BoW features. We performed optimization using the Adam algorithm [13]. The 1D-CNN learns to encode input sequence properties which are useful for the task of detecting hate speech in the sentence. CNN-based text classifications can learn features from words or phrases in different positions in the text.

4 Results

We classify the data with the models developed. After pre-processing the data, we used the BoW and TF-IDF approaches to convert text sentences into numeric vectors for the machine learning models. Four well-established machine learning algorithms were implemented on the datasets namely SVM, NBA, LRM and RFC. A deep

representation of the words and their relative meanings was done with Glove word embeddings and used by the 1D-CNN deep learning model.

Subsequently, the text was classified and the respective macro averaged F1 scores and accuracy results of all the models are shown in table 1 below. The proposed models for detecting and classifying hate speech in text were evaluated to identify the best algorithm.

Table 1. Accuracy and F1 values for all classification methods

Model	Features	F1	Accuracy
NBA	BoW	0.47	0.81
RFC	BoW	0.49	0.87
LRM	BoW	0.54	0.88
SVM	BoW	0.65	0.88
1D-CNN	GloVe	**0.66**	**0.90**

5 Conclusions

The difficulty of automatically recognizing hate speech in social media posts is addressed in this study. This research presents a hate speech dataset that was manually labeled and collected from a white supremacist online community. We discovered that the analysis generated significant hate preconceptions, as well as ranging levels of ethnic and religious-based stereotypes. Our findings have shown that the selection of word embeddings, the selected parameters and the optimizer have a high impact on the output achieved.

Hate speech in the social media space, which can have negative impacts on the society were detected easily and the high accuracy rate of the model will bring many benefits while reducing the damage. By assessing and comparing the performance of the various hate detection models, we found that word embeddings with 1D-CNN is an important tool for hate speech detection.

1D-CNN, a deep learning model, achieved the highest weighted macro-F1 score of 0.66 with a 0.90 accuracy. The results of the confusion matrix graphs in figures 1 to 5 demonstrated that GloVe embedding features were unable to correctly classify the test dataset. This could be due to the fewer training sentences used byy the GloVe word embedding algorithm. Furthermore, a closer examination of the figures reveal the order in which the models performed best in the dataset.

Acknowledgment

The work was done with partial support from the Mexican Government through the grant A1-S-47854 of the CONACYT, Mexico, and by the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico, under Grants 20211884, 20220859, and 20220553, EDI; and COFAA-IPN. We are grateful to Ona de Gibert Bonet and her colleagues for the dataset.

References

1. **Agrawal, S., Awekar, A. (2018).** Deep learning for detecting cyberbullying across multiple social media platforms. Proceedings of the European Conference in Information Retrieval (ECIR), Grenoble, France, pp. 141–153.
2. **Aroyehun, S. T., Gelbukh, A. (2018).** Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-1), Santa Fe, USA.
3. **Ashraf, N., Mustafa, R., Sidorov, G., Gelbukh, A. (2020).** Individual vs. group violent threats prediction in online discussions using deep learning. Companion Proceedings of the Web Conference 2020, April 20–24, 2020, Taipei, Taiwan, pp. 629–633.
4. **Chen, Y., Zho, Y., Zhu, S., Xu, H. (2012).** Detecting offensive language in social media to protect adolescent online safety. PASSAT 2012, International Conference on Social Computing (SocialCom), IEEE, Amsterdam, Netherlands, pp. 71–80.
5. **de Gibert, O., Perez, N., García-Pablos, A., Cuadros, M. (2018).** Hate speech dataset from a white supremacy forum. Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), Association for Computational Linguistics, Brussels, Belgium, pp. 11–20.

6. **Do, C. B., Ng, A. Y. (2005).** Transfer learning for text classification. Advances in Neural Information Processing Systems 18, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada.
7. **Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Rodr´ıguez, J. (2017).** A review on deep learning techniques applied to semantic segmentation. IEEE Transactions on Pattern Analysis & Machine Intelligence, Vol. 39, No. 04.
8. **Garg, N., Schiebinger, L., Jurafsky, D., Zou, J. (2017).** Word embeddings quantify 100 years of gender and ethnic stereotypes. CoRR, Vol. abs/1711.08412.
9. **Georgakopoulos, S. V., Tasoulis, S. K., Vrahatis, A. G., Plagianakos, V. P. (2019).** Convolutional neural networks for toxic comment classification. Proceedings of the 10th Hellenic Conference on Artificial Intelligence, SETN 2018, Patras, Greece, pp. 1–6.
10. **Graves, A., Schmidhuber, J. (2005).** Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks, Vol. 18, pp. 602–610.
11. **Hernández-Castañeda, A., Calvo, H., Gelbukh, A., García, F. J. (2017).** Cross-domain deception detection using support vector networks. Soft Computing, Vol. 21, No. 3, pp. 585–595.
12. **Juárez Gambino, O., Calvo, H. (2019).** Predicting emotional reactions to news articles in social networks. Computer Speech & Language, Vol. 58, pp. 280–303.
13. **Kingma, D. P., Ba, J. (2014).** Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.
14. **Kurita, K., Belova, A., Anastasopoulos, A. (2020).** Towards robust toxic content classification. EDSMLS 2020 (The AAAI-20 Workshop on Engineering Dependable and Secure Machine Learning Systems), New York City, United States.
15. **Majumder, N., Poria, S., Hazarika, D., Mihalcea, R., Gelbukh, A., Cambria, E. (2019).** Dialoguernn: An attentive rnn for emotion detection in conversations. Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, Honolulu, Hawaii, USA, pp. 6818–6825.
16. **Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013).** Efficient estimation of word representations in vector space. 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings.
17. **Mustafa, R. U., Ashraf, N., Ahmed, F. S., Ferzund, J., Shahzad, B., Gelbukh, A. (2020).** A multiclass depression detection in social media based on sentiment analysis. 17th International Conference on Information Technology–New Generations (ITNG 2020), Advances in Intelligent Systems and Computing, volume 1134, Las Vegas, Nevada, USA, pp. 659–662.
18. **Ojo, O. E., Gelbukh, A., Calvo, H., Adebanji, O. (2021).** Performance study of n-grams in the analysis of sentiments. Journal of the Nigerian Society of Physical Sciences, pp. 140–143.
19. **Ojo, O. E., Gelbukh, A., Calvo, H., Sidorov, G., Adebanji, O. (2020).** Sentiment analysis in texts on economic domain. Proceedings of the 19th Mexican International Conference on Artificial Intelligence - MICAI2020, Mexico City, Mexico.
20. **Ozoh, P. A., Adigun, A. A., Olayiwola, M. O. (2019).** Identification and classification of toxic comments on social media using machine learning techniques. International Journal of Research and Innovation in Applied Science (IJRIAS), Vol. IV, pp. 142–147.
21. **Pennington, J., Socher, R., Manning, C. (2014).** GloVe: Global vectors for word representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, pp. 1532–1543.
22. **Poria, S., Cambria, E., Gelbukh, A. (2016).** Aspect extraction for opinion mining with a deep convolutional neural network. Knowledge-Based System, Vol. 108, pp. 42–49.
23. **Sanguinetti, M., Poletto, F., Bosco, C., Patti, V., Stranisci, M. (2018).** An Italian Twitter corpus of hate speech against immigrants. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan.

Article received on 10/12/2021; accepted on 08/03/2022.
Corresponding author is Hiram Calvo.

ISSN 2007-9737

Arabic Text Mining for Price Prediction of Used Cars and Equipment

Belgacem Brahimi

Mohamed Boudiaf University of M'sila,
Laboratory of Informatics and its Applications of M'sila (LIAM),
Algeria

belkacem.brahimi@univ-msila.dz, belkacem.brahimi@yahoo.fr

Abstract. Nowadays, companies and businesspersons are increasingly interested in the web for its potential and opportunities in marketing and commercial activities. Despite the importance of Internet advertising of used goods available on the web, work targeting their analysis is still limited. It is crucial for both buyers and sellers to precisely estimate the price of used products available online. Textual information that describes second hand goods is very relevant for accurate price prediction, however common solutions use only structural features for price estimation. We study in this paper the utility of using text mining techniques as well as the role of textual data integration in improving price prediction of online classifieds in Arabic. In order to evaluate the proposed methods, we collected online advertisements for two cases: used cars and lots of construction equipment. Additionally, we applied prediction algorithms to estimate the prices, namely, regression-based algorithms, K– nearest neighbor and neural network. Experimental results showed that the integration of textual features in the prediction models improves significantly price prediction compared with baseline methods that use only structured features. The results proved also that regression models are the best option for price estimation.

Keywords. Text mining, supervised machine learning, regression, used cars prices, used equipment price, price prediction.

1 Introduction

Nowadays, the Internet world is growing considerably in terms of the number of users, websites and pages. Statistics in [1] show a dramatic evolution of the Internet from 1995 until the present day (from 16 million to 5,168 million users in 2021). This increase is due mainly to the democracy of the web and the low cost of publishing and consulting web content. Consequently, the web is becoming the first solution of social, E-commerce and marketing content comprising opinions and advertisements in different domains. In last recent years, there is an increasing interest of businesspersons, companies and users in analyzing and exploiting social media content. Indeed, the capabilities of such content are various; for example, in marketing, knowing users' opinions and attitudes is very useful for both customers and companies. Another relevant case concerns the prediction of economic and commercial indicators such as incomes and prices from online textual data. Owning such knowledge could help companies to still performing and competitive in the market.

However, the rapid evolution of the huge amount of web data poses a challenge regarding its effective exploitation in an optimal time and effort. Fortunately, in text mining, automatic predictive techniques are proposed to deal with such challenging tasks as they propose solutions to analyze data and predict required outcomes by computer programs. Due to the fact that the most typical form of online information is written words, text mining has a very high commercial potential. Indeed, a study showed that 80 % of a company's data was in textual form like emails and reports [2].

Despite the utility of employing predictive text mining techniques in social media content for business purposes, there has been limited work targeting this area. The situation in Arabic is even worse because, to the best of our knowledge, no

research effort has been devoted to the application of text mining methods for the task of prediction in business and commercial domains. Actually, the state-of-the-art of Arabic text mining confirms that most research efforts have been focused mainly on thematic text categorization [3-4-5-6-7-8], sentiment analysis [9-10-11-12-13], author attribution [14-15-16-17] and mining the holy Quran [18-19-20], while other proposed papers were interested in web pages clustering and annotation [21] and information extraction [22].

Recently, and in a business perspective, many companies specialized in marketing built their websites to provide online advertisements concerning several commercial activities. The domain of used cars and equipment is among the most interesting business sectors in Algeria because of their continuous growth and expansion in the last years. In fact, the decision of the Algerian government to ban the import of new cars has revolutionized the used cars and equipment markets. For example, the famous Algerian website Oued Kniss[1], specialized in posting advertisements concerning real estates, used equipment and cars, is the first Algerian website visited in Algeria, and ranked fourth among the first international websites visited in this country in 2021 [23]. This website was worth 40 billion Algerian dinars in 2014 with more than 800,000 advertisements [24].

In contrast of new cars advertisements in which attributes are categorical representing their components and options, the description of used cars and equipment by unstructured textual data is relevant for accurate price estimation. Indeed, some textual features including car description are very valuable, and thus should be taken in consideration in price prediction. For example, the state of a given car as nearly new or good as well as its components like engine, if new, repaired or revised, affects significantly the price of used car. Table 1 and 2 illustrate two examples of car and equipment advertisements (unit U equals 10,000 Algerian dinars). In these tables, we can see some relevant textual features (in bold) that could affect the price of used cars and equipment.

1 - www.ouedkinss.com

The goal of this research work is to explore the capabilities of using text mining techniques in Arabic, coupled with common predictive machine learning algorithms to estimate the prices of used goods like cars and equipment. In particular, we study the influence of the text preprocessing task on price valuation. We also investigate the impact of some data mining techniques such as feature selection on prediction results. In addition, we examine and compare the performance of some predictive algorithms like K–nearest neighbor, regression-based algorithms and neural networks in price forecasting of used cars and equipment. Finally, to evaluate the contribution of using text mining and integrating textual data in the prediction model, we compare the proposed methods with the same algorithms that employ only structured variables for price prediction.

We think that proposing such solutions for predicting accurate prices will be very helpful for both buyers and sellers. Indeed, providing a precise price estimation tool allows people to make the right decision of selling or buying used properties by avoiding overestimating or underestimating the real price [25].

The rest of the paper is organized as follows. In the next section, we give an overview on some studies related to the task of prediction in the commercial context by using text mining methods. Section 3 explains the process of gathering and preprocessing textual data and describes the solution for price valuation. After this, we present the results obtained with their interpretations in section 4. Finally, section 5 provides conclusions and possible improvements of the presented work.

2 Related Work

The aim of text mining is the analysis of large amounts of textual data and the detection of linguistic usage patterns to find useful information [26]. This research area uses natural language processing and data mining techniques to extract useful knowledge from texts.

In text mining, supervised methods are among the most popular techniques for mining such valuable information. In these methods, predictive models are built and learned, and then evaluated

Table 1. Sample announce of a used car with description

Car	
Model of the car	Chevrolet optra 4
Mileage in km	109,000 km
Year	2015
Description	لدي سيارة في حالة **ممتازة** فقط **خدش** صغير في الباب خلفي يمين. لا يوجد **دهن** . **محرك معاود و قوي جدا** . الباقي كل شيء **جيد**. (I have a car in **excellent condition** just a small **scratch** on the right rear door. There is no **paint**, **the engine is revised and very strong**. The rest is all **good**.)
Price of the car	140 U

Table 2. Sample announce of a lot of used equipment with description

Lot	
Lot ID	10234
Description	حصة تحتوي على شاحنة هيونداي ، حافلة ميني باص نوع كيا و سيارة بيك اب نوع تويوتا في **حالة جيدة**. (A lot contains a Hyundai truck, minibus Kia and a Toyota pick-up Toyota in **good condition**.)
Price of the lot	1410 U

Table 3. List of structured features in the car dataset

Feature name	**Type**	**Meaning**
Model	Text	Model of the car, example Chevrolet OPTRA
Mileage in km	Integer	Distance travelled by the car
Year	Integer	Year of manufacture
Price	Integer	The required price of the car

based on annotated examples with the aim of predicting the required results.

These forecasting techniques fall under two categories: classification and regression [27]. This distinction depends on the required type of prediction; in the classification task, an example is categorized in one of possible predefined classes, while regression models estimate the output of a given instance to a continuous numerical valuation [28].

In recent years, relevant research papers that investigated the task of predictive text mining models for marketing and business purposes have been proposed due to their importance for both customers and companies [25, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]. For example, in sentiment analysis context, suggested studies attempted to analyze the impact of consumers' sentiments on company's economic outputs [29], and to estimate revenue from opinions in Social Media [30]. The effect of news headlines [31] and sentiments on stock price estimation was also investigated in the work [32]. Moreover, some scientific articles studied the relation between purchase intention and product price [33, 34].

Other interesting studies exploited text mining approaches for the task of price prediction in markets. For example, the authors [35] presented a prediction method for crude oil prices. They indicated that their approach outperforms other

predictive methods. In a paper presented by [36], the researchers described a forecasting system based on text mining, called NewsCATS (News Categorization and Trading System) to predict trends in stock prices. Another interesting advertising activity in which price estimation is crucial is the real estate area. In this perspective, relevant scientific papers aimed to estimate the price of real estate classifieds [25, 37] and predict end-prices of online auctions [38] by the use of text mining methods.

Regarding Arabic text mining, research studies presented in the marketing and economic domain, that employ text mining techniques for price prediction are very rare. The present work aims to investigate this avenue of research.

3 Proposed Approach

3.1 Data Collection and Preparation

In this section, we explain the process of creating our datasets. Regarding the used cars domain, we collected Arabic advertisements from the first website of online advertising in Algeria (ouedkniss.com). The posts were gathered in the period between 10th August 2018 and 17th October 2018. We selected only texts in which web users employed standard or comprehensible Arabic in their online advertisements. Duplicate documents including identical descriptions of the second-hand cars were removed. Additionally, advertisements using Romanization of Arabic or different languages such as French were discarded from the corpus. Unrealistic announces requiring exaggerated prices were also rejected as they could affect negatively prediction model performance.

The obtained dataset contains 400 documents about the 20 most used models in Algeria. Each document in the data collection comprises two parts: the first part includes structured variables describing the used car like year of manufacture and mileage in km, while the second part of the document comprises textual data that describes the car state. The price in this dataset varies from 800,000 AD to 4.000,000 AD. AD means Algerian dinar. Table 3 shows a list of the structured features in the used car dataset.

To compile the second dataset related to used equipment of construction, we gathered examples from the same website (Ouedkniss). In addition, the same methodology of selecting documents in the first dataset of used cars is applied to create the second collection related to used equipment. In the second dataset, each document describes heavy equipment lots such as trucks and buses.

The obtained dataset comprises 482 texts, and the interval of price in this collection is between 1,000,000 AD and 28,000,000 AD.

Concerning data preparation, we performed usual text preprocessing techniques including tokenization, removing non Arabic letters and normalization of Alif – Taa and Yaa (ي -ة- ا-). In addition, stop words such as (على -في in English on-in), and words having length less than 2 letters were removed from the corpus.

In the preprocessing method, stemming is an optional task that reduces word forms to a unique representation (stem, base or root). In Arabic, widely known stemming methods are root stemming [39] and light-stemming [40]. In our work, we applied light stemming.

Regarding feature types, we used n-grams of words [41]. We recall that an n-gram of word is a contiguous sequence of n words from a given sample of text. An n-grams of size 1 is called unigrams; two successive words are called bigrams (or digrams) and size 3 is named trigrams.

The next step in the preprocessing task consists on assigning for each feature (word) a weight representing its relevance in the text. There are several weighting schemes such as Boolean weighting, Term Frequency TF and Term Frequency Inverse Document Frequency (TF.IDF) which is a combination of TF and Inverse Document Frequency (IDF). TF is the number of times a word occurs in a text, while IDF is the number of total documents over the number of documents containing word i. TF.IDF is a popular weighting scheme used in text mining applications such as information retrieval and text classification. This scheme reflects the relevance of a feature (word) in a given corpus.

Finally, we applied feature selection on the textual features to optimize the number of words in the dataset. We retained the most relevant features based on correlation between description

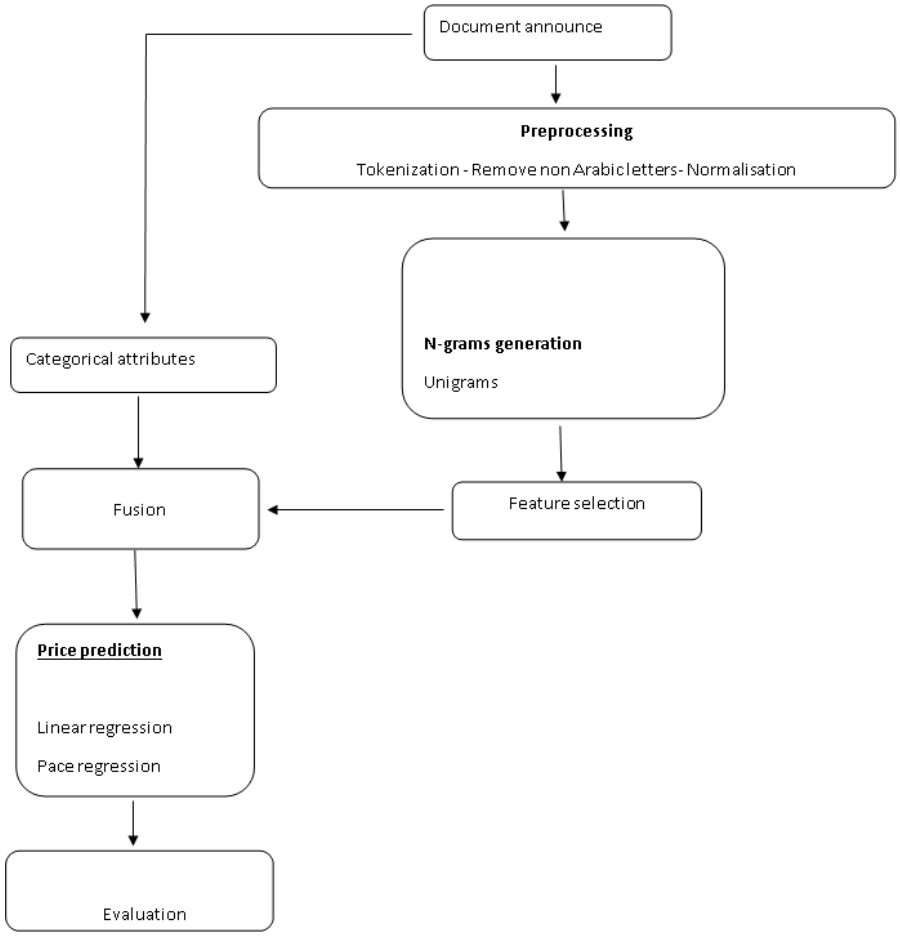


Fig. 1. Steps followed in our proposed approach

words and the output, i.e., the price. The optimal number of the most important features is determined empirically by experiments.

3.2 Used Algorithms

Regarding the price prediction model, and for comparison purpose, we applied four forecasting algorithms namely, linear regression (LinReg), pace regression (PaceReg), K-nearest neighbors (KNN) and finally Neural networks (NeuNet) for estimating the price of used cars and equipment. To determine the importance of textual information integration that includes the description of used cars in price valuation, we performed two prediction models. The first base model includes only categorical and structured data such as year of manufacture and mileage, while the second prediction model combines the first model (categorical features) with textual data that contains the description of the used car state. Figure 1 shows the different steps followed in the solution for integrating textual features to accurately predict the price of used cars and equipment.

3.3 Evaluation

To assess the performance of our predictive techniques, we calculated Root Mean Squared Error (RMSE), which is a well-known measure for evaluating prediction models. In addition, 10 fold-

Table 4. RMSE in the car dataset without textual data integration

Algorithms	RMSE
LineReg	32.791 ± 11.870
KNN (N=1)	73.631 ± 12.666
KNN (N=5)	59.165±12.021
KNN (N=10)	56.012±10.760
PaceReg	31.937±11.091
NeuNet	**30.539±10.770**

Table 5. RMSE in the car dataset with textual data integration

Algorithm	Feature size	RMSE	
		Unigrams	Bigrams
LineReg	10	27.839 ± 9.457	**27.585 ± 8.874**
	50	30.770 ± 8.798	31.414 ± 12.502
	100	30.476 ± 9.782	32.062 ± 13.055
KNN (N= 10)	10	56.106 ± 10.664	55.897 ± 10.887
	50	55.911 ± 10.881	**55.786 ± 10.728**
	100	55.907 ± 10.877	55.831 ± 10.791
PaceReg	10	**28.565 ± 8.567**	28.731 ± 8.370
	50	29.587 ± 8.996	31.228 ± 11.970
	100	32.230 ± 10.123	31.680 ± 9.079
NeuNet	10	29.136 ± 9.358	**28.526 ± 9.516**
	50	31.477 ± 9.223	29.081 ± 8.949
	100	32.397 ± 6.622	31.000 ± 8.243

cross validation was employed to average the performance results in the experiments.

Cross-validation is a popular technique used to evaluate and compare machine learning algorithms. For each iteration, we split data into two parts: one used to learn and build the prediction model, while the other part is utilized for the test step. The performance results of the 10 folds are then combined and averaged [42].

4 Experimental Results

The role of the experimental study is to evaluate and compare the performance of the proposed models for price forecasting. To perform experiments, we used the Rapid Miner software[2], which includes different tasks required to perform text preprocessing, prediction and evaluation.

Regarding the first text collection (car), we considered two scenarios. The first one is without considering textual data and using only structured features, while in the second setting, we integrated unstructured textual data representing the used car description in the prediction process.

Table 4 illustrates performance prediction results measured by (RMSE) for the four prediction algorithms without including textual information, that is the description of the car in the model. Through this table, the best result is obtained when the algorithm neural networks (NeuNet) is applied (30.539±10.770).

2 -http://rapid-i.com

Table 6. Example of some relevant words impacting the price of the car

Word	Weight	Word	Weight
جميلة (nice)	+22.551	قوة (power)	+13.684
جديدة (new)	+55.126	قوي (powerful)	+9.627
خدوش (scratches)	-4.900	نظيفة (clean)	+6.866
ضربة (choc)	-2.898	نظافة (cleanliness)	+11.451
حادث (accident)	-15.582	نقية (clean)	+20.740
مصبوغة (painted)	-27.879	صبغة (paint)	-15.582

Table 7. RMSE in the lots of equipment dataset

Algorithm	Feature size	RMSE	
		Unigrams	Bigrams
LineReg	10	457.488 ± 238.081	518.347 ± 231.582
	50	355.422 ± 262.844	428.134 ± 260.230
	100	**318.524 ± 274.326**	345.747 ± 278.843
KNN (N= 10)	10	466.195 ± 241.201	489.468 ± 243.697
	50	398.128 ± 263.475	452.539 ± 256.848
	100	415.493 ± 263.154	**395.590 ± 269.428**
PaceReg	10	459.948 ± 244.777	511.549 ± 233.568
	50	362.932 ± 263.316	429.844 ± 255.480
	100	**340.889 ± 268.593**	366.443 ± 271.005
NeuNet	10	524.814 ± 231.400	543.280 ± 235.249
	50	**368.699 ± 257.011**	430.134 ± 251.625
	100	443.707 ± 377.139	438.771 ± 321.049

In addition, KNN is the worst prediction algorithm, and it provides the best results when the number of neighbors equals 10.

Therefore, we continue to use this value for KNN in the next experiments.

In the second step, we integrated unstructured features including the description of the car in the prediction model. For this, we tested different text representation schemes in the experiments: binary, TF and TF*IDF. We found that TF*IDF is the best for all the prediction algorithms. Hence, we provide results concerning only this weighting scheme TF*DF for unigrams and bigrams of words. In addition, we performed pruning, which eliminates words that are correlated between them in order to optimise the list of relevant features in the final list of attributes. Table 5 depicts performance results of the four algorithms when employing unigrams and bigrams. The best results for each algorithm are highlighted in bold.

From the results provided in Table 5, we see that integrating textual information that contains the description of the used cars ameliorates price prediction for all the tested algorithms compared with the results of Table 4. We can also observe that the best algorithm for predicting car price is linear regression (LineReg).

Moreover, using bigrams enhances slightly the performance in three algorithms, and the optimal number of features equals 10 yielded the lowest values of RMSE. The first algorithm gained from the integration of unstructured textual data is LineReg as RMSE was reduced from 32.791 ± 11.870 to 27.585±8.874. This result agrees with

the findings of the paper [25], which confirmed that linear regression outperformed neural networks.

In order to go further in the analysis regarding the contribution of textual data integration for price estimation, we show in Table 6 a list of some relevant words that affect positively or negatively the prices of used cars.

We observe from Table 6 that some opinion words such as جديد- جميل (new and nice, in English) have a positive impact (weight) for increasing the price of the car, while some words related to the car description such as خدوش – حادث (accident and scratches in English), impact negatively its price. We see also that some words such as نقية – نظيفة (in English clean) share the same meaning. Therefore, we think that integrating a semantic approach that maps synonyms to their common concept could improve price prediction of used cars.

We continue our experiments with the second text collection of used equipment. The same preprocessing tasks were performed as in the first data collection of used cars. This data collection comprises only textual information (no categorical features).

The experimental outcomes are presented in Table 7 where the best results for each algorithm are highlighted in bold. The first remark is that the obtained results are modest when compared with the first dataset of used cars. These results go along with the conclusions of the study in [38] as the authors showed that their regression models did not provide good results for the prediction of end-prices of auction items. Our dataset is similar to the auctions dataset in [38] as it contains several heterogeneous items for sale.

According to Table 7, we see that, again, linear regression (LineReg) proves its superiority on the other algorithms in terms of RMSE, while KNN is the worst one. In addition, the optimal number of features is 100 in three algorithms. Regarding feature types, applying bigrams of words does not improve price prediction.

Finally, after carrying out our experiments on the two data collections, we have drawn some conclusions. Firstly, regarding the employed algorithms, the best ones using text mining and textual data for predicting the price are regression based models: linear and Pace regression as they provide the least RMSE values, while KNN is the worst model for price forecasting. In addition, the algorithm neural networks is applied with its defaults parameters; we think that optimizing these parameters could improve prediction results for this algorithm. Secondly, the integration of textual data that comprises the description of the car state ameliorates significantly price prediction results in the car dataset. Moreover, employing bigrams of words enhances price estimation in this corpus.

Concerning the datasets used in the experiments, the performance results of the second dataset that contains used equipment are lower than the car dataset. This is due to three factors. Firstly, each document in the equipment data collection comprises a description of many different equipment lots. In this case, the content of the lot is heterogeneous. Secondly, the description of the list of used items is not sufficient to describe them, in contrast of the car dataset, in which the description of the used car is provided in detail. Thirdly, the range of the price in the equipment dataset is larger than in the car corpus, and this obviously increases the price prediction error.

5. Conclusion and Future Work

In this paper, we studied the contribution of using text mining in Arabic for enhancing price prediction results for used cars and equipment. The idea behind this work is that textual information that describes a used good state is very relevant to precisely estimate its price. In particular, we used both types of features related to second-hand cars and equipment, namely, structured variables and textual data to makeprediction.

For this perspective, we compiled two datasets related to used cars and equipment. In addition, we employed and tested different predictive models, namely, K–nearest neighbors, regression based algorithms and neural networks to compare their performance results.

Experimental results proved that using text mining and considering textual data fusion in the prediction process improves price estimation. The results showed also that linear regression is the

most suitable model for the price prediction task. As future work, we intend to apply the proposed solution on other domains. We also think that deep learning techniques, information extraction and semantic approaches would be a solution for improving prediction results.

References

1. **Internetworldstats (2021).** https://www.internet worldstats.com/stats.htm.
2. **Tan, A.H., Ridge, K. (1999).** Text mining: The state of the art and the challenges. Proceedings of the PAKDD 1999 Workshop on Knowledge Discovery from Advanced Databases, Vol. 8, pp. 65–70.
3. **Khorsheed, M.S., Al-Thubaity, A.O. (2013).** Comparative evaluation of text classification techniques using a large diverse Arabic Dataset. Language Resources and Evaluation, Vol. 47, No. 2, pp. 513–538. DOI: 10.1007/s10579-013-9221-8.
4. **Al-Anzi, F.S., Abu-Zeina, D. (2017).** Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing. Journal of King Saud University-Computer and Information Sciences, Vol. 29, No. 2, pp. 189–195. DOI: 10.1016/j.jksuci.2016.04.001.
5. **Eldos, T.M. (2003).** Arabic text data mining: A root-based hierarchical indexing model. International Journal of Modelling and Simulation, Vol. 23, No. 3, pp. 158–166. DOI: 10.1080/02286203.2003.11442267.
6. **Nehar, A., Benmessaoud, A., Cherroun, H., Ziadi, D. (2014).** Subsequence kernels-based Arabic text classification. IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), pp. 206–213. DOI: 10.1109/AICCSA.2014.7073200.
7. **Atlam, E.S., Morita, K., Fuketa, M., Aoe, J. I. (2011).** A new approach for Arabic text classification using Arabic field- association terms. Journal of the American Society for Information Science and Technology, Vol. 62, No. 11, pp. 2266–2276. DOI: 10.1002/asi.21604.
8. **Zubi, Z.S. (2009).** Using some web content mining techniques for Arabic text classification. Proceedings of the 8th WSEAS International Conference on Data Networks, Communications, Computers, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society, pp. 73–84. DOI: 10.5555/1670344.1670357.
9. **Abdellaoui, H., Zrigui, M. (2018).** Using tweets and emojis to build TEAD: an Arabic dataset for sentiment analysis. Computación y Sistemas**,** Vol. 22, No. 3. DOI: 10.13053/CyS-22-3-3031
10. **Ahmad, K., Cheng, D., Almas, Y. (2007).** Multi-lingual sentiment analysis of financial news streams. International Workshop on Grid Technology for Financial Modeling and Simulation, SISSA Medialab, Vol. 26.
11. **Mulki, H., Haddad, H., Bechikh-Ali, C., Babaoğlu, I. (2018).** Tunisian dialect sentiment analysis: A Natural Language Processing-Based Approach. Computación y Sistemas, Vol. 22, No. 4. DOI: 10.13053/cys-22-4-3009.
12. **Aldayel, H.K., Azmi, A.M. (2016).** Arabic tweets sentiment analysis – A hybrid scheme. Journal of Information Science, Vol. 42, No. 6, pp. 782–797.
13. **Al-Sallab, A., Baly, R., Hajj, H., Shaban, K. B., El-Hajj, W., Badaro, G. (2017).** Aroma: A recursive deep learning model for opinion mining in Arabic as a low resource language. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), Vol. 16, No. 4, pp. 25.
14. **Alsmearat, K., Al-Ayyoub, M., Al-Shalabi, R., Kanaan, G. (2017).** Author gender identification from Arabic text. Journal of Information Security and Applications, Vol. 35, pp. 85–95.
15. **Abbasi, A., Chen, H. (2005).** Applying authorship analysis to Arabic web content. International Conference on Intelligence and Security Informatics, pp. 183–197, Springer, Berlin.
16. **Altheneyan, A.S., Menai, M.E.B. (2014).** Naïve Bayes classifiers for authorship attribution of Arabic texts. Journal of King

Saud University-Computer and Information Sciences, Vol. 26, No. 4, pp. 473–484.

17. **Alsmearat, K., Shehab, M., Al-Ayyoub, M., Al-Shalabi, R., Kanaan, G. (2015).** Emotion analysis of arabic articles and its impact on identifying the author's gender. IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), pp. 1–6.
18. **Sharaf, A. M. (2009).** The Qur'an annotation for text mining. Transfer report school of Computing, Leeds University.
19. **Muhammad, A.B. (2012).** Annotation of conceptual co-reference and text mining the Qur'an. University of Leeds.
20. **Alhawarat, M., Hegazi, M., Hilal, A. (2015).** Processing the text of the Holy Quran: A text mining study. International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 6, No. 2, pp. 262–267.
21. **Alghamdi, H.M., Selamat, A., Karim, N.S.A. (2014).** Arabic web pages clustering and annotation using semantic class features. Journal of King Saud University-Computer and Information Sciences, Vol. 26, No. 4, pp. 388–397.
22. **Harrag, F. (2014).** Text mining approach for knowledge extraction in Sahîh Al-Bukhari. Computers in Human Behavior, Vol. 30, pp. 558–566.
23. **Alexa (2021).** https://www.alexa.com/topsites/countries/DZ.
24. **Wikipedia (2018).** https://fr.wikipedia.org/wiki/Oued_Kniss.
25. **Abdallah, S., Khashan, D.A. (2016).** Using text mining to analyze real estate classifieds. International Conference on Advanced Intelligent Systems and Informatics, pp. 193–202, Springer, Cham.
26. **Sebastiani, F. (2002).** Machine learning in automated text categorization. ACM Computing Surveys (CSUR), Vol. 34, No. 1, pp. 1–47.
27. **Ye, N. (2013).** Data mining: Theories, algorithms, and examples. CRC press.
28. **Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. (1996).** Knowledge Discovery and Data Mining: Towards a Unifying Framework. In KDD, Vol. 96, pp. 82–88.
29. **Ghose, A., Ipeirotis, P.G. (2011).** Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No. 10, pp.1498–1512.
30. **Asur, S., Huberman, B.A. (2010).** Predicting the future with social media. Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01, pp. 492–499.
31. **Kirange, D.K., Deshmukh, R.R. (2016).** Sentiment Analysis of News Headlines for Stock Price Prediction. Composoft, an International Journal of Advanced Computer Technology, Vol. 5, No. 3, pp. 2080–2084.
32. **Oh, C., Sheng, O. (2011).** Investigating Predictive Power of Stock Micro Blog Sentiment in Forecasting Future Stock Price Directional Movement. In Icis, pp. 1–19.
33. **Grewal, D., Krishnan, R., Baker, J., Borin, N. (1998).** The effect of store name, brand name and price discounts on consumers' evaluations and purchase intentions. Journal of Retailing, Vol. 74, No. 3, pp. 331–352.
34. **Kwun, J. W., Oh, H. (2004).** Effects of brand, price, and risk on customers' value perceptions and behavioral intentions in the restaurant industry. Journal of Hospitality & Leisure Marketing, Vol. 11, No. 1, pp. 31–49.
35. **Yu, L., Wang, S., Lai, K.K. (2005).** A rough-set-refined text mining approach for crude oil market tendency forecasting. International Journal of Knowledge and Systems Sciences, Vol. 2, No. 1, pp. 33–46.
36. **Mittermayer, M.A. (2004).** Forecasting intraday stock price trends with text mining techniques. 37th Annual Hawaii International Conference on System Sciences, pp. 10.
37. **Stevens, D. (2014).** Predicting real estate price using text mining. Department of Communication and Information Sciences. Tilburg University.
38. **Ghani, R., Simmons, H. (2004).** Predicting the end-price of online auctions.

In International workshop on data mining and adaptive modelling methods for economics and management.

39. **Khoja, S., Garside, R. (1999).** Stemming Arabic text. Computing Department, Lancaster University, UK.
40. **Larkey, L.S., Ballesteros, L., Connell, M. E. (2007).** Light stemming for Arabic information retrieval. Arabic Computational Morphology, pp. 221–243.
41. **Shannon, C.E. (1948).** A mathematical theory of communication. Bell System Technical Journal, Vol. 27, No. 3, pp. 379–423.
42. **Manning, C.D., Schütze, H. (1999).** Foundations of Statistical Natural Language Processing. MIT press.

Article received on 05/09/2021; accepted on 05/12/2021.
Corresponding author is Belgacem Brahimi.

ISSN 2007-9737

Measuring the Storing Capacity of Hyperdimensional Binary Vectors

Job Isaías Quiroz Mercado, Ricardo Barrón Fernández, Marco Antonio Ramírez Salinas

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

jobquiroz@hotmail.com, barron2131@gmail.com, mars@cic.ipn.mx

Abstract. Hyperdimensional computing is a model of computation based on the properties of high-dimensional vectors. It combines characteristics from artificial neural networks and symbolic computing. The area where hyperdimensional computing can be applied is natural language processing, where vector representations are already present in the form of word embedding models. However, hyperdimensional computing encodes information differently, its representations can include the distributional information of a word in a given context and it can also account for its semantic features. In this work, we investigate the storing capacity of hyperdimensional binary vectors. We present two different configurations in which semantic features can be encoded and measure how many can be stored, and later retrieved, within a single vector. The results presented in this work lay the foundation to develop a concept representation model with hyperdimensional computation.

Keywords. Hyperdimensional computing, vector symbolic architectures, reduced representations.

1 Introduction

Hyperdimensional computing is a new model of computation based on the properties of high-dimensional vectors [1]. Hyperdimensional computing takes ideas from artificial neural networks, because it performs distributed processing, and it is also inspired on symbolic computing because complex structures, such as hierarchical trees or sequences, can be formed by manipulating symbols (vectors) representing simpler objects.

The set of architectures working under the hyperdimensional computing principles is known as Vector Symbolic Architectures (VSAs) [2]. In artificial neural networks, the use of vectors comes implicitly with the architecture description, in contrast, high-dimensional vectors in VSAs are not only part of the architecture but are the basic computing entities itself [1]. VSAs have been increasing in popularity during the last years; they have been applied in cognitive architectures [3], in analogy-based reasoning [4], to represent sequences and hierarchical structures [5, 6], and in pattern recognition [7].

Hyperdimensional computing has also been used in natural language processing (NLP). The Random Indexing technique [8], for example, uses high-dimensional vectors to create vector representations for texts.

This technique has a similar approach to vector semantic models, the current state-of-the-art models in most NLP applications. These models, commonly known as *word embeddings*, provide vector representation for words, paragraphs and entire documents, and they are based on the distributional hypothesis of meaning [9], which states that words with similar meaning tend to occur in similar contexts.

We propose a method for representing concepts using hyperdimensional computing principles, which are based on knowledge rather than in the distributional information of a word. These representations are created from a list of semantic features [10] encoded within a single high-dimensional vector. This work focuses on measuring the limit number of semantic features that a high-dimensional binary vector can successfully store. Our experimental results will be used for selecting the appropriate dimensionality of vectors within a

Table 1. Properties of arithmetic operations

Operation	Symbol	Properties
Addition (bundling)	+	- n-ary function, - Combines a set of vectors, - Elementwise majority with ties broken at random, - Resultant vector is similar to argument vectors.
Multiplication (binding)	⊗	- 2-ary function, - Combination of a pair of vectors, - Elementwise exclusive-or, - Invertible operation, - Distributes over addition, - Resultant vector is dissimilar to vectors being multiplied.

hyperdimensional computing system still in development.

The rest of the paper is organized as follows. Section 2 explains the general properties of hyperdimensional computing and describes how to encode semantic features into high-dimensional vectors. In Section 3 we present the experimental results that are later discussed in Section 4. Finally, Section 5 draws the conclusions of the work.

2 Methods

The most distinctive property of high-dimensional spaces (i.e. N > 1,000) is the *tendency to orthogonality*. This means that most of the space is *nearly* orthogonal to any given point [11]. For instance, if two random binary vectors are generated, it is highly probable that the Hamming distance between them is approximately N/2. As a consequence of this, arithmetic operations between this type of vectors yields to a new way for encoding information. In this section, we describe the basic arithmetic operators for high-dimensional vectors: addition and multiplication, and how they can be used to encode a list of semantic features within a single high-dimensional vector to represent a concept.

2.1 Arithmetic Operations

In general, there are two basic operations in hyperdimensional computing: binding (or multiplication) and bundling (or addition). These operations are used to encode, map and retrieve information. In this work we implement a framework called Binary Spatter Codes (BSC) [12], which uses binary vectors, see Table 1.

To represent higher level objects through arithmetic operations, first a set of *primitive* objects have to be defined and be associated with randomly generated vectors. The simplest method to combine a set of *primitive* objects is by bundling them together through addition. For instance, to create the vector class *Animals* the vector from each animal specie in the system can be added together, (1):

$$Animals = Dog + Cat + Birds + \ldots \quad . \tag{1}$$

While this method might be useful for some small systems, it does not allow to encode more complex relations as other methods. Gallant & Okaywe [5] presented another method for encoding a single sentence where, rather than bundling all vectors together, the subject, verb and object are multiplied by a *label vector*, after which all the vectors are added. For instance, the sentence “Mary loves pizza” will be encoded as in (2):

$$S = subj \otimes Mary + vrb \otimes loves + obj \otimes pizza \, . \tag{2}$$

Each vector added can be a randomly generated vector, or be another encoded sentence itself. Unlike the previous method, the order is encoded within the final vector, and therefore, the vectors produced for “Mary loves pizza” and “Pizza loves Mary” will be different.

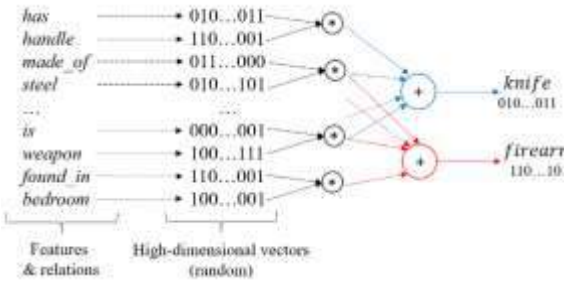


Fig. 1. Encoding concept's definition based on its semantic features. Two concept vectors are geometrically closer if they share the same semantic features

Table 2. List of semantic features for the concept *knife*

Feature	Classification
has a handle	Visual-form and surface
made of steel	Visual-form and surface
is shiny	Visual-form and surface
used for cutting	Function
used for killing	Function
is sharp	Tactile
is dangerous	Encyclopedic
found in kitchens	Encyclopedic
is a weapon	Taxonomic
is a utensil	Taxonomic

2.2 Encoding Semantic Features

Semantic features represent the basic conceptual components of meaning for a word [10]; they try to establish the meaning of a word in terms of its relationships with other words. Semantic features must be obtained from humans, either directly, in studies where humans are tasked with enumerating properties from a given concept, or indirectly, by taking information from knowledge bases. They incorporate different type of information, including both perceptual (e.g., shape and color), and non-perceptual attributes (e.g. taxonomic and functional). Table 2 gives an example of the semantic features in the McRae dataset [10] for the concept *knife*.

Within a semantic feature it is possible to identify two different parts: a relation (e.g. *has*, *is*, *used_for*) and a feature value (e.g. *handle, weapon*, *cutting*). From this observation, we propose to construct a vector representation as shown in equation (3):

$$Concept = \sum_{i=1}^{n} Relation_i \otimes Feature_i \,. \tag{3}$$

Vectors representing relations are called *relation* vectors. They represent the main semantic relations between two different words. Vectors representing features are called *feature* vectors. Both feature and relation vectors can be selected at random, or be an encoded vector itself. For instance, the *knife* concept will be encoded as (4):

$$knife = has \otimes handle + made_{of} \otimes steel + is \otimes utensil. \tag{4}$$

The hypothesis behind this encoding method is that another vector with a similar set of semantic features will be close to the original vector, Fig.1. In the case of the BSC the metric use to measure distance between two vectors is Hamming distance.

An interesting property of the multiplication operation is its invertibility. This means that it is possible to extract back a previously multiplied vector. For example, in (5) the vector Z is the multiplication between X and Y, however, since the XOR operation is its own inverse, it is possible to obtain X back by multiplying Z by Y (6):

$$Z = X \otimes Y, \tag{5}$$

$$Z \otimes Y = (X \otimes Y) \otimes Y,$$

$$X \otimes Y \otimes Y \,, \tag{6}$$

$$X \otimes 0 \,,$$

$$X.$$

As a consequence of the invertibility property, the representations created by the proposed method are *interpretable*, that is to say, once the final representation of a concept is created it is possible to analyze what features were included within the representation. For example, when the *knife* vector produced in (4) is multiplied by the relation vector *has*, the resultant vector would

Table 3. Precision for retrieving features

n_{pairs}	N = 2,000	N = 5,000	N = 7,000	N = 10,000
20	1.0	1.0	1.0	1.0
30	1.0	1.0	1.0	1.0
40	0.88	0.97	0.98	1.0
50	0.68	0.84	0.84	0.90
60	0.58	0.51	0.53	0.75
70	0.35	0.31	0.31	0.32
80	0.32	0.20	0.13	0.12

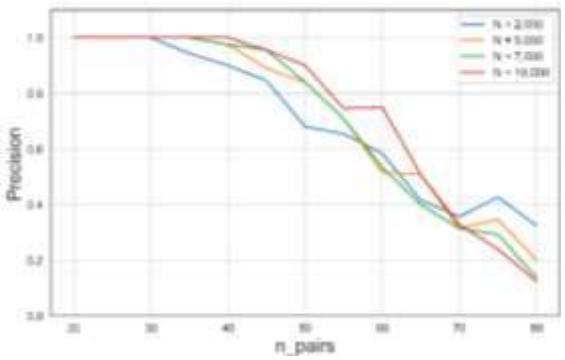


Fig. 2. Precision for the retrieving of features from high-dimensional vectors

include the vector *handle* plus an additional noise vector:

$$knife \otimes has = has \otimes handle \otimes has + \cdots + is \otimes utensil \otimes has \; handle + \cdots + is \otimes utensil \otimes has \; handle + noise \; . \quad (7)$$

This *noise* vector is the addition of all the other relation-feature pairs that do not have *has* as a relation vector. Due to the properties of the multiplication operation, this noise vector will be nearly orthogonal to *handle* and can be eliminated through an associative memory, an operation called *clean-up*. For more details and examples of the hyperdimensional computing operations and the use of autoassociative memories in VSAs refer to [1, 4, 5].

3 Experimental Results

In this section, we present the results of two experiments performed to test the maximum storing capacity of high-dimensional binary vectors. Each experiment was focused in a specific semantic features configuration. In each case, we quantified the storing capacity of hyperdimensional binary vectors for different vector sizes. The code for all the experiments is publicly available repository[1].

3.1 One Relation – One Feature Configuration

In this first experiment, we took a simple semantic feature configuration where each relation is associated with a single feature.

While this configuration is not common to find in knowledge bases, mainly because there are always more features than relations, it is the configuration storing the largest number of orthogonal vectors.

The parameters for this experiment were: **N**, the dimensionality of the vectors, and $\mathbf{n_{pairs}}$, the number of relation-feature pairs to encode. The encoded vectors have the form:

$$C = R_1 \otimes f_1 + R_2 \otimes f_2 + \cdots + R_{n_{pairs}} \otimes f_{n_{pairs}} \; . \quad (8)$$

Each relation and feature vectors (R_i and f_i, respectively) was randomly generated and paired up with another vector to create the concept vector C. After this, multiplications ($C \otimes R_i$) and *clean-up* operations were performed to extract back each feature (f_i).

Table 2 shows the precision, the relation between the number of encoded and retrieved features, for different **N** and $\mathbf{n_{pairs}}$ values. Fig. 2. illustrates these results.

The results presented indicate that the storing capacity of high-dimensional binary vectors do not increase in a linear fashion. As the dimensionality *N* increases the storing capacity increases but not in the same proportion.

For instance, the maximum number of relation-feature pairs that a 2,000-size vector can store is 30 pairs, by increasing the size to 10,000 the capacity increases to 40 pairs.

Given that the intended application for this method is to represent concept's definitions, 40 pairs is enough to describe the most important semantic features for a concept.

[1] https://github.com/jobquiroz/StoringCapacity_HDC

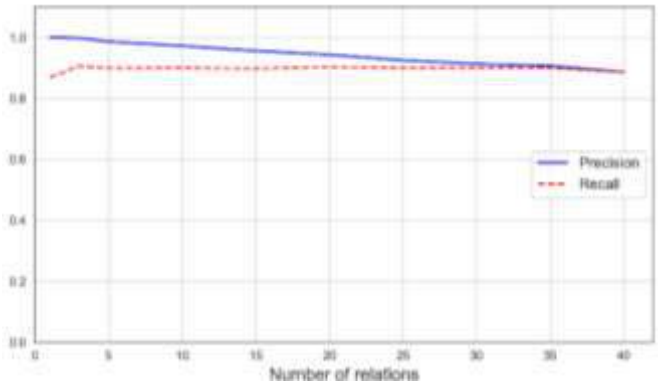


Fig. 3. Precision and recall for N = 2,000 and storing 40 features in *one relation – multiple features* configuration

Table 4. Precision and recall for feature retrieving using *one relation-multiple features* configuration ($\mathbf{n_{feat}}$ = 40)

n_rels	N = 2,000		N = 5,000		N = 7,000		N = 10,000	
	P	R	P	R	P	R	P	R
1	1.0	0.87	1.0	0.97	1.0	0.95	1.0	0.99
5	0.99	0.89	0.99	0.98	1.0	0.99	1.0	0.99
10	0.97	0.90	0.99	0.98	0.99	0.99	0.99	0.99
20	0.94	0.91	0.98	0.97	0.99	0.99	0.99	0.99
40	0.88	0.88	0.97	0.97	0.98	0.98	1.0	1.0

3.1 One Relation – Multiple Features Configuration

In this experiment, we focus on a more general configuration, where each relation could be associated with more than one feature. This configuration is more common to find in knowledge bases. For instance, in the *knife* concept from Table 1 the relation *is* was associated with five different features.

The parameters for this experiment were: **N**, the dimensionality of the vectors, $\mathbf{n_{rel}}$, the number of relations, and $\mathbf{n_{feat}}$ the total number of features to encode.

Unlike in the previous configuration, there are several ways to combine the set of relations with the set of features. This seems to lead to performing a full combinatorial analysis. However, since this experiment was focused on measuring the storing capacity of vectors, we care about the *number* of features associated to each relation rather than *which* features are with each relation. This assumption reduces the number of possibilities to analyze. Equation (8) express the possible ways to encode a concept with $\mathbf{n_{rel}}$ relations and $\mathbf{n_{feat}}$ features:

$$C = \sum_{i=1}^{n_{rel}} R_i \otimes \left[\sum_{k=1}^{m_i} f_k^i\right], \quad where \sum_i^{n_{rel}} m_i = n_{feat}\ with\ m_i > 0\ . \tag{9}$$

A configuration example for $\mathbf{n_{rel}}$ = 3 and $\mathbf{n_{feat}}$ = 6 is shown in (9).

$$C = R_1 \otimes [f_1^1] + R_2 \otimes [f_1^2 + f_2^2 + f_3^2] + R_3 \otimes [f_1^3 + f_2^3]. \tag{10}$$

To simplify the analysis, in this experiment we set a fixed number of features, $\mathbf{n_{feat}}$ = 40, and iterate over different $\mathbf{n_{feat}}$ and dimensionality values, Table 4. The reason for this is that, according to the previous experiment, at $\mathbf{n_{feat}}$ = 40 the precision for the feature retrievals do not reach 1 for most of the vector sizes tested. By leaving $\mathbf{n_{feat}}$ fixed we can observe how the rearrangement between features and relations change the precision of the retrievals.

Table 4. Precision and recall for feature retrieving using *one relation-multiple features* configuration ($\mathbf{n_{feat}}$ = 40)

	One relation – one feature	**One relation – multiple features**
Advantage	Straightforward retrieving process.	Less space needed within the definition vector
Disadvantage	More space needed within the definition vector	Retrieving can be ambiguous
Example	$Apple = is \otimes fruit + shape \otimes round + flavor \otimes tasty + has \otimes skin$ $is \otimes Apple = fruit$ (8 vectors encoded)	$Apple = is \otimes fruit + is \otimes round + is \otimes tasty + has \otimes skin$ $is \otimes Apple = [fruit, round, tasty]$ (6 vectors encoded)

In this experiment, we include the recall value, which measures the total amount of encoded features f_k^i that were actually retrieved. The recall value for all the measures in the previous experiment was the same than the precision value, meaning that when a feature was retrieved it was always the encoded feature; in this experiment this is not the case as Fig. 3 shows.

The *one relation – multiple features* configuration reorganizes the information by distributing the features among a lower number of relations. This configuration resembles more how concepts are commonly described in knowledge bases like ConceptNet [13]. Unlike in the previous experiment, it was necessary to measure the recall value of the retrieval operations because in some cases the list of retrieved features did not match the list of encoded features. This was especially problematic in lowest-dimension tested (N = 2,000).

4 Discussion

The goal of the present article was to measure the storing capacity of high-dimensional binary vectors following the Binary Spatter Codes framework.

Our experimental results showed that the relation between the increase in the size of the vectors do not maintain a linear relation with the total amount of items encoded.

Unlike other vector representations where each component stores specific information, the representations described in this article distribute the information across all components (holistic processing [1]).

The presented results also indicate that the determining factor in the overall storing capacity of the vectors is not the configuration used for encoding, but the total number of orthogonal vectors stored.

However, the configuration used dictates how the vectors are going to be retrieved. In the first configuration, after the inverse multiplication and the *clean-up* operations are performed, only one feature vector is obtained, while in the second configuration the final output is a list of features.

Table 4 summarizes the advantage and disadvantage of each configuration according to our experimental results.

As the example in Table 4 shows there are general relations (‘is’) that can be substituted by more specific relations (‘shape’, ‘flavor’) to make the retrieving less ambiguous.

However, adding more relations implies using more space. In the case of the goal application for this method, it should be noted that in knowledge bases like ConceptNet [13] and the McRae dataset [10], most concepts are characterized by less than 40 semantic features.

Based on our findings, we propose N = 10,000 as an appropriate vector size for representing concepts based on its semantic features. Vector sizes of 5,000 and 7,000 can also be worth considering if the processing speed is a constraint. In that case, the maximum number of semantic features to encode must be reduced accordingly.

4 Conclusions

This work presented an empirical exploration of the storing capacity of binary vectors using a VSA framework. We presented some aspects of hyperdimensional computing, a model of computation based on the manipulation of high dimensional vectors, and proposed a method for representing vectors based on a list of its semantic features.

We presented experimental results for encoding and then retrieving semantic features under two types of configurations: *one relation – one feature*, and *one relation – many features*. We identify the main advantage and disadvantage of each configuration and selected the 10,000-size vectors as most appropriate for representing concepts. This result will be later used to further develop this encoding method.

This works lays the foundation from a representation model intending to encode larger knowledge bases, like ConceptNet, for modeling language using hyperdimensional computing.

Acknowledgments

This work has been funded by SIP-IPN under grant SIP-20201415 and by CONACYT scholarship number 666415.

References

1. **Kanerva, P. (2009).** Hyperdimensional computing: An introduction to computing in distributed representation with high dimensional random vectors. Cognitive Computation, Vol. 1, No. 2, pp. 139–159.
2. **Gayler, R. (2003).** Vector Symbolic Architectures answer Jackendoff's challenge for cognitive neuroscience. ICCS/ASCS International Conference on Cognitive Science. Sydney Australia, pp. 133–138.
3. **Snaider, J., Franklin, S. (2014).** Vector LIDA. Procedia Computer Science, Vol. 41, pp. 188–203.
4. **Emruli, B., Sandin, F. (2013).** Analogical Mapping with Sparse Distributed Memory: A simple model that learns to generalize from examples. Cognitive Computation, Vol. 6, pp. 74–88.
5. **Gallant, S., Okaywe, T. (2013)**. Representing objects, relations and sequences. Neural Computation, Vol. 25, No. 8, pp. 2038–2078.
6. **Quiroz, J.I., Barrón, R., Ramírez, M.A. (2017).** Sequence prediction with Hyperdimensional Computing. Research in Computer Science, Vol. 138, pp. 117–126.
7. **Rahimi, A., Datta, S., Kleyko, D., Paxon, E., Olshausen, B., Kanerva, P., Rabaey, J. (2017).** High-Dimensional computing as a nanoscalable paradigm. IEEE Transactions on Circuits and Systems: Regular Papers, Vol. 99, pp. 1–14.
8. **Kanerva, P., Kristofersson, J., Holst, A. (2000).** Random Indexing of text samples for Latent Semantic Analysis. Proceedings of the 22nd Annual Conference of the Cognitive Science Society. New Jersey, USA.
9. **Harris, Z. S. (1954).** Distributional Structure. Word, Vol. 10, No. 2-3, pp. 146–162.
10. **McRae, K., Cree, G., Seidenberg, M., McNorgan, C. (2005).** Semantic feature production norms for a large set of living and nonliving things. Behavior Research Methods, Instruments & Computers, Vol. 37, No. 4, pp. 547–559.
11. **Kanerva, P. (1988).** Sparse Distributed Memory. Cambridge, MA: Bradford/MIT Press.
12. **Kanerva, P. (1996).** Binary spatter-coding of ordered *K*-tuples. Artificial Neural Networks – ICANN 96, pp. 896–873.
13. **Speer, R., Chin, J., Havasi, C. (2017).** ConceptNet 5.5: An open multilingual graph of general knowledge. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp. 4444–4451.

Article received on 05/03/2020; accepted on 19/02/2021.
Corresponding author is Ricardo Barrón Fernández.

ISSN 2007-9737

Test Case Generation Using Symbolic Execution

Saumendra Pattanaik[1], Bidush Kumar Sahoo[2], Chhabi Rani Panigrahi[3],
Binod Kumar Patnaik[1], Bibudhendu Pati[3]

[1] Siksha 'O' Anusandhan,
Dept. of Computer Science & Engineering,
India

[2] Gandhi Institute for Education & Technology Bhubaneswar,
Dept. of Computer Science & Engineering
India

[3] Rama Devi Women's University,
Dept. of Computer Science,
India

{saumendrapattnaik, binodpattanayak, bidush.sahoo,
panigrahichhabi, patibibudhendu}@gmail.com

Abstract. Testing is a well-known technique for identifying errors in software programs. Testing can be done in two ways: Static analysis and Dynamic analysis. Symbolic execution plays a vital role in static analysis for test case generation and to find the unreachable path with minimum test cases. Unreachable path is a part of a program which can never be executed i.e., the symbolic execution doesn't continue for that path and the current execution stops there. It generates a test suite for loop-free programs that is achieved by path coverage. In the best case program loops implies increase in the number of paths exponentially and in the worst case the program will not terminate. The functions of symbolic execution are test input generation, unreachable path detection, finding bugs in software programs, debugging. In this paper, we focus on dead code detection and test input generation using symbolic execution. Our execution for Java programs uses Java Path Finder (JPF) model tester. Our analysis shows that the symbolic execution method can be used to reduce symbolic execution time and to find out the unreachable path with less number of test cases.

Keywords. Symbolic execution, path coverage, unreachable path, test input generation.

1 Introduction

Testing is very important in the software evolution and maintenance process as it is required to find out the defects that were made in the development phase [21].

It is used for identifying the correctness and improving the quality of the software application. Testing can be done using static analysis or dynamic analysis. Symbolic execution plays an important role in static analysis for the test case generation and to explore the unreachable path with minimum number of test cases. Symbolic execution evaluates a program by considering inputs that results in execution of a program. This execution depends upon choosing of paths that are operated by a set of input values. A program is executed with symbols instead of real inputs in symbolic execution. Here, a source code is given as input for generating symbolic execution tree.

There are several reduction algorithms to reduce the symbolic execution tree. The reduction of the symbolic execution tree is done by eliminating the unreachable path and hence we generate the reduced test case generation. It basically focuses on generation of the test cases, unreachable path detection and model checking of concurrent programs which take inputs as the complex structures. The basic applications of symbolic execution are test input generation, unreachable path detection, finding bugs in software programs, debugging.

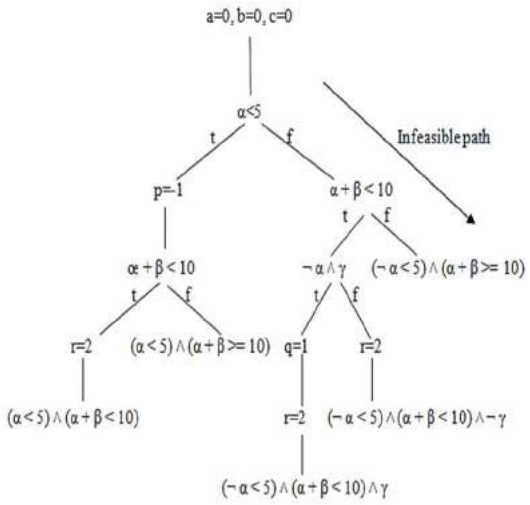


Fig. 1. A sample program on symbolic execution and the corresponding symbolic execution tree with symbolic values

Khurshid et al. [2] presented a two-fold observation of the traditional symbolic execution. Firstly, it enables the model checking of the concurrent programs to perform symbolic execution of the program.

Secondly, it provides a framework based on the symbolic execution algorithm that handles the complex data structures such as linked lists and trees. The symbolic execution is an efficient method that mainly focuses on achieving high coverage test suites, generates per-path guarantees and depicts the errors in the software programs.

Symbolic execution has been considered in the context of Symbolic Path Finder (SPF) [12].

SPF is an emblematic execution system based on the Java Path Finder (JPF) [13] show checking instrument set for Java Byte code investigation. Various applications of symbolic execution are test input generation, unreachable path detection, finding bugs in software programs, debugging.

In this paper, authors focus on dead code detection and test case generation techniques.

The paper is organized as follows. In Section 2, a sample program on symbolic execution is presented. Various dead code detection techniques proposed in the literature are presented in Section 3. Section 4 provides the symbolic execution techniques that are used in testing. In Section 5, the overall discussion on various techniques proposed in the literature for test case generation and dead code detection along with their advantages and disadvantages are presented.

2 Symbolic Execution Tree

A sample program to illustrate symbolic execution and the corresponding symbolic execution tree is shown as in Fig. 1.

```
int x = œ, y = ß, z = ŗ;
int p = 0, q = 0, r = 0;
if (x < 5) {
p = -1;
}
if (x + y < 10) {
if (!x && z)
{
q=1;
}
r=2;
```

Initially x, y and z have the symbolic values ŗ, ß and œ respectively. At each branch point, Path Condition (PC) is updated so that alternative paths can be chosen.

If the path condition evaluates to false, i.e., no data inputs pass through that path then it implies that the symbolic state is infeasible or is an unreachable path.

The unreachable path indicates that the symbolic execution for that path can not be continued and the current execution stops there.

3 Dead Code Detection Techniques

Dead code is the part of a program that can never be executed and whose result is never used in any other estimation. Various researchers [9, 22] and other sources as in [23] have found that the classical software code bases contains 5-10% “dead code”, i.e., code that can be eliminated without lowering the functionalities.

There are different techniques used for the detection of the dead code analysis. These techniques are described in the following subsections.

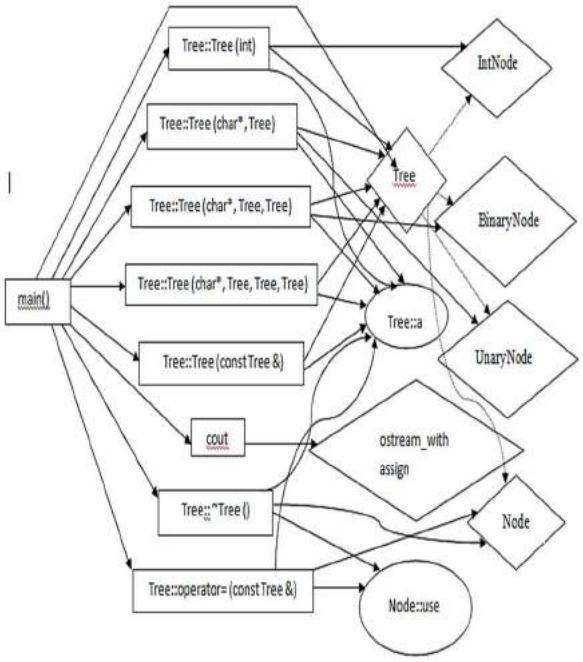


Fig. 2. Shows that all the members of the test class are exercised for the sample Tree class

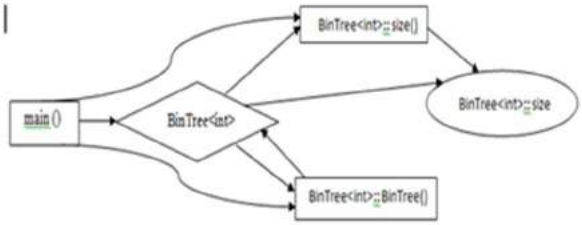


Fig. 3. Dependencies of Program entities on BinTree: size i.e. directly or transitively

3.1 Forward Reachability Analysis

Yih-Farn Chen et al. [9] basically found the representation logics that are used for reachability of code and detection of unreachable code techniques. These two tasks help to eliminate excess software baggage and holds software reuse metrics.

Forward Reachability Analysis is used for detecting dead code by determining a Reachable Entity Set and computing software reuse metrics. The reuse metrics is used to compute the code reuse and improve the quality and productivity. Reuse metrics consists of cost benefit analysis, percentage of reuse, reusability assessment, maturity assessment and failure modes analysis. There are two choices based on reachability of code and are described as follows:

- Software Reuse: It is also known as code reuse. It means using the existing software or to build a new software by using software knowledge. The goal of this is to reduce the cost of software production.

- Dead Code Detection: The part of the source code of a program which can never be executed, i.e., the symbolic execution does not

continue for that path and the current execution stops there:

```
Class Tree {
    Public:
    Tree (int);
    Tree (Char*, Tree);
    Tree (Char*, Tree, Tree);
    Tree (Char*, Tree, Tree, Tree);
    Tree (Const Tree & a);
    ~Tree () {
    if (-–b->use == 0)
    delete b;}};
    void operator = (const Tree & a);
    main () {
    Tree t = Tree (“-”, Tree (“*”, 7), Tree (“+”, 5, 3), Tree (“-”, 4,
                    5, 6));
    cout<< t << “\n”;
    t = Tree (“-”, t, t);
    cout<< t << “\n”;
    t = Tree (“+”, t, t, t);
    cout<<t<<”\n”;}
```

3.2 Reverse Reachability Analysis

Various authors have examined to use this technique of reachability code analysis. Researchers have found to support the complete reachability analysis that has been defined as an objective patterns at the selected abstraction level so that the programmers has the ability to analyze the different safeness implementations.

Reverse Reachability Analysis helps programmers to determine all the program entities that are dependent on an entity either directly or indirectly. If BinTree::size is altered, then other entities in the graph may be affected and is shown as in Fig. 3.

3.3 Visibility Analysis

Gansner et al. [9] accomplishes the reachability investigation on the regulation relationship in the class legacy chain of importance and discovers all part capacities and factors obvious to class BinaryNode in Koenig's illustration [10]. Regulation relationship happens between each parent class and a part.

Perceivability Analysis settles which part capacities and factors in a class legacy chain of command are seeable to a determined class. All part capacities in BinaryNode are seeable to itself. All open and shielded individuals from Node are likewise noticeable to BinaryNode on the grounds that BinaryNode has an open legacy association with Node.

Likewise, Node is a companion of Tree and along these lines all individuals from Tree are noticeable to Node. The visibility analysis of BinaryNode and Node is shown in Fig. 4.

4 Symbolic Execution Techniques in Testing

In this section, different techniques used for symbolic execution is presented.

4.1 Dynamic Symbolic Execution (DSE)

Data Flow Testing (DFT) [6] focuses on introducing a hybrid DFT structure. The heart of the structure is based on DSE and checks the reachability in software model checking to improve the testing performance.

DSE is a dynamic approach. It is a novel and efficient approach for automatic generation of the test cases. It combines the classical symbolic execution with real execution and generates many possible program pathways in the given amount of time. It aims at covering feasible pairs.

It takes the desired def-use pair du (l_d, l_u, x) as input and the Control Flow Graph (CFG) is constructed. It determines the test case for feasible test objectives and removes infeasible test objectives. It starts with arbitrary test input values. These test input values cause the execution path to cover the def-use pair. There are two approaches to deal with this problem.

Firstly we remove the invalid branching nodes by redefinition pruning technique and secondly, it applies Cut-Point Guided Search (CPGS) to choose which branching point to initially take. This approach can develop the DFT by 60-80% as compared to the testing span.The search results shows that the Dynamic Symbolic Execution approach reduces the DFT by 40% as compared to testing span to improve the testing performance than the Counter Example-Guided Abstraction Refinement (CEGAR) based approach.

Path explosion is a challenging problem in this approach because in a cheap amount of time to trigger the desired pair. Henceforth, more work will be done based on this approach to enhance the

data flow testing technique on large programs. Fig. 5 shows the workflow of DSE.

It starts with an arbitrary test inputs values followed by an execution path. Then, it removes the invalid branching nodes by redefinition pruning technique and finds the known paths by generating the constraint system.

4.2 Counter Example-Guided Abstraction Refinement (CEGAR)

CEGAR [6] is a static approach. It is utilized for creating the experiments and checking the fleeting wellbeing properties of the product. Here, a source program and an impermanent wellbeing determination is taken, which either demonstrates that the program fulfills the particular or produces a counter example to show the infringement.

This approach works in two stages: show checking and experiment created from the counterexamples. It first begins with a base or coarse program deliberation and over and again channels it so as to test for infeasible combine. We have to set a checkpoint to decide if the variable banner is valid.

The defutilize match is infeasible when the check point is inaccessible and a counterexample is returned when the def-utilize combine is attainable.

This approach is more intense than DSE approach as it enhances the scope by 20%. Consequently, more work will be done to make a profound correlation between both the methodologies. The workflow of CEGAR is shown in Fig. 6.

A C program is taken as input to the abstract model.

It then goes to the model checker where it automatically checks whether the model satisfies the given requirements and generates no error or bug found.

Then by the help of model checker, it goes to the simulator by taking counterexample as input where a program allows a computer to execute programs written for a different operating system by generating the simulation successfully and detecting the bug.

After that it goes to the refinement process where it removes the invalid the execution paths.

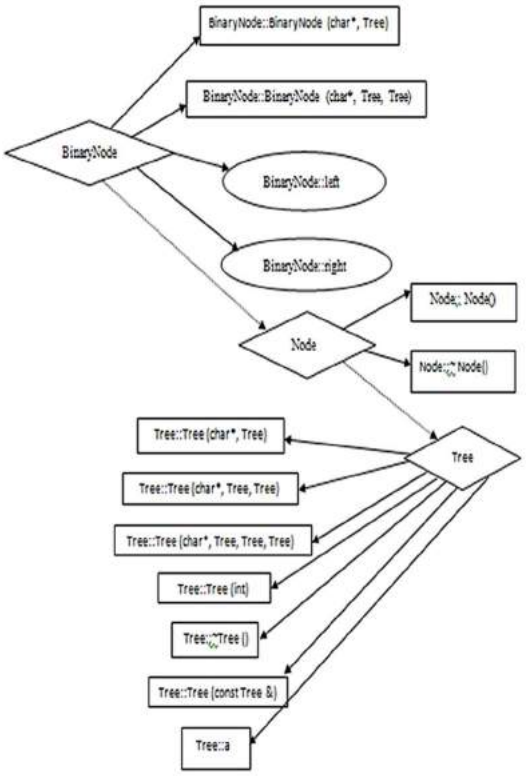


Fig. 4. Visibility analysis of BinaryNode and Node

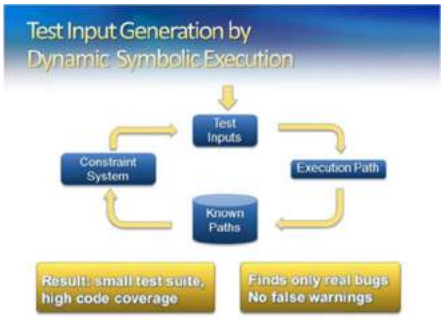


Fig. 5. Workflow of Dynamic Symbolic Execution

4.3 Document-Assisted Symbolic Execution (DASE)

DASE [3] is another novel approach for programmed age of the experiments and mistake recognizable proof to expand the adequacy and effectiveness of the representative execution. DASE separates the information impulse from

archives consequently and afterward utilizes the info impulse to center around testing mistake dealing with codes, which are vital that assistance the inquiry systems to enhance the representative execution and process the execution ways.

A program report is given as info that naturally separates the information impulse from that. There are two classes of info impulse: the setup of an information record and the substantial information estimations of order line decision.

The outcome shows that when contrasted with KLEE, DASE recognized 12 obscure imperfections that KLEE neglected to distinguish and out of 88 just 6 have been affirmed by the engineers.

DASE improves line investigation, branch examination and call investigation by 14.2–120.3%, 2.3–167.7%, and 16.9–135.2% in contrast with KLEE. Testing with invalid information is a testing issue. Subsequently, keeping in mind the end goal to test a program with invalid info esteems DASE approach centers by counterbalancing the information requirements.

4.4 Directed Automated Random Testing (DART)

DART [11] approach is also known as Concolic testing [14]. It dynamically operates the symbolic execution when the program is accomplished on real values. It supports two events: a real event and a symbolic event. A real event outlines all mutable to the real values and a symbolic event an outline all mutable to the non-real values.

A program is taken as input and the corresponding symbolic execution tree is drawn. It first starts with the generation of random inputs and execute the program concretely and symbolically. At each branch point, either it will follow the true path or it will end with the false path and generates the Path Condition (PC) for each path.

It finally surveys that all the paths of the program are examined and generates the test inputs. Path explosion and constraint solving is a challenging problem. There are two approaches to solve the path explosion problem: heuristics approach and sound program analysis approach.

There are two approaches to solve the constraint solving problem: irrelevant constraint elimination approach and incremental solving. Hence, more research can be performed for

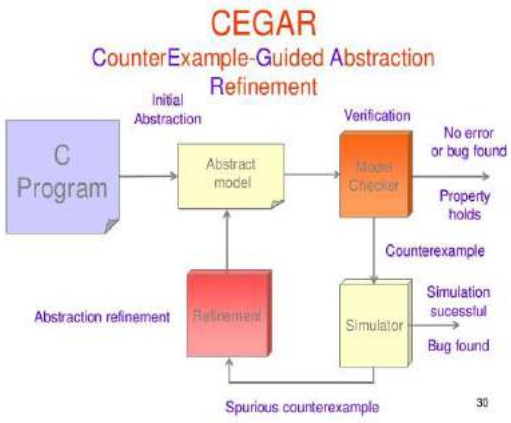


Fig. 6. Workflow of Counter Example-Guided Abstraction Refinement

automatic generatation of test inputs that covers the software bugs; generates the test suite that achieves high coverage; gives per-path guarantees and finding defects in software from low-level to high-level application programs.

The program analyzer which analyzes the behavior of the computer programs is divided into two parts. The first part is path selector in which control flow graph is taken as input and the second part is test data generator in which both control flow graph and data dependence is taken as input.

Data dependence is a condition where the program instruction refers to the data of the previous instructions. Then the path selector generates the selected paths and goes to the test data generator where the selected path information is taken as input to the path selector and produces the test data. Fig. 7 shows the workflow of DART.

4.5 Executed-Generated Testing (EGT)

EGT approach [15] is the instance of the modern symbolic execution techniques. It functions entirely between the real event and the symbolic event of the program. A source program is given as input and operated in the same program when the values are real.

It has the ability to mix both real and symbolic execution dynamically before finding all actions when the input values are all real. If so, then the action is executed in the same program or if more than one value is symbolic, the operation is

executed symbolically by maintaining a path condition for each path.

The disadvantages of this approach are elimination of irrelevant constraints, path explosion and memory modeling.

Hence, more research can be done in contributing a fashion to develop the test inputs that finds the errors which ranges from low to high-level syntactic features. Fig. 8 shows the workflow of EGT.

4.6 conc-iSE: Incremental Symbolic Execution Approach of Concurrent Programs

Incremental Symbolic Execution approach [16] for simultaneous programming is an approach to produce new test contributions by investigating the new execution ways between the two program variants.

These two program renditions i.e., old program variant P and new program form P' with an arrangement of execution abstract of program P is taken as information and over and over break down the present execution ways utilizing P'. Its yield is to examine the new recognition in P'.

This could be conceivable by evacuating the excess execution ways by rundown based calculation.

Consequently, amid the representative execution of P′, it breaks the regressive change-affect investigation. In this investigation, it fundamentally measures the arrangement of directions that may influence the changed guidelines set up of estimating the arrangement of guidelines that may be influenced by the changed directions [17].

The outcome demonstrates that this approach would overall be able to lessen the emblematic execution time and to expel the excess execution ways and string interleaving in the incremental representative execution.

The upper piece of the figure begins with the two program variants i.e., old program form P and new program adaptation P' with an arrangement of execution summation of program P is taken as info and more than once examine the present execution ways utilizing P' and produces the new recognition in P' by pruning the repetitive states or execution ways.

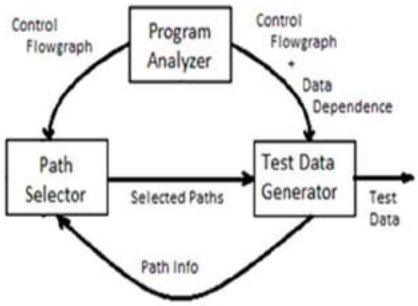


Fig. 7. Workflow of directed automated random testing

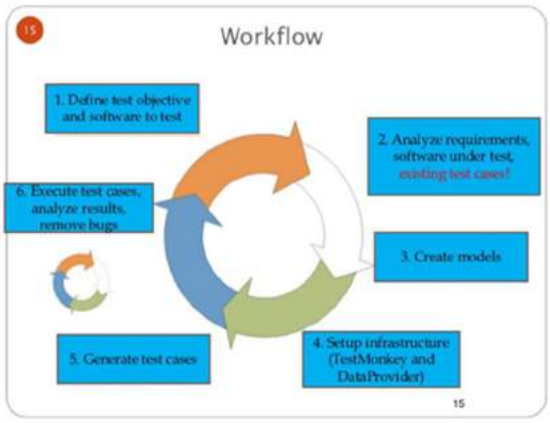


Fig. 8. Workflow of executed-generated testing

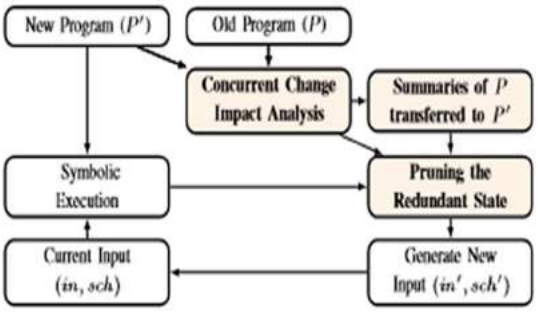


Fig. 9. Workflow of incremental symbolic execution

The lower some portion of the figure begins with a subjective current test inputs. Amid the representative execution, the new states are created and each new state produces another match incorporates the information info and string

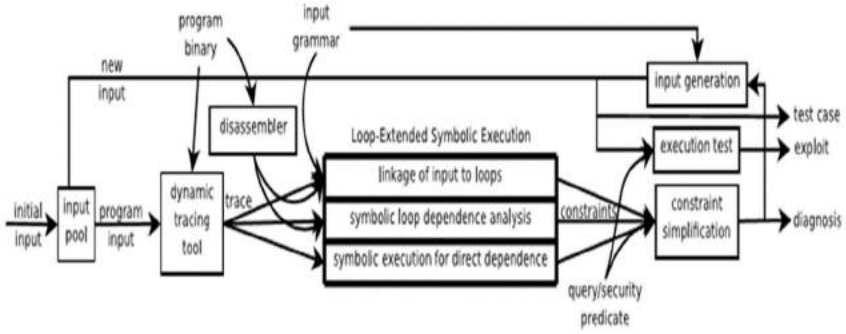


Fig. 10. Workflow of Loop-Extended Symbolic Execution

interleaving to touch base at the new state. The workflow of conc-iSE is shown in Fig. 9.

4.7 Loop-Extended Symbolic Execution (LESE)

Another representative execution strategy called LESE is proposed in [18] that sums up from the genuine execution to an arrangement of program executions that includes the distinctive number of redundancies for each circle as in the underlying execution.

Circle Extended Symbolic Execution is utilized to acquire appropriate outcomes when contrasted with emblematic execution when it is utilized as a part of projects with circles.

It makes mistake recognizing apparatuses more proficient and permits age of the experiments to achieve high test scope as fast as could reasonably be expected. A source program with circles is adopted as contribution to this strategy.

It begins with the predicate which is otherwise called inquiry predicate.

The predicate might be the branch condition related with the program point, and an execution that achieves the point, yet does not fulfill the predicate and after that yields to determine that the condition on a contribution to the program which creates the execution brings about a similar way and furthermore points the predicate to be valid.

It has the issue of recognizing and deciding support abundance impulse that produces on unmodified Windows and Linux sets.

The outcome demonstrates that the Loop-Extended Symbolic Execution can make an assortment of program examination, including the security applications, faster and more productive 19]. The workflow of LESE is shown in Fig. 10.

5 Discussion

In this section, we present the test case generation and dead code detection proposed in the literature along with their advantages and disadvantages, which are summarized in Table 1.

6 Conclusion

Various techniques used in symbolic execution provide a way to improve the test case generation and bug detection that achieves high test coverage suites, gives per-path correctness guarantees.

It also reduces the overall symbolic execution time and has the capacity to mix both real and symbolic execution [20]. In this work, we present different techniques that reduce the DFT in terms of testing time to improve the testing performance.

From the study, it is found that reduction of path explosion, constraint solving, determining buffer excess compulsion and the reduction of execution time in testing are some of the explored area in this field and can be taken as future work.

Tabla 1. Comparison of the proposed method with state of the art

Sl No.	Techniques Used	Test Case Generation	Dead Code Detection	Advantages	Disadvantages
1.	Dynamic Symbolic Execution (DSE)[11]	Analyzes program paths by determining path constraints.	Analysis of the reachability of code	-Designed a combined symbolic execution for automatic DFT.	-Path explosion is a challenging problem as it is required to find execution path to cover the desired pair. -DSE reduces the DFT in terms of testing time to improve the testing performance.
2	Counter Example-Guided Abstraction Refinement (CEGAR) [12]	Coverage criteria i.e., statement or branch coverage from counterexample paths.	Checking the practicability of the execution paths.	-Introduced a smooth encoding of DFT via CEGAR.	- Applying DFT on large multi-threaded software programs produces a broad analysis on it.
3.	Document-Assisted Symbolic Execution (DASE)[13]	Extracts input compulsion from documents automatically and focuses on execution paths that resemble valid inputs for improvement in the effectiveness of symbolic execution.		-Proposed and implemented an approach to improve symbolic execution for generation of test cases and bug finding.	-In order to test a program with invalid inputs, DASE approach focuses by cancelling out the input constraints.
4.	Directed Automated Random Testing (DART) [14]	Generates test suites that achieve high- coverage.	Examines the infeasible paths of the program using the depth-first search strategy.	-Capacity to combine both real and symbolic execution. -Proposed an effective symbolic execution technique to generate the inputs and performs symbolic execution dynamically. -Provides per-path correctness guarantees.	- Path explosion -Constraint solving
5.	Executed-Generated Testing (EGT) [15]	Approves the formation of high-coverage test suites.	Combines both real and symbolic execution and for the current path a path condition is maintained.	-Capacity to combine both real and symbolic execution. -Proposed a strategy to blend genuine and emblematic execution powerfully before checking each activity when esteems are for the most part genuine. -Achieves high test coverage suites.	-Elimination of irrelevant constraints. -Path explosion.
6.	conc-iSE: Incremental Symbolic Execution approach of concurrent programs[16]	Analyzes only the executions that affect the code changes between two versions of a program.	Reduces the overall symbolic execution time.	-Adapted an approach for concurrent programs to generate the new test inputs between the two program versions. -Reducing the symbolic execution time and removing the redundant execution path.	
7.	Loop-Extended Symbolic Execution (LESE) [18]	Allows to achieve high test coverage more quickly and to acquire better results when it is used in programs with loops.	-Analyzing buffer-overflow accountabilities in software programs after developing refuted candidates.	-Proposed a new approach by allowing test case generation to achieve high test coverage and automatic bug detection tools more effective.	-Identifying and determining buffer excess compulsion is a challenging problem that produces on unmodified Windows and Linux pairs.

References

1. **Cadar, C., Sen, K. (2013).** Symbolic execution for software testing: three decades later. Proceedings Communications of the ACM, Vol. 56, No. 2, pp. 82–90. DOI: 10.1145/2408776.2408795.

2. **Khurshid, S., Păsăreanu, C.S., Visser, W. (2003).** Generalized symbolic execution for model checking and testing. Proceedings International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Springer, Berlin, Heidelberg. pp. 553–568. DOI: 10.1007/3-540-36577-X_40.

3. **Wong, E., Zhang, L., Wang, S., Liu, T., Tan, L. (2015).** Dase: Document-assisted symbolic execution for improving automated software testing. Proceedings 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, pp. 620–631. DOI: 10.1109/ICSE.2015.78.
4. **Csallner, C., Tillmann, N., Smaragdakis, Y. (2008).** DySy: Dynamic Symbolic Execution for Invariant Inference. Proceedings of the 30th international conference on Software engineering, pp. 281–290.
5. **Guo, S., Kusano, M., Wang, C. (2016).** Conc-iSE: Incremental symbolic execution of concurrent software. Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, pp. 531–542. DOI: 10.1145/2970276.2970332.
6. **Su, T., Fu, Z., Pu, G., He, J., Su, Z. (2015).** Combining symbolic execution and model checking for data flow testing. Proceedings 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, Vol. 1, pp. 654–665. DOI: 10.1109/ICSE.2015.81.
7. **Kersten, R., Person, S., Rungta, N., Tkachuk, O. (2015).** Improving coverage of test cases generated by symbolic pathfinder for programs with loops. ACM SIGSOFT Software Engineering Notes, Vol. 40, No. 1, pp. 1–5. DOI: 10.1145/2693208.2693243.
8. **Saxena, P., Poosankam, P., McCamant, S. Song, D. (2009).** Loop-extended symbolic execution on binary programs. Proceedings of the eighteenth international symposium on Software testing and analysis, pp. 225–236. DOI: 10.1145/1572272.1572299.
9. **Chen, Y.F., Gansner, E.R., Koutsofios, E. (1998).** A C++ data model supporting reachability analysis and dead code detection. IEEE Transactions on Software Engineering, Vol. 24, No. 9, pp. 682–694. DOI: 10.1109/32.713323.
10. **Koenig, A. (1988).** An example of dynamic binding in C++. Journal of Object-Oriented Programming, Vol. 1, No. 3, pp. 60–62.
11. **Godefroid, P., Klarlund, N., Sen, K. (2005).** DART: directed automated random testing. Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation, pp. 213–223. DOI: 10.1145/1065010.1065036.
12. **Păsăreanu, C.S., Rungta, N. (2010).** Symbolic PathFinder: symbolic execution of Java bytecode. Proceedings of the IEEE/ACM international conference on automated software engineering, pp. 179–180. DOI: 10.1145/1858996.1859035.
13. **Anand, S., Păsăreanu, C.S., Visser, W. (2007).** JPF–SE: A symbolic execution extension to java pathfinder. International conference on tools and algorithms for the construction and analysis of systems, pp. 134–138. Springer, Berlin, Heidelberg.
14. **Sen, K., Marinov, D., Agha, G. (2005).** CUTE: a concolic unit testing engine for C. ACM SIGSOFT Software Engineering Notes, Vol. 30, No. 5, pp. 263–272. DOI: 10.1145/1095430.1081750.
15. **Betts, A., Chong, N., Deligiannis, P., Donaldson, A.F., Ketema, J., (2017).** Implementing and evaluating candidate-based invariant generation. IEEE Transactions on Software Engineering, Vol. 44, No. 7, pp. 631–650. DOI: 10.1109/TSE.2017.2718516.
16. **Ernst, M.D., Cockrell, J., Griswold, W.G., Notkin, D. (2001).** Dynamically discovering likely program invariants to support program evolution. IEEE Transactions on Software Engineering, Vol. 27, No. 2, pp. 99–123. DOI: 10.1109/32.908957.
17. **Abdulla, P., Aronis, S., Jonsson, B., Sagonas, K. (2014).** Optimal dynamic partial order reduction. ACM SIGPLAN Notices, Vol. 49, No. 1, pp. 373–384. DOI: 10.1145/2578855.2535845.
18. **Chattopadhyay, A. (2014).** Dynamic invariant generation for concurrent programs. Doctoral dissertation, Virginia Tech.
19. **Nimmer, J.W., Ernst, M.D. (2002).** Invariant inference for static checking: An empirical evaluation. ACM SIGSOFT Software Engineering Notes, Vol. 27, No. 6, pp. 11–20. DOI: 10.1145/605466.605469.
20. **Allen-Weiss, M. (2007).** Data structures and algorithm analysis in C++. Pearson Education India.
21. **Panigrahi, C.R., Mall, R. (2010).** Model-based regression test case prioritization. ACM SIGSOFT Software Engineering Notes, Vol. 35, No. 6, pp. 1–7. DOI: 10.1145/1874391.1874405.
22. **Streitel, F, Steidl, D., Jürgens, E. (2014).** Dead code detection on class level. Softwaretechnik-Trends, Vol. 34, No. 2.
23. **Pizzutillo, P. (2013).** Static analysis: Leveraging source code analysis to reign in application maintenance cost.

Article received on 27/12/2020; accepted on 14/11/2021.
Corresponding author is Chhabi Rani Panigrahi.

ISSN 2007-9737

Sentiment Analysis of COVID19 Reviews Using Hierarchical Version of d-RNN

Arindam Chaudhuri

Samsung R & D Institute Delhi India,
NMIMS University Mumbai,
India

arindamphdthesis@gmail.com

Abstract. In recent years understanding person's sentiments for catastrophic events has been a major subject of research. In recent times COVID19 has raised psychological issues in people's minds across world. Sentiment analysis has played significant role in analysing reviews across wide array of real-life situations. With constant development of deep learning based language models, this has become an active investigation area. With COVID19 pandemic different countries have faced several peaks resulting in lockdowns. During this time people have placed their sentiments in social media. As review data corpora grows it becomes necessary to develop robust sentiment analysis models capable of extracting people's viewpoints and sentiments. In this paper, we present a computational framework which uses deep learning based language models through delayed recurrent neural networks (d-RNN) and hierarchical version of d-RNN (Hd-RNN) for sentiment analysis catering to rise of COVID19 cases in different parts of India. Sentiments are reviewed considering time window spread across 2020 and 2021. Multi-label sentiment classification is used where more than one sentiment are expressed at once. Both d-RNN and Hd-RNN are optimized by fine tuning different network parameters and compared with BERT variants, LSTM as well as traditional methods. The methods are evaluated with highly skewed data as well as using precision, recall and F1 scores. The results on experimental datasets indicate superiority of Hd-RNN considering other techniques.

Keywords. Sentiment analysis, viewpoints, sentiments, RNN, d-RNN, BERT, Hd-RNN.

1 Introduction

Coronavirus 2019 (COVID19) [1, 2, 3, 4] is a global pandemic since past two years. It has almost ruined mankind with catastrophic implications [5] having major impact on global economy. This has led to unprecedented rise in unemployment, psychological issues and depression of people around world. The abrupt social, economic and travel changes have motivated research in several domains [6, 7, 8, 9]. In India COVID19 has adversely affected its economy in past two years [10]. It has battered Indian economy in two major COVID19 waves as shown in Fig. 1 which country has seen.

As of January 2022 considering oficial figures, India stands at second highest number of confirmed cases in world after United States and third highest number of deaths after United States and Brazil. However it is assumed that there has been high degree of under reporting in COVID19 cases.

During this phase of 2 years there has been unprecedented growth of social media usage such as Twitter by people where they have expressed several concerns related to their living conditions, psychology [11, 12, 13, 14] and mental health [15]. This data has been used for research in behavioural sciences [16]. It has also been used for personality prediction [11, 12] as well as understanding trends and backgrounds of users online [17, 18].

In view of this research on sentiment analysis of COVID19 reviews has become centrestage of attention for social media analytics.Sentiment analysis involves natural language processing (NLP) [19, 20, 21] in order to extract systematically affective states, attitudes and opinions of individuals or social groups [22, 23, 24] across various domains such as politics, sociology, business intelligence, etc. [25].

Fig. 1. COVID19 has battered Indian economy

Classification of sentiments within any specific domain involves any event, item, topic, product, application or others. The importance of sentiment analysis lies in many tasks such as digital mental health [26], recommendation systems [27] as well as intelligent cognitive assistants [28].

The major prima face in using sentiment analysis is towards text classification having polarities [29]. Polarity comes in several forms including sentiment class having several values such as very positive, positive, neutral, negative and very negative. However, polarity values and classes differ from one application to another. Sentiment expressions with happy, sad, angry etc emotions alongwith sentiment classes can be analyzed through sentiment analysis.

Text classification can be either subjective or objective [30] in nature. Sentiment analysis approaches are basically classified as three major groups [29] viz (a) lexicon-based approach (b) machine learning approach and (c) hybrid approach. In lexicon-based approaches, polarity is extracted with respect to predefined lexicon or dictionary. Sentiment analysis model is trained using dataset commonly known as corpus. In machine learning approaches, algorithm is trained considering given dataset in order to build classification model which extracts sentiment polarity of given text. Hybrid approach is most suitable for real life applications which combine both lexicon-based and machine learning methods to perform sentiment analysis activity.

One of the commonly used machine learning approaches is artificial neural networks (ANN) [31, 32]. Closely associated with ANN are deep learning networks [33, 34, 35].

In recent past both these methods have been used in abundance in sentiment analysis research [36]. Deep learning networks are more powerful than ANNs. They are blessed with better data representation capabilities with good features and multiple representation levels than ANNs. This allows different computer recognition tasks including text mining, image processing and pattern recognition to be performed more smoothly and efficiently. Deep learning networks allow representation of words considering textual data and produce word embedding which could be used by different machine learning approaches.

Several types of deep learning networks have been used for NLP such as feedforward neural networks, convolutional neural networks (CNN) and recurrent neural networks (RNN) [33, 34]. Deep learning networks have outperformed other machine learning methods in several NLP tasks including machine translation and named-entity recognition [37, 38]. Deep learning-based approaches for sentiment analysis are further classified in accordance to deep learning networks being used. There are basically three different categories viz: (a) feed forward neural network-based sentiment analysis, (b) CNN based sentiment analysis, and (c) RNN based sentiment analysis approaches. The majority of sentiment analysis approaches in previous few decades have been focused towards identification of text polarity. Deep learning models have played a significant role in forecasting COVID19 infection trends [39] in various parts of world.

In this research work, novel sentiment analysis methods have been used considering state-of-the-art methods towards understanding people's behavior arising from COVID19.

This work provides motivational direction towards making society aware of the fact that dissemination of useless information in a sensitive topic like COVID19 is harmful to mankind. It considers psychological well-being with respect to constant rise in number of active COVID19 cases during peak periods.

Considering abundant corpus of text data available for this research, here computational modeling and machine learning methods [20, 21, 36] have been primarily used. However, there have been number of challenges arising due to testing and reporting of COVID19 cases [39].

It has been observed that an appreciable amount of available reported data is plagued with false figures. This issue is addressed by re-collecting that portion of data from other reliable sources [39].

The computational framework highlights multi-label sentiment classification with more than one sentiment expressed at once. In order to achieve this deep learning based language models considering delayed recurrent neural networks (d-RNN) and hierarchical version of d-RNN (Hd-RNN) are used to address rise of COVID19 cases in different parts of India.

All sentiments are reviewed with respect to time window spread across January 2020 and June 2021 when India witnessed most active cases.

Both d-RNN and Hd-RNN are optimized by fine tuning different network parameters. The experimental hypothesis is made stronger by performing comparative performance analysis of d-RNN and Hd-RNN models with some traditional machine learning models like naive bayes (NB), support vector machine (SVM), k-nearest neighbor (kNN), random forest (RF), gradient boosting (GB), ada boost (AB) and decision trees (DT) [36].

We use different variants of BERT [36] in order to evaluate test datasets with prediction validation. LSTM [20, 21] and BD-LSTM [20, 21] models are also used in comparing performance of d-RNN and Hd-RNN.

The results on experimental datasets highlight Hd-RNN's superiority with respect to other techniques.

The novelty of this work is attributed through following points: (a) Preparation and analysis of COVID19 sentiment data spanning across time period of 1.5 years, (b) Computational framework comprising of novel RNN models such as d-RNN and Hd-RNN, (c) Comparative analysis with other models relevant to COVID19, (d) All methods are evaluated with highly skewed data where precision, recall and F1 scores are used.

This paper is organized as follows. In section 2 work related to sentiment analysis is presented. The computational methodology is discussed in section 3. In section 4 experiments and results are highlighted. Finally, conclusion is given in section 5.

2 Related Work

The process of feature extraction from text for NLP related tasks is known as word embedding [40, 41, 42]. Some of the common examples of word embedding include sentiment analysis. As such significant amount of research work has been performed on sentiment analysis using machine learning. It is basically obtained using methods where words or phrases from vocabulary are mapped to real number vectors. It involves mathematical embedding from large corpus with multi-dimensions per word to a vector space. The lower dimension here is used by machine learning or deep learning models for text classification [40].

Basic word embedding methods such as bag-of-words (BOW) [43] and term frequency inverse document frequency (TF-IDF) [44] do not have context awareness and semantic information in embedding. This problem is related to skip-grams [45] which use n-grams involving bigrams and tri-grams to develop word embedding. This allow adjacent word token sequences which needs to be skipped [46].

There has been considerable progress in word embedding and language models since last few decades. In [47] word2vec embedding is proposed which uses feedforward neural network model. They learn association between words from text dataset which detects synonymous words or suggest additional words with respect to a partial sentence.

It uses continuous bag-of-words (CBOW) or continuous skip-gram model architectures in order to produce distributed words' representation. The method creates large vector which represent each unique word in corpus. Here semantic information and relation between words are preserved. For any two sentences which do not have much common words, their semantic similarity is captured using word2vec [47].

However, word2vec does not well represent word context. To obtain vector representations for words, in GloVe [48] words are mapped into meaningful space where word distance is related to semantic similarity. It uses matrix factorization in order to construct large matrix of co-occurrence information. This results in representation which shows linear substructures of word vector space.

With top list words embedding feature vectors match within certain distance measures.

The relations between words such as synonyms, company-product relations etc can be found using GloVe. A gender-neutral GloVe has been proposed [49] where it has gender biased information.

In [50] word embedding methods evaluation is provided methods including GloVe [54], skip-gram and continuous space language models (CSLM) [51]. It has been observed that in all language related tasks skip-gram and GloVe have outperformed CSLM.

In [52], an evaluation of word embedding methods have been performed for biomedical text analysis applications. Here it has been observed that word embedding trained from clinical notes and literature better captured word semantics.

In [53], classification of twitter data using machine learning is performed with 89.47% accuracy.

In [54], sentiment analysis on Uri attack has been performed in order to mine emotions and polarity on Twitter data. The dataset comprised of about 5000 tweets. The experimental results showed Uri attack disgusted 94.3% of individuals.

In [55], sentiment analysis using machine learning for business intelligence has performed.

In [56], analytical categorization and evaluation of prevalent testing techniques and deployment of sentiment analysis machine learning techniques on different applications have been performed.

In [57], Naïve Bayes and OneR have been used for sentiment analysis.

Another work from [58] also uses various machine learning algorithms such as naïve bayes, support vector machine (SVM), logistic regression, decision trees, k-nearest neighbor and random forest for sentiment analysis.

In [59], distribution of COVID19 vaccines implies an urgent need to track and understand public opinion on an ongoing basis to establish baseline vaccine confidence levels in order to detect early confidence loss warnings.

Research on public attitudes with facebook and twitter towards COVID19 vaccinations uses Artificial Intelligence analysis [60] in UK and US.

In [61], a study is performed on social network analysis of COVID19 sentiments using machine learning methods with twitter data. This study creates sentiment analysis through large number of tweets. The results are categorized considering consumers' viewpoint into positive and negative with respect to tweets [62].

Another significant work on sentiment analysis for COVID19 and infectious diseases can be found [63].

In [64], public sentiments on COVID19 using machine learning for tweet classification is performed where classification methods like naive bayes and logistic regression are used.

LSTM and BERT language models have also provided promising results [65, 66] for sentiment analysis. Both LSTM and BERT belong to family of RNNs [67, 68].

LSTMs are characterised by two important gates viz forgot and output gates which makes them efficient sentiment analysis models. LSTM has feedback relation which helps in processing both single and sequential data points. Many versions of LSTM have been developed till date. For smaller datasets bidirectional LSTM models produce better results than BERT models.

These models are trained in lesser time than their pre-trained counterparts [69]. In LSTM are that words are passed in and generated sequentially. For capturing true meaning of words even bi-directional LSTMs do not have good performance.

These issues are resolved by BERT efficiently [70]. BERT is based on transformers where each output element is connected to each input element. They have been unveiled by Google in 2017. BERT architecture proved to be game changer in NLP which allow transfer learning usage in several tasks. It uses adjacent text in order to assist machines in interpretation of ambiguous language in text.

Some recently proposed significant sentiment analysis approaches are presented in Table 1.

In [71], feed forward neural network is used on Arabic tweets with accuracy and precisión of 90% and 93.7% respectively.

Appreciable results are available in [72] where CNN is used on movie review datasets.

In [73] and [74], CNN have been used on SentiStrength (text) and SentiBank (visual) as well as manually annotated datasets with accuracy values of 79% and 95% respectively. With 8,000

Table 1. Recently proposed significant sentiment analysis approaches

Paper	Year	ANN Models	Datasets	Results
[75]	2020	CNN	8000 comments and posts	Accuracy 90.9%
[83]	2020	word embedding	Tweets	Accuracy 62.8%
[76]	2019	CNN and LSTM	Lithuanian Internet comments	Accuracy 70.6%
[80]	2019	LSTM	Militarylife PTT	Accuracy 85.4% F1-Score 88.41%
[81]	2019	LSTM	Roman Urdu datasets	Accuracy 95.2%
[74]	2018	CNN	Manually annotated dataset	Accuracy 95%
[79]	2018	LSTM	504 news headlines and 675 microblog messages	
[78]	2018	RNN	Twitter posts and news headlines in financial domain	
[71]	2017	Feed Forward	Arabic tweets	Accuracy 90% Precision 93.7%
[77]	2017	RNN	Amazon health product reviews, SST-1 and SST-2	GRU is best (Accuracy)
[72]	2016	CNN	Movie reviews and IMDB	
[73]	2015	CNN	SentiStrength (text) SentiBank (visual)	Accuracy 79%
[82]	2015	word embedding	SemEval 2013	
[84]	2015	auto-encoder	Arabic Tree Bank	Accuracy 73.5%

comments and posts CNN [75] have produced accuracy of 90.9%.

In [76] CNN and LSTM models on Lithuanian internet comments have provided an accuracy of 70.6%. In [77] RNNs with Amazon health product reviews, SST-1 and SST-2 considering accuracy have given best results for GRU. Another significant results can be found in [78] and [79] with RNNs.

In both of these Works datasets considered constituted new headlines with microblog messages as well as twitter posts. In [80] LSTM on Militarylife PTT datasets have produced accuracy and F1 Score of 85.4% and 85.4% respectively. In [81] LSTM on roman urdu datasets have produced accuracy of 95.2%. In [82] and [83] word embedding have been used on SemEval 2013 and tweets. In [84] auto-encoders have been used on Arabic tree bank.

3 Computational Methodology

The computational methodology for sentiment analysis of COVID19 reviews is presented in this section. The section starts with discussion on COVID19 datasets. This is followed by brief introduction on RNN and delayed RNN. Based on RNN and d-RNN, Hd-RNN is presented.

Next an approximation of hierarchical delayed RNN with hierarchical bidirectional RNN is performed. The section concludes with twitter-based sentiment analysis framework for COVID19 in India.

3.1 Datasets

Due to non-availability of any standard COVID19 datasets, in this research we have developed datasets considering various factors. Some of the significant factors considered while preparing this dataset include time window, geographical region, age group, gender, testing frequency and COVID19 vaccine taken. The datasets are prepared considering tweet data available from Twitter with respect to abovementioned factors.

The prepared datasets are validated against available benchmark datasets [36]. The time window here spans from January 2020 to June

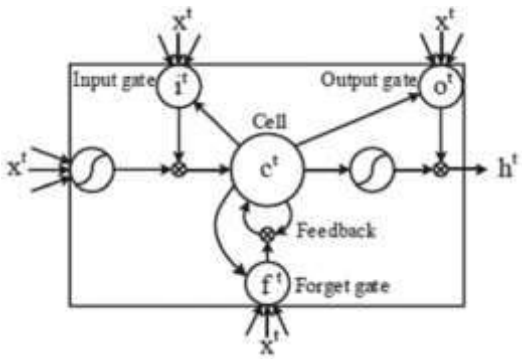


Fig. 2. LSTM memory block with single cell [20]

2021 when there were most active cases were detected in India. It is during this period India had two COVID19 waves, first one from March 2020 to September 2020 and second one from February 2021 to May 2021.

The geographical region included those states where most COVID19 cases are detected during stated time window.

In view of this, 5 top Indian states viz Maharashtra, Delhi, Karnataka, Kerala and Tamil Nadu are considered. The age groups are divided into 5 intervals viz 0-20, 20-40, 40-60, 60-80 and 80+. The gender considered as male and female. The testing frequency involves COVID19 tests performed per day by both government and non-government testing agencies. The COVID19 vaccine taken highlights those people in abovementioned states who have been vaccinated.

3.2 Recurrent Neural Networks

RNNs are an important category of deep learning networks [85] with infinite impulse response. The different computational units of RNNs are connected together to form directed circle. As a result of this, these networks create an internal state which highlights dynamic temporal behavior.

The arbitrary input sequences are processed through internal memory. This makes them readily applicable for non-segmented handwritten or speech recognition tasks. They are basically Turing complete [86] which provides capability to run arbitrary programs in order to process arbitrary input sequences.

Since RNNs can be trained in either supervised or unsupervised manner they are more efficient than traditional ANNs and SVMs [85, 86]. RNNs learn intrinsic characteristics about data without target vector's help. This learning capability is stored as network weights. The network's unsupervised training has similar input as target units.

In deep learning network's architecture is optimized through several routines. The network is treated as directed graph where different hidden units are connected to each other. Each hidden layer in network is non-linear combination of layers.

This is because combination of outputs from all previous units works with their activation functions. Each hidden layer becomes optimally weighted and non-linear, when optimization routine is applied to network. Each hidden layer becomes low dimensional projection of below layer, when each sequential hidden layer has fewer units than one below it.

The recurrent structure of network allows modeling of contextual information for temporal sequences. Due to issues of vanishing gradients and error blowing up problems [86], it is very difficult to train these networks with commonly used activation functions.

This is addressed through LSTM architecture [85, 86] which replaces non-linear units in traditional RNNs. LSTM memory block with single cell is highlighted in Fig. 2. It has one self-connected memory cell and three multiplicative units viz input, forget and output gates. These gates store and access long range temporal sequence based contextual information. The activations of memory cell and three gates are available in [85]. It has been shown that topological enhancements RNNs increases their expressive power and representation capacity [87].

The two most common enhancement strategies are: (a) stacked RNNs which increases learning non-linear functions capacity and (b) bidirectional processing which uses acausal information in sequence. The basic mathematical background to increase network's depth for single layer RNN is presented here. In view of this, let us consider an input sequence $\{y_p\}_{p=1,\ldots\ldots,P}, y_p \in \mathbb{R}^m$ such that single layer RNN is specified as:

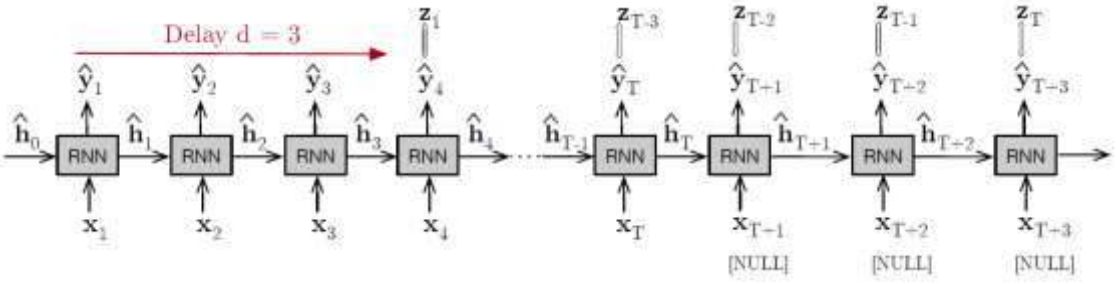


Fig. 3. d-RNN with sequence of T elements [87]

$$\hat{s}_p = g\left(\hat{W}_y y_p + \hat{W}_s \hat{s}_{p-1} + \hat{b}_s\right), \quad (1)$$

$$\hat{t}_p = h\left(\hat{W}_o \hat{s}_p + \hat{b}_o\right). \quad (2)$$

Here $g(\cdot)$ and $h(\cdot)$ are element-wise activation functions, $\hat{s}_p \in \mathbb{R}^n$ represent hidden state at timestamp p with n units and $\hat{t}_p \in \mathbb{R}^n$ represent network outputs.

The parameters include input weights $\hat{W}_y$, recurrent weights $\hat{W}_s$, bias term $\hat{b}_s$, output weights $\hat{W}_o$, bias term $\hat{b}_o$ with initial state as $\hat{b}_o$. The depth in RNNs is basically provided by stacked recurrent units [87]. With respect to equations (1) and (2) a stacked RNN with j layers are represented as:

$$s_p^{(1)} = g\left(W_y^{(1)} y_p + W_s^{(1)} s_{p-1}^{(1)} + b_s^{(1)}\right), i = 1, \quad (3)$$

$$s_p^{(i)} = g\left(W_y^{(i)} s_p^{(i-1)} + W_s^{(i)} s_{p-1}^{(i)} + b_s^{(i)}\right), \quad i = 2, \ldots\ldots, j \quad , \quad (4)$$

$$t_p = h\left(W_o s_p^{(j)} + b_o\right). \quad (5)$$

Here activation function and parametrization abide by single layer RNN. The weights and bias terms for each layer j are represented as $W_y^{(i)}$, $W_s^{(i)}$ and $b_s^{(i)}$. For this layer hidden state at timestamp p is $s_p^{(i)}$. Corresponding to j layers, stacked RNN has initial hidden state vectors as $s_0^{(1)}, \ldots\ldots, s_p^{(j)}$.

3.3 Basic d-RNN

An alternative means to increase RNNs' depth is to consider time within single layer RNN. Single layer RNNs are restricted considering number of non-linearities applied to recent inputs. [87] have addressed this restriction by adding intermediate non-linearities between input elements. Here computational steps are added between elements in sequence which increases runtime complexity. The delayed recurrent neural networks (d-RNNs) addresses this by increasing effective depth through introduction of delay between input and output.

The d-RNN can be defined as single layer RNN such that for any input respective output is obtained d timesteps later as shown in Fig. 3. Here d is network's delay. The initial hidden state for d-RNN is initialized in similar manner as an RNN. Delaying output requires special considerations on data that differ slightly from RNN. Input sequences need to have $P + d$ elements.

Depending on task being solved this can be achieved by adding null input element or including d additional elements in input sequence. When doing forward pass over d-RNN for inference, outputs from $p = 1$ to d are discarded as output appears after a delay. The output sequence has P elements. Training loss is computed by comparing expected output for input with delay factor. Thus, gradients are backpropagated only from delayed outputs.

Another RNN very similar in structure to single layer RNN is stacked RNN where additional in between layer connections are placed which adds depth in network. Any stacked RNN can be configured into a single-layer d-RNN which produces exact hidden states and output sequences. The depth from in between layer connections are replaced with temporal depth applied through output delays. Considering above equations parameters of single layer RNN using

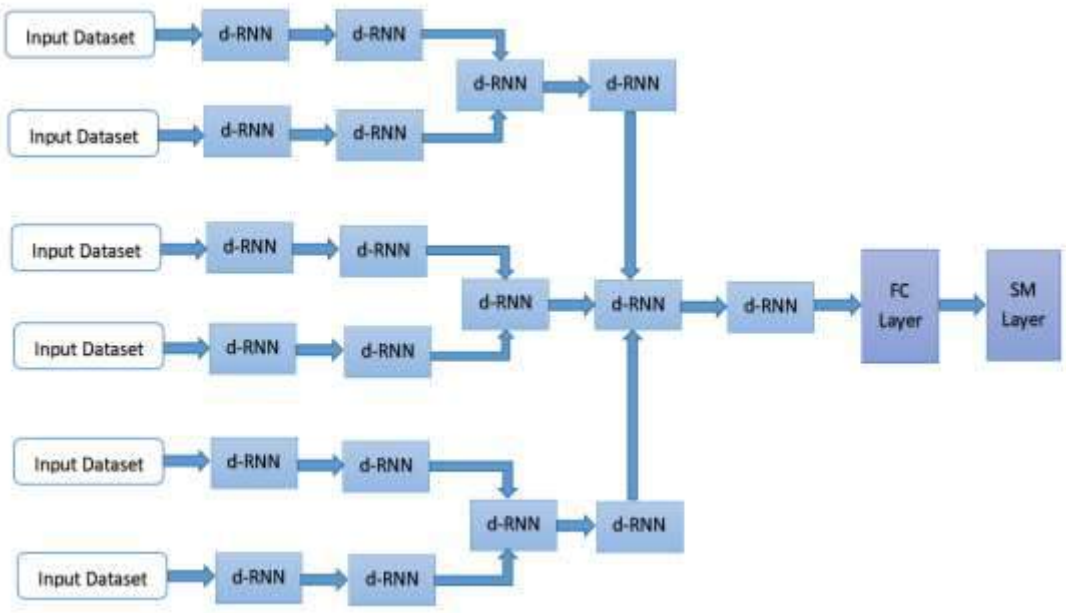


Fig. 4. Schematic representation of architecture of Hd-RNN [36]

weights and bias terms of k-layer stacked RNN as highlighted in [87]. It is observed from [87] that stacked RNN's each layer is converted into group of units in single layer RNN.

The structure of recurrent weight matrix $\widehat{W}_h$ considers hidden state to act as buffer. Here each group of units receive inputs from itself and previous group.

This buffering mechanism processes information which eventually arrives at output after $k-1$ timesteps. The model achieved is d-RNN with delay $k-1$ and sparsely constrained weights. It is to be noted that d-RNN performs identical computations as stacked RNN by maintaining depth in layers for depth in time.

It has been proved that d-RNN parameterized by above equations is exactly equivalent to stacked RNN in above equations. This proposition can be extended towards recurrent cells with more complexity. A k-layer stacked RNNs can be represented as single layer d-RNN. The d-RNN in its weight matrices has specific sparsity structure which is not present in generic RNN or d-RNN. As such stacked RNN and d-RNN with sparsely constrained weights models are equivalent in nature. They can be interchanged using weight matrix definitions in above equations.

3.4 Hierarchical d-RNN

Considering RNN and d-RNN in previous sections, we now discuss Hd-RNN [36] for semantic analysis of COVID19 reviews. The better approximation of network topologies like stacked RNN, bidirectional RNN and stacked bidirectional RNN by d-RNN with faster runtimes [36] serve major motivation. Hd-RNN differs from its non-hierarchical counterparts with respect to better classification accuracy taking similarities and running time parameters as data corpus grows [36].

A schematic representation of architecture of Hd-RNN is shown in Fig. 4. Here, d-RNN is used to model temporal sequences in COVID19 reviews. The results from d-RNN are combined together to form Hd-RNN. Hd-RNN comprises of 7 layers d-RNN$_1$→d-RNN$_2$→d-RNN$_3$→d-RNN$_4$→d-RNN$_5$→fc→sm.

Here, d-RNN$_i$ with $(i = 1, 2, 3, 4, 5)$ depict layers with d-RNN nodes, fc is fully connected layer and sm is softmax layer. Each layer in Hd-RNN constitutes classifier's hierarchy and addresses

classification tasks [36] which plays a vital role in network's success.

In order to recover any single hierarchy, we can run split d-RNN on small subset of reviews having few words [36]. This helps in computation of seed classification value.

The input dataset subsets are developed randomly which is initiated at layer d-RNN$_1$. With initial classification value remaining part of data corpus is placed into seed class for which average similarity is present.

This leads to classification of entire dataset using only similarities to words with respect to small subset. This process is applied recursively to each class such that Hd-RNN is build up considering only small similarities fraction. The classification process continues till d-RNN$_5$.

This recursive phase has no measurements between classes at previous split. This results in robust version of Hd-RNN which aligns its measurements ms in order to resolve higher class structure resolution. The pseudo code for Hd-RNN is highlighted in Algorithm below.

Algorithm: $\mathrm{HdRNN}\left(\mathrm{dRNN}, ms, \{x_i\}_{i=1}^{WR_j}, CS_j\right)$

if $WR_j < ms$ **then return** $\{x_i\}_{i=1}^{WR_j}$

Select $W \subseteq \{x_i\}_{i=1}^{WR_j}$ of size w uniformly at random

$C_1', \ldots\ldots, C_{CS_j}' \rightarrow \mathrm{dRNN}(W, CS_j)$

Set $C_1 \leftarrow C_1', \ldots\ldots, C_{WR_j} \leftarrow C_{WR_j}'$

for $x_i \in \{x_i\}_{i=1}^{WR_j} \backslash W$ **do**

$\forall s \in [CS_j],\ \alpha_s \leftarrow \frac{1}{|C_j'|}\sum_{x_h \in C_j'} S(x_i, x_h)$

$C_{\mathrm{argmax}_{s \in [CS_j]}\alpha_s} \leftarrow C_{\mathrm{argmax}_{s \in [CS_j]}\alpha_i} \cup \{x_i\}$

endfor

output $\left\{C_s, \mathrm{HdRNN}\left(\mathrm{dRNN}, ms, C_s, CS_j\right)\right\}_{j=1}^{CS_j}$

Hd-RNN is specified with respect to success probability of success in recovering true hierarchy CS^*, measurement ms and runtime complexity. Certain restrictions are placed on similarity function SM such that similarities agree with hierarchy up to some random noise:

P1 For each $x_i \in CS_j \in CS^*$ and $i \neq j$ we have:

$$\min_{x_p \in CS_j} \mathbb{E}\mathrm{x}\mathbb{p}\left[S(x_i, x_p)\right] - \min_{x_p \in CS_j'} \mathbb{E}\mathrm{x}\mathbb{p}\left[S(x_i, x_p)\right] \geq \delta \geq 0.$$

Here expectations are taken with respect to possible noise on SM.

P2 For each $x_i \in CS_j$, a set of W_j words of size w_j drawn uniformly from CS_j satisfies:

$$\mathbb{P}\mathrm{r}\mathbb{o}\mathbb{b}\left(\min_{x_p \in CS_j} \mathbb{E}\mathrm{x}\mathbb{p}\left[S(x_i, x_p)\right] - \sum_{x_p \in W_j} \frac{S(x_i, x_p)}{w_j} > \epsilon\right) \leq 2e^{\left\{\frac{-2w_j\epsilon^2}{\sigma^2}\right\}}.$$

Here $\sigma^2 \geq 0$ parameterizes noise on similarity function SM. Similarly set W_j of size w_j drawn uniformly from class CS_j with $i \neq j$ satisfies:

$$\mathbb{P}\mathrm{r}\mathbb{o}\mathbb{b}\left(\sum_{x_p \in W_j} \frac{S(x_i, x_p)}{w_j} - \min_{x_p \in C_j} \mathbb{E}\mathrm{x}\mathbb{p}\left[S(x_i, x_p)\right] > \epsilon\right) \leq 2e^{\left\{\frac{-2w_j\epsilon^2}{\sigma^2}\right\}}.$$

The condition **P1** highlights similarity from word y_i to its class should have expectation larger than similarity from same word in other class. This relates towards tighter classification condition [36] with lesser stringency than earlier results. The condition **P2** highlights within-and-between-class similarities which concentrate away from each other. This condition is satisfied when similarities remain constant in expectation perturbed with respect to any subgaussian noise.

Considering feature learning d-RNN extracts temporal features for sentiment data sequences. After obtaining sentiment sequence features, fully connected layer fc and softmax layer sm performs classification. This architecture addresses vanishing gradient problem [36]. Network neurons are adopted in last recurrent layer d-RNN$_5$. The first four d-RNN layers use tanh activation function. This is trade-off between improving representation ability and avoiding any over fitting. The number of weights in network is more than in tanh neuron.

The network can be overfitted with limited data training sequences. Specifically, when CS is known and constant across splits in hierarchy, above assumptions are practically violated. This is resolved by fine tuning this algorithm with heuristics. The eigengap is employed where CS is chosen such that eigenvalues gap of Laplacian is large. All subsampled words in data are discarded with low degree when restricted to sample which removes underrepresented sample classes.

In averaging phase if words in data are not similar to any represented class, new word class is developed.

3.5 Approximation with Hierarchical Bidirectional Recurrent Neural Networks

It is very well known that d-RNN can have equivalent structure as stacked RNN when its weight matrices are constrained. If these constraints are ignored, d-RNN peeks at future inputs. It computes delayed output considering time using also inputs which are beyond specified timestep.

An analogous idea has been used as benchmark for bidirectional recurrent neural networks (BRNN) [36, 88]. It has been shown that BRNN are superior to d-RNN considering relatively simple problems. However, it is not clear that this comparison holds true for problems requiring more non-linear solutions. The similar proposition holds for hierarchical bidirectional recurrent neural networks (HBRNN) [36].

If a recurrent network computes its output for specified time by exploiting future input elements, what are necessary conditions in order to approximate its BRNN and HBRNN. Moreover, can d-RNN and Hd-RNN have similar results. And with these conditions, is it more beneficial to use d-RNN and Hd-RNN instead of BRNN and HBRNN. There are number of non-linear transformations [36] where each network applies to any input element prior to computation of output at initial timestep. Only past inputs are processed by generic RNN where number of non-linearities decrease when inputs are close to initial timestep.

BRNN has similar behavior considering causal inputs. For acausal inputs it is symmetrically augmented. For casual inputs Hd-RNN has similar behavior with higher number of non-linearities. This remains for first d acausal inputs with non-linearities decreasing. For Hd-RNN to have at least similar number of non-linearities as BRNN for every sequential element, a delay is required which is twice as sequence length. Hd-RNN can superceed BRNN when non-linear influence of nearby acausal inputs on learned function is superior than farther elements. When Hd-RNN is used in order to approximate BRNN, it also decreases computational cost. For length of considerable sequence stacked BRNN computes both forward and backward RNNs for each layer prior to computation of next layer. As a result of this synchronization parallelization is not allowed which increases runtime. Forward passes for d-RNN takes additional steps, but synchronization does not affect it. In highly parallel hardwares, runtime of k layer stacked BRNN is at least k times slower than d-RNN or Hd-RNN. In línes with d-RNN, Hd-RNN can also be used in critical output values in near real time applications [89, 90].

3.6 Twitter Based Sentiment Analysis Framework for COVID19 in India

In this study a novel framework is presented which uses twitter information in order to understand public behavior in India during COVID19 pandemic. This research addresses sentiments of people in different parts of India during outbreak of pandemic in 2020 and 2021.

Considering datasets defined in section 3.1 our analysis is mainly concentrated in states of Maharashtra, Delhi, Karnataka, Kerala and Tamil Nadu where maximum number of COVID19 active cases are detected. Fig. 5 highlights major components of this framework. It is to be noted that framework is attributed with multi-label classification with multiple outcomes. The social media language has been rapidly evolving. As a result of this special phrases, emotion symbols and abbreviations are present in tweets. These are transformed for building language models [36]. The languages predominant in stated 5 Indian states are Marathi, Hindi, Kannada, Malayalam and Tamil which are used in combination with English. Hence, a transformation is also performed for certain words, emotions and character symbols highlighted in these stated languages.

The computational framework for sentiment analysis involves following steps: (a) tweet extraction, (b) tweet pre-processing, (c) model development with training, and (d) prediction. Tweet extraction involves processing on COVID19 dataset mentioned in section 3.1. During this process special symbols and abbreviations present in tweets are transformed towards building language model. Few such instances are represented in Table 2.

It is to be noted that transformation is performed for specific words, emotions and character symbols which are represented in Marathi, Hindi, Kannada, Malayalam and Tamil.

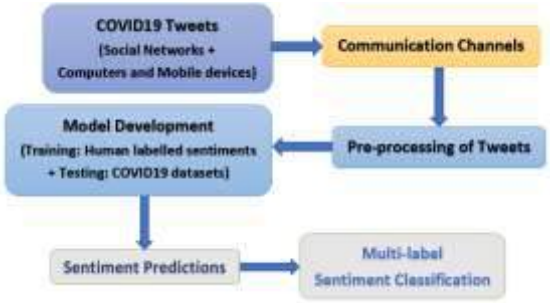


Fig. 5. Twitter based sentiment analysis framework for COVID19 [36]

Table 2. Instances of special symbols and abbreviations present in tweets

Original Phrases	Transformed Word
Facemasks	face masks
Socialdistancing	social distancing
	smiling faces
	beds
	sanitisors
	winks

The experimental dataset features 17 different sentiments [36], which are labelled by a group of 1000 experts for 100,000,000 tweets during COVID19 in 2020 and 2021. In tweet pre-processing each word is converted into its corresponding GLoVe vector.

Here, each word is converted into a 500-dimensional vector. The main reason behind selection of GLoVe embedding involves good results it has shown for sentiment analysis [48].

From each word, GloVe vector is passed towards respective d-RNN and Hd-RNN models for training. The trained models are evaluated initially. After successful evaluation they are used for COVID19 sentiment analysis considering test data from states of Maharashtra, Delhi, Karnataka, Kerala and Tamilnadu.The trained models are used for classification of 17 sentiments.The evaluation metric is so chosen such that it evaluates trained models in best possible way [36].

Hence, it is required that metric captures correct loss [36] arising from misclassification and gives best representative score. As such classification here is of multi-label in nature and is based on binary cross entropy loss, hamming loss, jaccard coefficient score, label ranking average precision score and F1 score [36]. Binary cross entropy loss represents softmax activation alongwith cross entropy loss.

Hamming loss generates bit string of class labels using XOR between actual and predicted labels and averages with respect to dataset instances. Jaccard coefficient score measures overlap between actual and predicted labels with respect to similarity and diversity attributes. Label ranking average precision score calculates percentage of higher ranked labels which resemble true labels considering given samples. F1 score highlights balance between precision and recall measures.

Since classification here is of multi-label in nature, combination of two variants of F1 score viz F1 macro and F1 micro are used towards evaluation of training models.

Considering experimental framework highlighted in Fig. 5 multi-label classification is performed with respect to d-RNN and hierarchical d-RNN models. We use 80:10:10 ratio of dataset

Table 3. Analysis of training performance for d-RNN, Hd-RNN and BERT for COVID19 datasets

Metric used	d-RNN	Hd-RNN	BERT
Binary Cross Entropy Loss	0.279	0.260	0.370
Hamming Loss	0.160	0.155	0.145
Jaccard Score	0.435	0.437	0.530
Label Ranking Precision Score	0.507	0.517	0.779
F1 Macro	0.449	0.460	0.570
F1 Micro	0.500	0.505	0.589

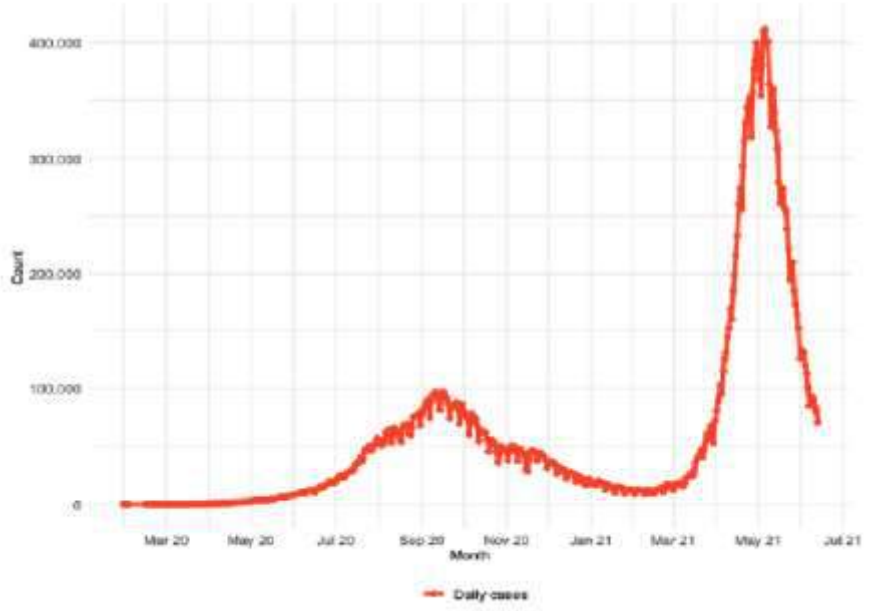


Fig. 6. India with first major peak (mid-September 2020) and second major peak (mid-May 2021) [36]

for training, validation and testing. The aforementioned models are trained using COVID19 dataset [36] stated in section 3.1. This dataset has 100,000,000 tweets collected between March 2020 to June 2021.

In d-RNN and Hd-RNN hyperparameters are determined considering experiments performed. GloVe embedding uses word vector of size 500 in order to provide data representation [48].

A dropout regularization probability of 0.75 is used for d-RNN and hierarchical d-RNN models which feature 500 input units, two layers with 128 and 64 hidden units and an output layer with 17 units for sentiment classification. Here, BERT model is used as benchmark to validate results obtained from d-RNN and hierarchical d-RNN models. The main reason for considering BERT models lies in success obtained by using these models in sentiment analysis [36]. In BERT default hyperparameters are used. For BERT base uncased model learning rate is tuned.

BERT architecture has dropout layer and linear activation layer with 17 outputs corresponding to 17 sentiments. Table 3 shows model training results for 10 experiments with different initial weights and biases for respective models with different performance metrics. In India given huge population with large number of densely populated cities [36, 39] COVID19 management is plagued with major challenges.

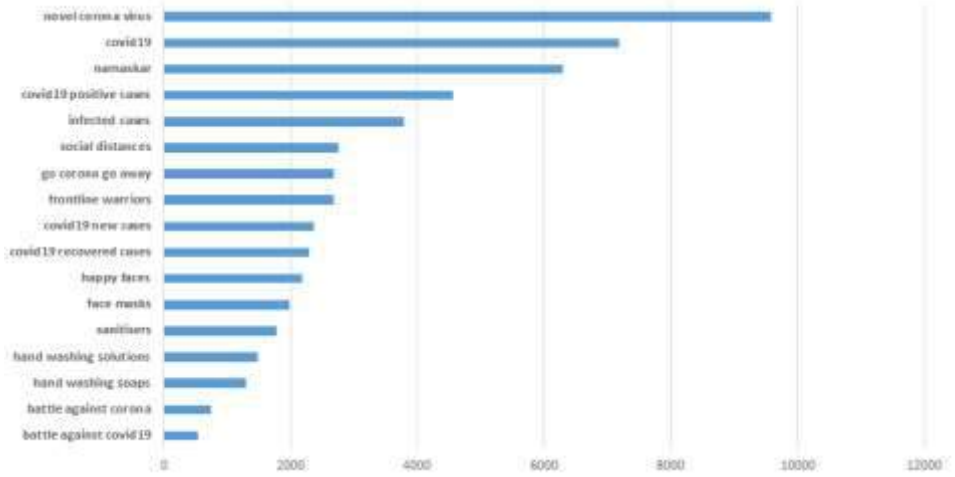


Fig. 7. Bi-grams for cases prevalent in India

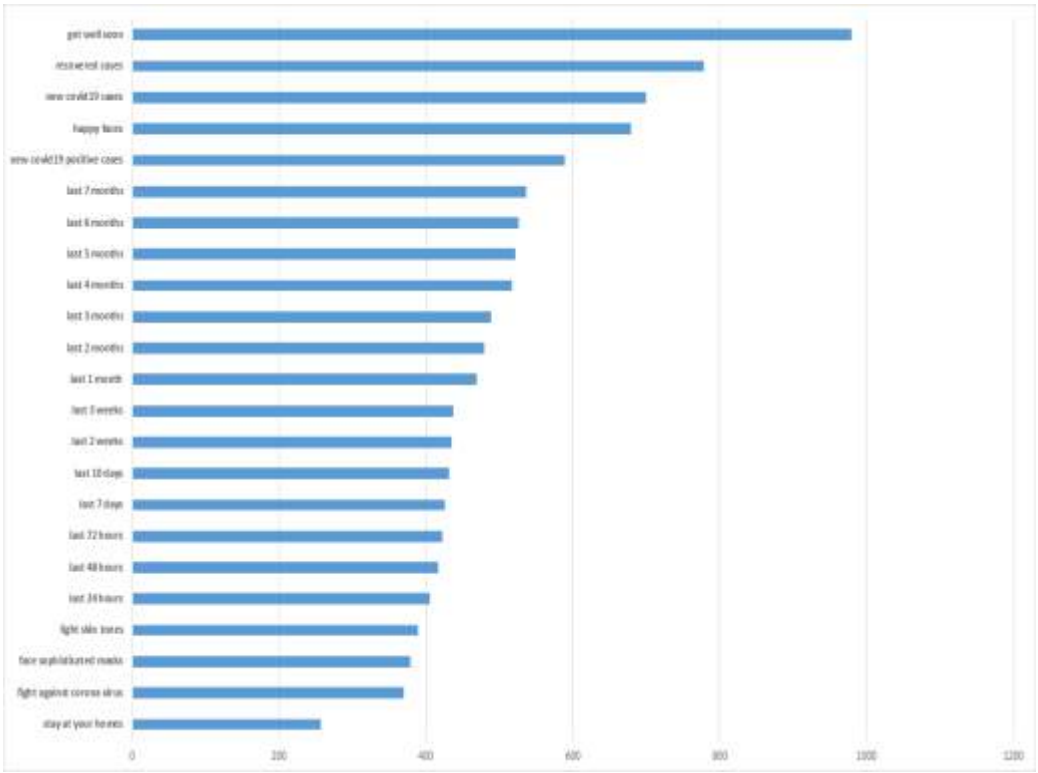


Fig. 8. Tri-grams for cases prevalent in India

The first COVID19 case in India was detected on 30th January 2020. Thereafter India started lockdown as a result of which situation gradually changed. On 22nd March 2021 India had more than 11.6 million confirmed cases with more than 160,000 deaths. This made India third highest with confirmed cases after United States and Brazil. India was 8th in world with more than 300,000

Table 4. Certain situations of tweets which are captured in most prominent bi-grams

Month	Tweets	Bi-gram
May 2020	…….#lockdown times#.....	pointing up
July 2020	……face masks mandatory……	pointing up
September 2020	…..positive cases increasing….	Pointing up
March 2021	……..vaccination necessary…..	with folded hands
April 2021	…….#lockdown times#.....	pointing up
May 2021	…..vaccination mandatory…..	with folded hands
June 2021	……..partial lockdown…….	Pointing up

Table 5. Certain situations of tweets which are captured in most prominent tri-grams

Month	Tweets	Tri-gram
September 2020	…..positive cases increasing….	Pointing up
March 2021	……..vaccination necessary…..	with folded hands
April 2021	…….#lockdown times#.....	pointing up
May 2021	…..vaccination mandatory…..	with folded hands
June 2021	……..partial lockdown…….	Pointing up

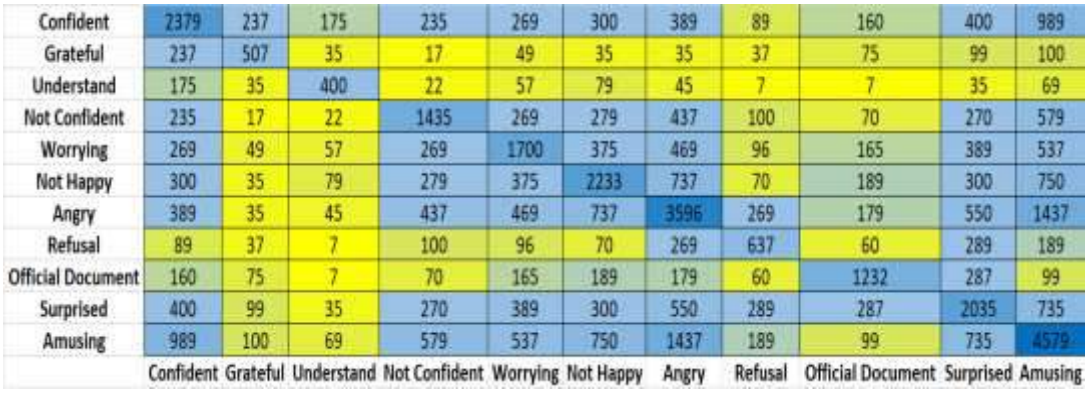


Fig. 9. Heatmap depicting occurrences of given sentiment with respect to remaining sentiments for tweets from training dataset [36]

active cases prior to second wave. India witnessed its first major peak around middle of September 2020 with close to 100,000 cases daily. This gradually decreased to around 11,000 cases daily by end of January 2021. In March 2021 India witnessed its second peak when cases began rising faster and by 22nd March, 2021 India had 47,000 daily new active cases [39].

In first six months, states of Maharashtra, Delhi and Tamil Nadu led COVID19 infections [39] with city of Mumbai having highest number of active cases. In later half of 2021, Delhi cases reduced but still remained in leading 8 states [39]. In 2021, Maharashtra continued as state of highest infections and in March, 2021 it featured more than half of new active cases on a weekly basis. Delhi continued with less than a thousand daily active cases. Alongwith this Karnataka, Kerala and Tamilnadu continued to show good number of daily active cases.

Table 6. Comparative analysis of k-fold cross validation and accuracy of all models [36]

Models	k-fold (1)	k-fold (2)	k-fold (3)	k-fold (4)	k-fold (5)	test level
NB	0.79	0.80	0.82	0.79	0.84	0.84
SVM	0.82	0.84	0.85	0.87	0.87	0.87
kNN	0.75	0.76	0.77	0.78	0.79	0.79
RF	0.79	0.77	0.78	0.80	0.79	0.80
GB	0.84	0.85	0.87	0.84	0.85	0.87
AB	0.85	0.84	0.82	0.84	0.85	0.85
DT	0.77	0.78	0.79	0.80	0.80	0.80
d-RNN	0.95	0.96	0.97	0.96	0.96	0.96
Hd-RNN	0.96	0.97	0.98	0.99	0.97	0.99

Table 7. Comparative analysis of k-fold cross validation and F1 Score of all models [36]

Models	k-fold (1)	k-fold (2)	k-fold (3)	k-fold (4)	k-fold (5)	test level
NB	0.80	0.80	0.82	0.84	0.85	0.85
SVM	0.84	0.85	0.87	0.88	0.89	0.89
kNN	0.77	0.78	0.79	0.80	0.80	0.80
RF	0.80	0.79	0.79	0.82	0.80	0.82
GB	0.85	0.87	0.87	0.88	0.85	0.88
AB	0.87	0.85	0.84	0.85	0.87	0.87
DT	0.79	0.79	0.82	0.82	0.82	0.82
d-RNN	0.96	0.97	0.97	0.97	0.97	0.97
Hd-RNN	0.97	0.98	0.98	0.99	0.99	0.99

Table 8. Comparative analysis of k-fold cross validation and precision of all models [36]

Models	k-fold (1)	k-fold (2)	k-fold (3)	k-fold (4)	k-fold (5)	test level
NB	0.82	0.84	0.84	0.80	0.85	0.85
SVM	0.84	0.85	0.87	0.88	0.89	0.89
kNN	0.77	0.78	0.79	0.80	0.80	0.80
RF	0.82	0.79	0.80	0.82	0.80	0.82
GB	0.85	0.87	0.88	0.85	0.88	0.88
AB	0.87	0.85	0.84	0.85	0.87	0.87
DT	0.79	0.80	0.80	0.80	0.80	0.80
d-RNN	0.96	0.97	0.98	0.97	0.98	0.98
Hd-RNN	0.98	0.98	0.98	0.99	0.99	0.99

Table 9. Comparative analysis of k-fold cross validation and recall of all models [36]

Models	k-fold (1)	k-fold (2)	k-fold (3)	k-fold (4)	k-fold (5)	test level
NB	0.79	0.79	0.80	0.82	0.84	0.84
SVM	0.82	0.84	0.85	0.86	0.88	0.88
kNN	0.76	0.77	0.78	0.79	0.79	0.79
RF	0.79	0.78	0.78	0.80	0.79	0.80
GB	0.84	0.85	0.85	0.87	0.84	0.87
AB	0.85	0.84	0.82	0.84	0.87	0.87
DT	0.78	0.78	0.80	0.80	0.79	0.79
d-RNN	0.95	0.96	0.96	0.96	0.96	0.96
Hd-RNN	0.96	0.97	0.97	0.98	0.98	0.98

Table 10. Comparative analysis of BERT variants, LSTM and BD-LSTM with d-RNN and Hd-RNN models (L: hidden layers, H: hidden size, A: attention heads) [36]

BERT/Other Models	Accuracy	Precision	Recall	F1 Score
L-2 H-128 A-2	0.61	0.60	0.57	0.61
L-2 H-256 A-4	0.82	0.80	0.77	0.79
L-2 H-512 A-8	0.84	0.82	0.85	0.85
L-2 H-768 A-12	0.84	0.80	0.85	0.85
L-4 H-128 A-2	0.80	0.79	0.78	0.80
L-4 H-256 A-4	0.82	0.84	0.85	0.84
L-4 H-512 A-8	0.84	0.82	0.80	0.82
L-4 H-768 A-12	0.87	0.84	0.85	0.84
L-6 H-128 A-2	0.82	0.80	0.84	0.82
L-6 H-256 A-4	0.82	0.84	0.82	0.82
L-6 H-512 A-8	0.85	0.82	0.82	0.82
L-6 H-768 A-12	0.87	0.85	0.84	0.84
L-8 H-128 A-2	0.80	0.80	0.82	0.82
L-8 H-256 A-4	0.82	0.84	0.82	0.82
L-8 H-512 A-8	0.85	0.82	0.82	0.82
L-10 H-128 A-2	0.82	0.80	0.84	0.82
L-10 H-256 A-4	0.84	0.80	0.85	0.85
L-10 H-512 A-8	0.84	0.82	0.84	0.85
L-12 H 128 A-2	0.82	0.80	0.87	0.87
L-12 H-256 A-4	0.85	0.80	0.80	0.85
L-12 H-512 A-8	0.89	0.84	0.89	0.89
LSTM	0.90	0.93	0.94	0.95
BD-LSTM	0.93	0.94	0.95	0.96
d-RNN	0.96	0.95	0.97	0.97
Hd-RNN	0.99	0.96	0.95	0.99

We have applied COVID19 datasets from different parts of India including nationwide COVID19 sentiments with 5 major states having majority of COVID19 cases.

The trends have shown that these 5 states had two major peaks followed by several minor ones. However, India had first major peak around mid-September, 2020 and second major peak around mid-May, 2021 as shown in Fig. 6 [36].

4 Experiments and Results

Considering computational methodology in section 3, in this section experiments and results are presented.

4.1 Analysis of COVID19 Results

The test dataset [36, 39] contains COVID19 tweets between March 2020 to June 2021.

It comprised of more than 750,000 tweets from India. Five more datasets are generated considering states of Maharashtra, Delhi, Karnataka, Kerala and Tamilnadu with around 22,000 tweets each.

It is observed that number of tweets in India follows identical trend as number of COVID19 cases increase till July, 2020 after which number of tweets decline.

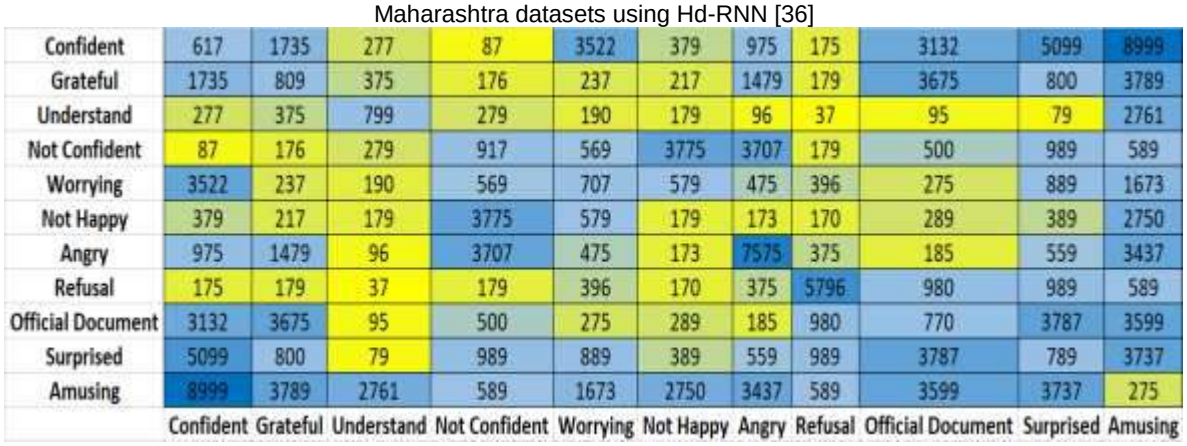
Again, number of COVID19 cases increase till April, 2021 after which number of tweets decline. There is similar pattern for Maharashtra and Kerala.

	Confident	Grateful	Understand	Not Confident	Worrying	Not Happy	Angry	Refusal	Official Document	Surprised	Amusing
Confident	55217	17235	2737	875	3522	375	4975	175	3132	5095	8686
Grateful	17235	17809	375	175	237	217	1479	79	1675	800	1789
Understand	2737	375	3799	219	190	179	96	7	9	7	61
Not Confident	875	175	219	7917	3569	1775	3707	175	170	289	589
Worrying	3522	237	190	3569	23707	379	475	196	175	489	673
Not Happy	375	217	179	1775	379	10179	1737	170	289	389	2750
Angry	4975	1479	96	3707	475	1737	45975	275	187	559	3437
Refusal	175	79	7	175	196	170	275	5596	280	389	489
Official Document	3132	1675	9	170	175	289	187	280	21770	387	599
Surprised	5095	800	7	289	489	389	559	389	387	37789	3735
Amusing	8686	1789	61	589	673	2750	3437	489	599	3735	37275

Fig. 10. Heatmap showing number of occurrence of given sentiment with respect to remaining sentiments for India datasets using Hd-RNN [36]

	Confident	Grateful	Understand	Not Confident	Worrying	Not Happy	Angry	Refusal	Official Document	Surprised	Amusing
Confident	5528	7237	2737	8735	3523	475	4979	275	3135	7095	8989
Grateful	7235	17809	2375	179	269	228	1979	179	2875	889	3789
Understand	2737	2375	3779	319	190	179	96	37	79	89	261
Not Confident	8735	179	319	8917	5569	1775	3709	275	179	3289	3589
Worrying	3523	269	190	5569	28707	879	475	196	275	3789	1673
Not Happy	475	228	179	1775	879	179	1937	170	289	489	3750
Angry	4979	1979	96	3709	475	1937	975	375	89	959	479
Refusal	275	179	37	275	196	170	375	559	599	389	489
Official Document	3135	2875	79	179	275	289	89	599	2770	987	3599
Surprised	7095	889	89	3289	3789	489	959	389	987	3789	3775
Amusing	8686	3789	261	3589	1673	3750	479	489	3599	3775	3779

Fig. 11. Heatmap showing number of occurrence of given sentiment with respect to remaining sentiments for Maharashtra datasets using Hd-RNN [36]



	Confident	Grateful	Understand	Not Confident	Worrying	Not Happy	Angry	Refusal	Official Document	Surprised	Amusing
Confident	617	1735	277	87	3522	379	975	175	3132	5099	8999
Grateful	1735	809	375	176	237	217	1479	179	3675	800	3789
Understand	277	375	799	279	190	179	96	37	95	79	2761
Not Confident	87	176	279	917	569	3775	3707	179	500	989	589
Worrying	3522	237	190	569	707	579	475	396	275	889	1673
Not Happy	379	217	179	3775	579	179	173	170	289	389	2750
Angry	975	1479	96	3707	475	173	7575	375	185	559	3437
Refusal	175	179	37	179	396	170	375	5796	980	989	589
Official Document	3132	3675	95	500	275	289	185	980	770	3787	3599
Surprised	5099	800	79	989	889	389	559	989	3787	789	3737
Amusing	8999	3789	2761	589	1673	2750	3437	589	3599	3737	275

Fig. 12. Heatmap showing number of occurrence of given sentiment with respect to remaining sentiments for Delhi datasets using Hd-RNN [36]

For Delhi, Karnataka and Tamilnadu situation is slightly different as first peak was observed in July, 2020 with increasing tweets that declined afterwards and did not keep up with second peak

of cases in September, 2020. Again, number of COVID19 cases increase till April, 2021 after which number of tweets decline. This indicates that as more cases were observed in early months, there was much concern which eased before major peak was reached and number of tweets were drastically decreased. There could be signs of fear, depression and anxiety as tweets decreased drastically after July, 2020 with increasing cases. Fig. 7 and Fig. 8 show number of bi-grams and tri-grams prevalent in India. In bi-grams it is observed that novel corona virus is most used followed by covid19. Similarly, namaskar is most used followed by covid19 positive cases and infected cases. In order to provide better understanding of tweets some examples are highlighted as shown in Table 4. In case of tri-grams we can find more information in tweets as shown in Table 5.

Fig. 9 shows number of occurrences of a given sentiment in relation to rest of tweets sentiments in training datasets [36, 39]. Now we present results of COVID19 tweets prediction in India, Maharashtra, Delhi, Karnataka, Kerala and Tamilnadu by considering them as individual datasets. The dataset mentioned in Section 3.1 has been used here for training. Fig. 10 above presents distribution of sentiments predicted by Hd-RNN for respective datasets considering stated span of time. Hd-RNN has provided best results for training data. In Fig. 11 sentiments predicted are reviewed considering heatmap in order to examine number of occurrences of given sentiment with respect to rest sentiments. These heatmaps indicate how two sentiments have been expressed and provides more insights regarding positive and negative sentiments. Fig. 12 provides visualisation of tweets distribution with number of combination sentiments.

4.2 Validation of COVID19 Results

After an initial analysis of results in previous section, we present further validation of obtained results here. In order to achieve this cross validation is used. This is a resampling method which evaluates developed models on small data samples. The given data samples are divided into number of groups such that we have k-fold cross validation. It is used in order to estimate model's ability on unknown data. It provides lesser degree of biasedness or positive estimation considering model's ability in comparison to other approaches. As such it considers simple training and testing splits.

The success of k-fold cross validation lies in choosing a set of split numbers instead of any specific split number. This helps in checking acceptability of dataset as well as addresses issues related to skewness in datasets.

Since COVID19 datasets have an uneven class distribution, most intuitive performance metric to be used here is precision. Closely associated with precision is recall performance metric which also works well here as well as F1 Score which is weighted average of precision and recall. Another important performance metric which could have been used here is accuracy but because datasets are not symmetric, it will work well here.

In order to make our experimental hypothesis stronger we now present comparative performance analysis of d-RNN and Hd-RNN models with some traditional machine learning models such as NB, SVM, kNN, RF, GB, AB and DT. Table 6 highlights k-fold cross validation and accuracy comparison of all these models. Table 7 presents k-fold cross validation and F1 Score comparison of all these models. Table 8 shows k-fold cross validation and precision comparison of all these models. Table 9 shows k-fold cross validation and recall comparison of all these models.

As BERT model has been used here for evaluating test datasets with prediction validation, we use different variants of BERT [36] as well as LSTM [36] and BD-LSTM [36] models as shown in Table 10 in order to compare performance of d-RNN and Hd-RNN models.

5 Conclusion

In this work we have presented sentiment analysis of COVID19 infections with d-RNN and Hd-RNN as major computational models. The time span considered is from January 2020 to June 2021. We considered tweets from different regions of India. We reviewed tweets from specific regions such as Maharashtra, Delhi, Karnataka, Kerala and Tamilnadu. The models are trained with COVID19 datasets hand labelled tweets. The majority of tweets have highlighted optimism, fear and

uncertainty during infections of COVID19 cases in India. There has been variability in number of tweets during peak of new cases.

The predictions indicate that although majority of population have been optimistic, a significant group has been disturbed by way pandemic was handled by Indian government.

This computational framework can be used for better COVID19 management in order to support cases of depression and mental health issues. The experimental results are validated considering various traditional machine learning as well as different variants of BERT models. The results with d-RNN and Hd-RNN highlight superiority of proposed methods.

The computational model can be used for different regions, countries, ethnic and social groups. This can also be extended to understand reactions towards vaccinations with rise of antivaccine sentiments given fear, insecurity and unpredictability of COVID19 situations. This computational framework incorporates topic modelling with sentiment analysis which provides more details during COVID19 cases with respect to various government rules and regulations. As concluding remarks we would like to mention that present Indian government has been unsuccessful in addressing stress and strain through which country's economy is passing through.

References

1. **Gorbalenya, A.E., Baker, S.C., Baric, R.S., De Groot, R.J., Drosten, C., Gulyaeva, A.A., Haagmans, B.L., Lauber, C., Leontovich, A.M., Neuman, B.W., Penzar, D., Perlman, S., Poon, L.L.M., Samborskiy, D.V., Sidorov, I.A., Sola, I., Ziebuhr, J. (2020).** The species severe acute respiratory syndrome-related coronavirus: classifying 2019-nCoV and naming it SARS-CoV-2. Nature Microbiology, Vol. 5, No. 4, pp. 536–544. DOI: 10.1038/s41564-020-0695-z.
2. **Monteil, V., Kwon, H., Prado, P., Hagelkrüys, A., Wimmer, R.A., Stahl, M., Leopoldi, A., Garreta, E., Hurtado del Pozo C., Prosper, F., Romero J.P., Wirnsberger, G., Zhang, H., Slutsky, A.S., Conder, R., Montserrat, N., Mirazimi, A., Penninger, J.M. (2020).** Inhibition of SARSCoV-2 infections in engineered human tissues using clinical-grade soluble human ACE2. Cell, Vol. 181, No. 4, pp. 905–913. DOI: 10.1016/j.cell.2020.04.004.
3. **WHO. (2020).** Coronavirus disease 2019 (COVID-19): Situation report, 72. World Health Organization, April, 1, 2020.
4. **Cucinotta, D., Vanelli, M. (2020).** WHO declares COVID-19 a pandemic. Acta Bio-medica: Atenei Parmensis, Vol. 91, No. 1, pp. 157–160. DOI: 10.23750/abm.v91i1.9397.
5. **Wikipedia.** COVID19.
6. **ILO, FAO, IFAD, WHO. (2020).** Impact of COVID-19 on people's livelihoods, their health and our food systems. World Health Organization (WHO).
7. **Siche, R. (2020).** What is the impact of COVID-19 disease on agriculture? Scientia Agropecuaria, Vol. 11, No. 1, pp. 3–6. DOI: 10.17268/sci.agropecu.2020.01.00.
8. **Richards, M., Anderson, M., Carter, P., Ebert, B. L., Mossialos, E. (2020).** The impact of the COVID-19 pandemic on cancer care. Nature Cancer, Vol. 1, No. 6, pp. 565–567. DOI: 10.1038/ s43018-020-0074-y.
9. **Tiwari, A., Gupta, R., Chandra, R. (2021).** Delhi air quality prediction using LSTM deep learning models with a focus on COVID-19 lockdown. DOI: 10.48550/arXiv.2102.10551.
10. **Upadhyay, A. (2021).** Impact of Covid-19 on Indian economy. The Times of India.
11. **Golbeck, J., Robles, C., Edmondson, M., Turner, K. (2011).** Predicting personality from twitter. IEEE 3rd international conference on privacy, security, risk and trust and IEEE 3rd international conference on social computing, pp. 149–156. DOI: 10.1109/PASSAT/SocialCom.2011.33.
12. **Quercia, D., Kosinski, M., Stillwell, D., Crowcroft, J. (2011).** Our twitter profiles, our selves: Predicting personality with twitter. IEEE 3rd international conference on privacy, security, risk and trust and IEEE 3rd international conference on social computing, pp. 180–185. DOI: 10.1109/PASSAT/SocialCom.2011.26.
13. **Bittermann, A., Batzdorfer, V., Müller, S.M., Steinmetz, H. (2021).** Mining twitter to detect hotspots in psychology. Zeitschrift für Psychologie, Vol. 229, No. 1, pp. 3–14. DOI: 10.1027/2151-2604/a000437.
14. **Lin J. (2015).** On building better mousetraps and understanding the human condition: Reflections on big data in the social sciences. The ANNALS of the American Academy of Political and Social Science, Vol. 659, No. 1, pp. 33–47. DOI: 10.1177/0002716215569174.

15. **Coppersmith, G., Dredze, M., Harman, C. (2014).** Quantifying mental health signals in Twitter. Proceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality, pp. 51–60.
16. **Murphy, S.C. (2017).** A hands-on guide to conducting psychological research on Twitter. Social Psychological and Personality Science, Vol. 8, No. 4, pp. 396–412. DOI: 10.1177/1948550617697178.
17. **Zhou, Y., Na, J.C. (2019).** A comparative analysis of Twitter users who tweeted on psychology and political science journal articles. Online Information Review, Vol. 43, No. 7, pp. 1188–1208. DOI: 10.1108/OIR-03-2019-0097.
18. **Wang, W., Hernandez, I., Newman, D.A., He, J., Bian, J. (2016).** Twitter analysis: Studying US weekly trends in work stress and emotion. Applied Psychology, Vol. 65, No. 2, pp. 355–378. DOI: 10.1111/apps.12065.
19. **Manning, C., Schutze, H. (1999).** Foundations of statistical natural language processing. MIT Press.
20. **Chaudhuri, A., Ghosh, S.K. (2016).** Sentiment analysis of customer reviews using robust hierarchical bidirectional recurrent neural network. Artificial Intelligence Perspectives in Intelligent Systems, Springer, Cham, Vol. 464, pp. 249–261. DOI: 10.1007/978-3-319-33625-1_23.
21. **Chaudhuri, A. (2019).** Visual and text sentiment analysis through hierarchical deep learning networks. Springer Briefs in Computer Science, Springer. pp. 1–98.
22. **Liu, B., Zhang, L. (2012).** A survey of opinion mining and sentiment analysis. Mining Text Data, Springer, pp. 415–463. DOI: 10.1007/978-1-4614-3223-4_13.
23. **Medhat, W., Hassan, A., Korashy, H. (2014).** Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, Vol. 5, No. 4, pp. 1093–1113. DOI: 10.1016/j.asej.2014.04.011.
24. **Hussein, D. (2018).** A survey on sentiment analysis challenges. Journal of King Saud University – Engineering Sciences, Vol. 30, No. 4, pp. 330–338. DOI: 10.1016/j.jksues.2016.04.002.
25. **Beigi, G., Hu, X., Maciejewski, R., Liu, H. (2016).** An overview of sentiment analysis in social media and its applications in disaster relief. Sentiment Analysis and Ontology Engineering, Springer, Cham, Vol. 639, pp. 313–340. DOI: 10.1007/978-3-319-30319-2_13.
26. **Drus, Z., Khalid, H. (2019).** Sentiment analysis in social media and its application: Systematic literature review. Procedia Computer Science, Vol. 161, pp. 707–714. DOI: 10.1016/j.procs.2019.11.174.
27. **Kolenik, T., Gams, M. (2021).** Intelligent cognitive assistants for attitude and behavior change support in mental health: State-of-the-art technical review. Electronics, Vol. 10, No. 11, pp. 1–34. DOI: 10.3390/electronics10111250.
28. **Alhijawi, B., Awajan, A. (2021).** Prediction of movie success using twitter temporal mining. Proceedings of 6th International Congress on Information and Communication Technology, Lecture Notes in Networks and Systems, Springer, Vol. 235, pp. 105–116. DOI: 10.1007/978-981-16-2377-6_12.
29. **Kolenik, T., Gams, M. (2021).** Persuasive technology for mental health: One step closer to (Mental Health Care) equality? IEEE Technology and Society Magazine, Vol. 40, No. 1, pp. 80–86. DOI: 10.1109/MTS.2021.3056288.
30. **Biltawi, M., Etaiwi, W., Tedmori, S., Hudaib, A., Awajan, A. (2016).** Sentiment classification techniques for Arabic language: A survey. 7th International Conference on Information and Communication Systems (ICICS), pp. 339–346. DOI: 10.1109/IACS.2016.7476075.
31. **Fang, X., Zhan, J. (2015).** Sentiment analysis using product review data. Journal of Big Data, Vol. 2, No. 5, pp. 1–14. DOI: https://doi.org/10.1186/s40537-015-0015-2.
32. **Haykin, S. (2008).** Neural networks and learning machines.3rd ed. Prentice Hall. pp. 1–906
33. **Moreno, L.M., Kalita, J. (2017).** Deep Learning applied to NLP. arXiv e-prints DOI: 10.48550/arXiv.1703.03091.
34. **Deng, L., Yu, D. (2014).** Deep Learning: Methods and Applications. Foundations and Trends in Signal Processing, Publishers Inc., Vol. 7, No. 3–4, pp. 197–387. DOI: 10.1561/2000000039.
35. **Young, T., Hazarika, D., Poria, S., Cambria, E. (2018).** Recent trends in deep learning based natural language processing. IEEE Computational Intelligence Magazine, Vol. 13, No. 3, pp. 55–75. DOI: 10.1109/MCI.2018.2840738.
36. **Chaudhuri, A. (2021).** Sentiment analysis on COVID19 data in India using advanced machine learning methods. Samsung R & D Institute Delhi, India, Technical Report, TR-8989.
37. **Gholizadeh, S., Zhou, N. (2021).** Model explainability in deep learning based natural language. Processing. arXiv:2106.07410:2106.07410. DOI: 10.48550/arXiv.2106.07410.
38. **Deng, L., Liu, Y. (2018).** Deep learning in natural language processing. Springer. pp. 1–327

39. **Chaudhuri, A., Ghosh, S.K. (2022).** COVID19 forecasting in India through deep learning models. Recent Advances in AI-enabled Automated Medical Diagnosis. Taylor and Francis [in press].
40. **Li, Y., Yang, T. (2018).** Word embedding for understanding natural language: A survey. **Srinivasan, S. (ed)** Guide to Big Data Applications, Springer, Cham, Vol. 26, pp. 83–104. DOI: 10.1007/978-3-319-53817-4_4.
41. **Kutuzov, A., Øvrelid, L., Szymanski, T., Velldal, E. (2018).** Diachronic word embeddings and semantic shifts: A survey. Proceedings of COLING 2018, DOI: 10.48550/arXiv.1806.03537.
42. **Ruder, S., Vulić, I., Søgaard, A. (207).** A survey of cross-lingual word embedding models. Journal of Artificial Intelligence Research, DOI: 10.48550/arXiv.1706.04902.
43. **Zhang, Y., Jin, R., Zhou, Z.H. (2010).** Understanding bag-of-words model: A statistical framework. International Journal of Machine Learning and Cybernetics, Vol. 1, No. 1, pp. 43–52. DOI: 10.1007/s13042-010-0001-0.
44. **Ramos, J. (2003).** Using TF-IFD to determine word relevance in document queries. 1st Instructional Conference on Machine Learning, Vol. 242, No. 1, pp. 29–48.
45. **Goodman, J.T. (2001).** A bit of progress in language modeling. Computer Speech & Language, Vol. 15, No. 4, pp. 403–434. DOI: 10.1006/csla.2001.0174.
46. **Guthrie, D., Allison, B., Liu, W., Guthrie, L., Wilks, Y. (2006).** A closer look at skip-gram modelling. 5th International Conference on Language Resources and Evaluation, pp. 1222–1225.
47. **Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. (2013).** Distributed representations of words and phrases and their compositionality. Computation and Language, DOI: 10.48550/arXiv.1310.4546.
48. **Pennington, J., Socher, R., Manning, C.D. (2014).** GloVe: Global vectors for word representation. Empirical Methods in Natural Language Processing, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.
49. **Zhao, J., Zhou, Y., Li. Z., Wang, W., Chang, K.W. (2018).** Learning gender-neutral word embeddings. EMNLP´18, DOI: 10.48550/arXiv.1809.01496.
50. **Ghannay, S., Estève, Y., Camelin, N., Deléglise, P. (2016).** Evaluation of acoustic word embeddings. 1st Workshop on Evaluating Vector-Space Representations for NLP, pp. 62–66.
51. **Schwenk, H. (2007).** Continuous space language models. Computer Speech & Language, Vol. 21, No. 3, pp. 492–518. DOI: 10.1016/j.csl. 2006.09.003.
52. **Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., Kingsbury, P., Liu, H. (2018).** A comparison of word embeddings for the biomedical natural language processing. Journal of Biomedical Informatics, Vol. 87, pp. 12–20. DOI: 10.1016/j.jbi.2018.09.008.
53. **Naresh, A., Venkata Krishna, P. (2021).** An efficient approach for sentiment analysis using machine learning algorithm. Evolutionary Intelligence, Vol. 14, No. 2, pp. 725–731. DOI: 10.1007/s12065-020-00429-1.
54. **Kawade, D.R., Oza, K.S. (2017).** Sentiment analysis: Machine learning approach. International Journal of Engineering and Technology, Vol. 9, No. 3, pp. 2183–2186. DOI: 10.21817/ijet/2017/ v9i3/1709030151.
55. **Chaturvedi, S., Mishra, V., Mishra, N. (2017).** Sentiment analysis using machine learning for business intelligence. IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, pp. 2162–2166. DOI: 10.1109/ICPCSI.2017.8392100.
56. **Shathik, A., Karani, K.P. (2020).** A literature review on application of sentiment analysis using machine learning techniques. International Journal of Applied Engineering and Management Letters, Vol. 4, No. 2, pp. 41–77. DOI: 10.5281/zenodo.3977576.
57. **Singh, J., Singh, G., Singh, R. (2017).** Optimization of sentiment analysis using machine learning classifiers. Human-centric Computing and Information Sciences, Vol. 7, No. 32, pp. 1–12. DOI: 10.1186/s13673-017-0116-3.
58. **Raza, H., Faizan, M., Hamza, A., Mushtaq, A., Akhtar, N. (2019).** Scientific text sentiment analysis using machine learning techniques. International Journal of Advanced Computer Science and Applications, Vol. 10, No. 12, pp. 157–165.
59. **De Figueiredo, A., Simas, C., Karafillakis, E., Paterson, P., Larson H. (2020)**. Mapping global trends in vaccine confidence and investigating barriers to vaccine uptake: a large-scale retrospective temporal modelling study. The Lancet, Vol. 396, No. 10255, pp. 898–908. DOI: 10.1016/S0140-6736(20)31558-0.
60. **Hussain, A., Tahir, A., Hussain, Z., Sheikh, Z., Gogate, M., Dashtipour, Ali. A, Sheikh, A. (2021).** Artificial intelligence-enabled analysis of public attitudes on facebook and twitter towards COVID-19 vaccines in the United Kingdom and the United States: Observational Study. Journal of Medical Internet Research, Vol. 23, No. 4, pp. 1–10. DOI: 10.2196/26627.

61. **Hung, M., Lauren, E., Hon, E.S., Birmingham, W.C., Xu, J., Su, S., Hon, S.D., Park, J., Dang, P., Lipsky, M. (2020).** Social network analysis of COVID-19 sentiments: Application of Artificial Intelligence. Journal of Medical Internet Research, Vol. 22, No. 8, pp. 1–13. DOI: 10.2196/22590.
62. **Sarlan, A., Nadam, C., Basri, S. (2014).** Twitter sentiment analysis. 6th International Conference on Information Technology and Multimedia, pp. 212–216. DOI: 10.1109/ICIMU.2014.7066632.
63. **Alamoodi, A.H., Zaidan, B.B., Zaidan, A.A, Albahri, O.S., Mohammed, K.I., Malik, R.Q., Almahdi, E.M., Chyad, M.A., Tareq, Z., Albahri, A.S., Hameed, H., Alaa, M. (2021).** Sentiment analysis and its applications in fighting COVID-19 and infectious diseases: A systematic review. Expert Systems with Applications, Vol. 167, pp. 1–13. DOI: 10.1016/j.eswa.2020.114155.
64. **Samuel, J., Ali, G.G., Rahman, M., Esawi, E., Samuel, Y. (2020).** COVID-19 public sentiment insights and machine learning for tweets classification. Information, Vol. 11, No. 6. DOI: 10.3390/info11060314.
65. **Hochreiter, S., Schmidhuber, J. (1997).** Long short-term memory. Neural Computation, Vol. 9, No. 8, pp. 1735–1780. DOI: 10.1162/neco.1997 .9.8.1735.
66. **Devlin, J., Chang, M.W., Lee, K., Toutanova, K. (2018).** BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv 1810.04805. DOI: 10.48550/arXiv. 1810.04805.
67. **Omlin, C.W., Giles, C.L. (1996).** Constructing deterministic finite-state automata in recurrent neural networks. Journal of ACM, Vol. 43, No. 6, pp. 937–972. DOI: 10.1145/235809.235811.
68. **Omlin, C.W., Giles, C.L. (1992).** Training second-order recurrent neural networks using hints. 9th International Conference on Machine Learning, pp. 361–366. DOI: 10.1016/B978-1-55860-247-2.50051-6.
69. **Schmidhuber, J. (2015).** Deep learning in neural networks: An overview. Neural Networks, Vol. 61, pp. 85–117. DOI: 10.1016/j.neunet.2014.09.003.
70. **Schuster, M., Paliwal, K.K. (1997).** Bidirectional Recurrent Neural Networks. IEEE Transactions on Signal Processing, Vol. 45, No. 11, pp. 2673–2681. DOI: 10.1109/78.650093.
71. **Altaher, A. (2017).** Hybrid approach for sentiment analysis of Arabic tweets based on deep learning model and features weighting. International Journal of Advanced and Applied Sciences, Vol. 4, No. 8, pp. 43–49. DOI: 10.21833/ijaas.2017.08.007.
72. **Gao, Y., Rong, W., Shen, Y., Xiong, Z. (2016).** Convolutional neural network-based sentiment analysis using adaboost combination. International Joint Conference on Neural Networks, pp. 1333–1338. DOI: 10.1109/IJCNN.2016.7727352.
73. **Cai, G., Xia, B. (2015).** Convolutional neural networks for multimedia sentiment analysis. Natural Language Processing and Chinese Computing, Lecture Notes in Computer Science, Vol. 9362, pp. 159–167. DOI: 10.1007/978-3-319-25207-0_14.
74. **Rani, S., Kumar, P. (2018).** Deep learning based sentiment analysis using convolution neural networks. Arabian Journal for Science and Engineering, Vol. 44, No. 4, pp. 3305–3314. DOI: 10.1007/s13369-018-3500-z.
75. **Kumar, A., Srinivasan, K., Cheng, W.H., Zomaya, A.Y. (2020).** Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data. Information Processing and Management, Vol. 57, No. 1, pp. 102–141. DOI: 10.1016/j.ipm.2019.102141.
76. **Kapociute-Dzikiene, J., Damaševičius, R., Woźniak, M. (2019).** Sentiment Analysis of Lithuanian texts using traditional and deep learning approaches, Computers, Vol. 8, No. 4, pp. 1–16, DOI: 10.3390/computers8010004.
77. **Baktha, K., Tripathy, B.K. (2017).** Investigation of recurrent neural networks in the field of sentiment analysis. International Conference on Communication and Signal Processing, pp. 2047–2050. DOI: 10.1109/ICCSP.2017.8286763.
78. **Shijia, E., Yang, L., Zhang, M., Xiang, Y. (2018).** Aspect based financial sentiment analysis with deep neural networks. The Web Conference, pp. 1951–1954. DOI: 10.1145/3184558.3191825.
79. **Piao, G., Breslin, J.G. (2018).** Financial aspect and sentiment predictions with deep neural networks: An ensemble approach. The Web Conference, pp. 1973–1977. DOI: 10.1145/3184558.3191829.
80. **Chen, L.C., Lee, C.M., Chen, M.Y. (2019).** Exploration of social media for sentiment analysis using deep learning. Soft Computing, Vol. 24, No. 11, pp. 8187–8197. DOI: 10.1007/s00500-019-04402-8.
81. **Ghulam, H., Zeng, F., Li, W., Xiao, Y. (2019).** Deep learning-based sentiment analysis for Roman Urdu text. Procedia Computer Science, Vol. 147, pp. 131–135. DOI: 10.1016/j.procs.2019.01.202.
82. **Tang, D. (2015).** Sentiment-specific representation learning for document-level sentiment analysis. 8th ACM International Conference on Web Search and Data Mining, pp. 447–452. DOI: 10.1145/2684822.2697035.

83. Chakraborty, K., Bhatia, S., Bhattacharyya, S., Platos, J., Bag, R., Hassanien, A.E. (2020). Sentiment analysis of COVID-19 tweets by deep learning classifiers — A study to show how popularity is affecting accuracy in social media. Applied Soft Computing, Vol. 97, pp. 1–14. DOI: 10.1016/j.asoc.2020.106754.

84. Al Sallab, A., Baly, R., Badaro, G., Hajj, H., El Hajj, W., Shaban, K.B. (2015). Deep learning models for sentiment analysis in Arabic. 2nd Workshop on Arabic Natural Language Processing, pp. 9–17.

85. Heaton, J. (2015). Deep learning and neural networks. Artificial Intelligence for Humans, Vol. 3. Heaton Research, Inc.

86. Chaudhuri, A. (2015). Semantic analysis of customer reviews with machine learning methods. Samsung R & D Institute, Delhi, India, Technical Report, TR-3699.

87. Turek, J., Jain, S., Vo, V., Capota, M., Huth, A., Willke, T. (2019). Approximating stacked and bidirectional recurrent architectures with the delayed recurrent neural network. Proceedings of Machine Learning Research, DOI: 10.48550/arXiv.1909.00021.

88. Graves, A., Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, Vol. 18, No. 5-6, pp. 602–610. DOI: 10.1016/j.neunet.2005.06.042.

89. Guo, T., Xu, Z., Yao, X., Chen, H., Aberer, K., Funaya, K. (2016). Robust online time series prediction with recurrent neural networks. IEEE International Conference on Data Science and Advanced Analytics, pp. 816–825. DOI: 10.1109/DSAA.2016.92.

90. Arik, S.O., Chrzanowski, M., Coates, A., Diamos, G., Gibiansky, A., Kang, Y., Li, X., Miller, J., Ng, A., Raiman, J., Sengupta, S., Shoeybi, M. (2017). Deep voice: Real-time neural text-to-speech. 34th International Conference on Machine Learning, Vol. 70, pp. 195–204.

Article received on 17/02/2022; accepted on 27/03/2022.
Corresponding author is Arindam Chaudhuri.

ISSN 2007-9737

Selection of the Decision Variables for the Habanero Chili Peppers (Capsicum chinense Jacq.) Using Machine Learning

Blanca C. López-Ramírez[1], Francisco Chablé-Moreno[2], Francisco Cervantes-Ortiz[2]

[1] Tecnológico Nacional de Mexico/IT.Roque,
Department of Systems and Computing,
Mexico

[2] Tecnológico Nacional de Mexico/IT.Roque,
Department of Agricultural Sciences,
Mexico

{blanca.lr, francisco.cm, francisco.co}@roque.tecnm.mx

Abstract. Data science is an area that allows a gathering of data from several prospects, being at once, collaborative and multidisciplinary. It is an area so promising and open to research from different problems, including the challenges of agronomy science throughout the study of the exploitation of database knowledge. In this work, we will study if it is possible to identify some determined variable that could allow to response to the questions, ¿Is it possible to know the genotype from a habanero pepper plant, knowing some plant measure? also, ¿Is it possible to identify the yield through the plant height? The goal is to identify the proficiency of each one of the involved areas on the preparation, processing, and database, as the necessary methods and tools to gather relevant information to the expertise; derivable from techniques as Neural Networks Algorithms and Statistics. The outcome earned, prove even tho the statistics operations revealed results by a descriptive category besides a predictive one; The Neural Networks Algorithms find results within the prescriptive category, displayed on work and that represent a very interesting answer resulting from applying questions that were not obviously in basic analysis.

Keywords. Data analytics, rescriptive analysis, neural networks algorithms, post-decision, state variable.

1 Introduction

This work describes the study of the behavior of growing variables from 8 habanero chili genotype(Capsicum chinense Jacq.) through machine learning. The objective is to find the existence of a relationship between the phenomenological development variables and yield. It is interesting to know if the yield is directly related to the plant height or if the leaf growth is related to the regional weather.

Gathering this information, different genotypes had been cultivated under the same weather, and measuring and data collection too. The information that this research presents, not only implies raw data of variables but also presents a descriptive and prescriptive data analysis, applying data science techniques as statistics and neural networks.

Agricultural production has had 3 stages; the first one has been when the labor was rich and intense until 1920, later, the second stage when the industrial revolution provide heavy machine to the fieldwork and the seed study to improve farming, finally, the third one began since 2010 until today, which it is known as the Agriculture 3.0, this stage is considered by the use of innovative technology for different studies that allow a genetic and phenologic improvement of farms assisting to take decisions based on data analysis and data obtained from external sources.

The good decisions based on agricultural information give higher productivity, practicing

sustainability even helping to provide transparency to consumers who may want to know more about their foods [8]. Kumar [17] in 2017, mentions that one of the aspects which impacts the livelihoods and rural prosperity is agricultural management. The options of agricultural growth and connections with farm investment are the key element to agricultural development strategy.

In 2019, Mexico was in third place in the agricultural sector in America Latina and 11th place at the global level. Agriculture is the mainstay sector in the country's economy which wields a socio-economic impulse; with the culture of continuous improvement and the incorporation of technologies on the field, Mexico in the last year became within the 10 first agri-food products exporter [20]. The importance of habanero chili production has been used by different fields; in the pharmaceutic industry, according to Lopez-Puc [19] who studies the implementation of biotechnology in gathering varieties of habanero chili by its high capsaicinoides and capsaicina content, which are the raw material to the ointment production that relieves arthritis pain. In the agro-alimentary industry, for its proceeding in several foods that contain, habanero chili has elevated vitamin and minerals index [29].

It is used as an electrical system and irrigation coating to avoid rodents attacks. Lopez [19] also indicates in a significate way that, habanero chili is one of the least harmful chilies and is considered powerful healing, besides, it helps with gastritis and hemorrhoid problems. The 80% from chili habanero production is marked as dried fruit and 20% as sauce production like pasta and dehydrated [7]. Habanero chili it has been cultivated in the Yucatan's peninsular and is the leading production all over Latinoamerica in 2018 [30, 35]. Like farming, it has strong economic importance to vegetable producers in Yucatan state: remains in second place after the tomato farming concerning ground farming and, due to their demands to weather, the use of the controlled environment is higher [18].

There have been already published research documents terms as agronomics, climatic, genetic, chemical, among others as Santana et al. [31]. who investigated the formation of shoots of habanero chili plants with supplements in the shoots applying variables concentration of kinetina, benciladenina, and tidiazuron. On the other hand, as times go by and the social requirements for having data control, its management and generation of information in areas like medicine, astronomy, chemistry, biology, among others, has been supporting technological and specialization challenges cause of the data volume demands, the idea of gathering not only useful and in time information, but the knowledge acquired [24, 32].

However, with technological advance and social requirements, the statistic has been involved by its essence which is identified as science and classified depending on the purpose of data in statistic descriptive, demographic, probabilistic, or administrative in order to identify a phenomenon, concept, or incident [13, 3, 34]. In 1999, Witten et al. has taken the first steps to design learning methods through data exploration techniques, applying systems through data analysis with the WEKA tool, using the RJ48 algorithm, they isolate qualified attributes for a market and price of mushrooms labeling. Witten et al. claim to support a minimum work of programming to achieve the learning.

Majumdar [20] in 2017 analyzed the data sets from different public databases through data mining techniques applied in the agronomic area, particularly, clustering techniques. The work was done with wheat farming. The groups were divided by districts assessing different variables like atmospheric humidity, pH, temperature, among others. The efficiency of the annual crop was obtained by linear regression. As well, Amato et al. [2] in 2013 make sure about the nonparametric techniques usually exceed the parametric. Amarato realized research with discriminating data analysis tools that adapt to images hiperespectrality to identify the use of farmed agriculture soil. The good results he acquired for the classification using the previous transformation of data and he remarked that the false positives could be prevented, also using a group of data little training and reaching robustness and capacity to identify the categories in its study. Conversely, Kanahal [16] in 2019 studied the needs of farmers basing on a questionnaire applied

to experts in consideration of different variables as revenue, farm size, and an agricultural occupation for modeling, and the adoption of a predictive direction system. In his work, he mentions that the farm size, revenue level, and agriculture occupation are important facts in the adoption modeling and the applying of the GPS system.

Issad [1] in 2019 submits a revision of the implementation and study of Data Mining Techniques in agriculture. In this work, he mentions the Padalulu's et al. study, his proposal is the estimate of fertilizer and irrigation, as well as Perea et al. [12], in 2019 they had worked on decision trees and genetic algorithms to predict irrigation events. The wheat production was reinforced by satellite climatic studies in Australia, creating empiric models to predict the efficiency, however, Cai et al. [5] in 2019 observed the benefit of not only gathering the climatic data via satellite but they compared climatic data already acquired. They had used the regression method known as LASSO and three learning methods to build prediction empiric models. They claim that the method based on automatic learning overcome the regression method. And due to the successful work achieved, their suggestion is to apply the same techniques to other crops.

Rajeswari [27] in 2017, create a model for data analysis bigdata through the cloud, he analyzed variables as fertilization, growth, market, as well as requirements for growth. Initially, his proposal was the extraction of information through the digital interconnection of routine objects with the internet (IoT Devices), storing it in one of the databases in the cloud, subsequently, pre-processing and labeling were made, getting a selection of attributes completing the application of a pattern algorithm of MapReduce prediction.

It is well known that problems in the real world are even complex, nonlinear, and a stage that could be charged with multivariable uncertainties, multimodal, discontinuous, or exponential. This is an immersion to get to the point; studying a data collection or getting a processing and analysis algorithm, does not ensure the information gathered [35, 6]. Although it is been a long time since the statistic was considered by the specialists as an individual science in the research study, like Stigler [32] in 1986 mentioned in his book the requirement of intervention in some disciplines in the employment of the data analysis management, consequently, this premise has been fundamental support to get to the source called the Data Science [4, 10]. Data Science is an interdisciplinary field for the research generation that is useful, relevant, and innovative, converted into knowledge, the multidisciplinary thought from fields as programming, communication, management, sociology, and mastery of topical, are intended to a reflection of Data Science [35, 6, 34, 33].

The researches already realized in the agronomic area apply theories after the data collection, either, using data in a computing system to produce information with a specific target, also there are works where technologies and smart algorithms have been applied for data analysis like [22, 36, 15]. However, studies are ensuring that the model system is important on data knowledge, this is, the recognition of involved disciplines in a solution to make a collaborative work and systematic that performs a succeed on the project [26].

The expansion of the system consists of several phases: Delay phase (phase "lag) is a short period of adaptation or an increasing of startup to the half; transition phase, accelerated growth, which leads to exponential growth (logarithm phase); negative acceleration phase, imbalance phase, which leads to a stationary phase: characterized by the net coefficient of increasing declares null. Based on studies realized by Hernandez et al. [14], in grafted cucumber plant had been observed that fruit weight by plant correlated positively and significantly with plant height r=0.63*, stem diameter r=0.59* and leaf number r=0.54*, revealing these characters are important for this production system.

While Estrada et al. [9], evaluating tomato genotypes, found that the leaf length correlated with the fruit diameter, the highest correlation match to the length and wide of leaf $r = 0.96$, also highlight, by the number of leaves by plant, the number of clusters by plant, number of fruits by cluster, the average weight of fruits, and the fruit diameter, which means that are correlated. Other studies, like Pinedo [25] realized, studying the

genetic improvement of Camu-camu, had found a correlation between the leaf length, petiole length, and the fruit weight, that is a specie with high vitamin C content; as equal as Nieto et al. [23] evaluating the chirimoya selections, they had found that the limbo area, the leaf perimeter, the petiole length, and the longitudinal axis, were variables that had higher correlations with a highly significative level $(P \leq 0.01)$, proving its dependence; in this studies that have been realized in different species, its been determined a high correlation besides submitting a logistic type of growing that match absolutely in the gathered results on the current investigation.

This work suggests a study of determinate variables of a data collection from habanero pepper plants settled at the beginning of Data Science. It is analyzed the data representation complex, the vision of behavior over and as time goes by, preserves the mastery of knowledge of the data topic and its generalities on the experimental study.

As a result of our work, we have found that is possible to identify the relation between determinant variables. Even better, the Neural Network algorithm proved to be a great support widget in the research of experimental variables, emphasizing the relevance of the variable which has the reply to particular incidents, in other words, a prescriptive analysis of data has been developed with great success. The relevance of this study lies that no work has studied the variable knowledge relation of habanero pepper through computational algorithm techniques.

2 Related Work

The applied methodology was through the Data Science philosophy, where its discipline simulates a coordinate and systematic mechanism. In this section, the first steps were described, as well as the experimental design of the plants and their treatment, which is critical for the experimental replication as well as the procurement process and the data management of the experiment at the 3 harvest events conclusion. In addition, the Artificial Neural Network used for knowledge analysis has been presented.

Table 1. Genotypes of C. chinense in greenhouse experiments

Gen	Site	Name
G2	PGHBN2-130217-C1	Gliese 204 Chato1
G3	PGHBN3-130217-C1	Gliese 204 Chato1
G4	PGHBN4-130217-C1	Betelgeuse 2
G5	PGHBN5-130217-C1	Betelgeuse 2
G6	PGHBN6-130217-C1	Rigel 1
G7	PGHBN7-130217-C1	Gliese 204 Chato1
G8	PGHBN8-130217-C1	Rigel 1
G9	PGHBN9-130217-C1	Gliese 204 Chato1

Fig. 1. Growing habanero pepper plants

2.1 Experimental Crop Design

The experimental design was completely random with 8 treatments and 23 replications, getting 134 experimental units. The assessed variables in the habanero pepper plants were: plant height, leaf width, leaf length, bloom time al first and second shoot, and the second shoot, bloom time, number of fruits, and the greenhouse weather.

The selection of the seed was the beginning, until obtaining data from three harvests. In Table 1 the genotypes under study for their adaptation are presented. The conditions used in the greenhouse and the treatment of the plants used for the study variables are shown.

The measurement of the variables was performed in centimeters below describe each of them:

— Plant height (API). Seized from the bottom of the stem to the highest apex of the plant in centimeters.

— Number of primary stems (NTP). The number of secondary limbs of the leaf from the core stem was quantified.

— Number of secondary stems (NTS). The number of tertiary stems was quantified based on their secondary stem.

— Leaf length (LHj). The leaf development was quantified in the core rib at different sampling dates, using as fundamental the start of leaf limb.

— Leaf width (Anh). The leaf plant evolution was evaluated from its middle segment in terms of different sampling dates meanwhile its growth.

— Fruit length (LF) The fruit length will be measured.

— Fruit width (AF). By Measuring the widest fruit section, the width will be determined.

— Environmental temperature, in the greenhouse.

— Closed bloom (FC). To evaluate this variable the flower bud has to emerge on the three emergency days, closed bloom (days).

— Semi-open bloom (FSA). It had been qualified when the bloom is 30% open, but without pollination, the views as semi-closed bloom were selected.

— Open bloom (FA). It will be observed if the bloom is completely open and the anthers are dehiscences (days).

— Pollination (Pol). The bloom condition will be monitored if it bears fruit, the result will be recognized as pollination.

— Fruit Color (CF). The color might be considered based on the pigmentation acquired over the maturation process.

— green-dark fruit (CVOF). The fruit will reach this color as of the 20 days of pollination.

Fig. 2. Identification of habanero pepper plants

— Light-green color (CVCIF). After 3 or 4 weeks later of pollination change this color (visual).

— Light-green with red or orange pigmentation according to its genotype (CPRN). In a period of 4.5 to 5.5 weeks of pollination, this is the pigmentation produced in the fruit (Visual).

2.2 Design and Analysis of Data

With globalization advances and global warming, the agronomic field is concerned by different purposes of the farming process such as: cost reduction, resource optimization, and/or the genetic improvement of the product. The habanero pepper experts who have been contributing to this project, not only are interested in weather adaptation in protected crop to improve efficiency and production, but also the phenomenological condition from variables of the habanero pepper plant.

In the first analysis phase, the registered data were achieved and reviewed in numerical value with a spreadsheet. The data were evaluated and discerned through a set of continuous and discrete data values for a first descriptive diagnostic that were determined by the information gathered.

For the data analysis, a statistical function was investigated to reach the descriptive analysis as Pearson's correlation test between the variable of height-leaf, height-stem, and finally, fruit and plant of the three crop periods.

2.2.1 Data Modeling

A set of data values was used carefully evaluated with the intention of noise-canceling which might harm the analysis process, irrelevant variables were isolated, in other words, those that don not have a coincidence relation between them. Once the database has been already reviewed, an entraining function was applied to a set of data preparing it and using it in a Neural Network Algorithm. The tests were made once the model has been trained over the set of data to reach the accurate model.

The tests that were made in this stage had the intention to create a machine learning model to explain if the plant height might have an impact on the production, or any determinant variable. Through the multiple linear regression function with function 1, a diagnostic and descriptive model has been presented describing the issue and what can be achieved through an set of exposed data:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon, \tag{1}$$

whereβ_0 is the independent term and Y is the desired value.β_1, β_2,...,β_p are the biased coefficients of regression. ϵ is the perceived error due to no controlled variables. Then, the model of multiple linear regression is interpreted with function 2:

$$\begin{aligned} R' = \beta_0 + \beta_1 Higth + \beta_2 Leaf1L + \\ \beta_3 Leaf1 + \beta_4 Leaf2L + \beta_5 Leaf2w \\ \beta_6 Temperature + \beta_7 Plants + \epsilon. \end{aligned} \tag{2}$$

Statistics models help us to qualified the validation and reliability of the gathered data. The study of correlation has been used to know the relation level during the variables growing in the study, another one, is the regression to identify the dependent and independent variable of the leaf and fruit growing, to evaluate and validate the relationship between both.

Among the logarithmic and exponential functional models, a variant of the regression of each genotype was analyzed by genotype and to know the data prove. Then, an analysis of lineal regression had been performed providing the dependency between the expressed data, in other words, the dependent and independent variables.

Once the behavior of the dependent variables has been recognized, machine learning through supervised learning has been suggested. The data already shown have been analyzed as the dependence of the variable to the response variable. In order to know the effect of temperature on plant growth or a variable that directly affects, or even, to identify the growth-production based on the genotype, the paradigm of the Artificial Neural Network was presented which supports the optimization and prediction of the reply variable.

2.2.2 Machine Learning: Artificial Neural Network

Since 1888, Ramón and Cajal [28] prove that the nervous system is formed by interconnected neurons that learn the influence of external information. There are different ways to learn as they are: through new connections, through connection breaks, links between neurons, or the neurons reproduction. Artificial paradigm of Neuronal Networks simulate the biological Neuronal Network, the standard model arises in 1986 by Rumelhart and McClelland as defined in function 3, we become aware of Artificial Neural Network in [21]:

$$f_i \left(\sum_{i=1}^{n} w_{ij} x_j - \theta_i \right), \tag{3}$$

where x_j represents an entry and w_{ij} the synaptic weights, the propagation rule used is to mix linearly the entries and the synaptic weights depending on the function 4:

$$h_i(x_1, \ldots, x_n, w_{i_1}, \ldots, w_{i_n}) = \sum_{i=1}^{n} w_{ij} x_j. \tag{4}$$

The Adaline Artificial Neural Network (AANN) was used in Adalina, introduced by Winrow and Hoff in 1960, and the name comes from ADAptative LInear Neuron, operational input of continuous values intended to classify data. Different from other neuronal networks, one additional parameter

Algorithm 1 ANN training

```
Train_ANN (A,D) [ANN: Network Neural, D:
Data set, IL: Input Layer,
OL:Output Layer, e:error]
w randomly chosen weights --> ANN  [-1,1]
Repeat
    For (x,c) de D
        Insert x en IL
        Propagate the values of
             neurons from the IL (forward)
        Read the output y(x) = OL
        Calculate the(x) = |c-y(x)|
        Use error e(x) to adjust w
             and minimize the x error
Return w
```

has been incorporated called bias giving a free degree to the model (see function 3):

$$E(W) = \frac{1}{2}\sum_{r=1}^{N}\sum_{i=1}^{m}(c_i^r - y_i^r)^2. \tag{5}$$

The notable thing about this model is the learning rule, identified as LMS (Least Mean Square), learning is regulated by the weights selected to the error made by the neuron. With the cost function, network optimization has been acquired (see function 5). And the Artificial Neural Network training steps are in the algorithm 1.

3 Results and Discussions

The study presents a close relationship between the fruit and leaf variables, as we can see in Figure 3, where the value of the determining variables has a logistic behavior. During the first data gathering, the fruit and leaves variables show fast germination and tend to converge towards slower germination.Coefficient of determination is $R^2 = 0.94$ of the LHj and AnH variables and the length growth of leaf which is already explained in the width leaf development.

Difference between leaf 1 and leaf 2 growing by its position in the plant could be considered, due to

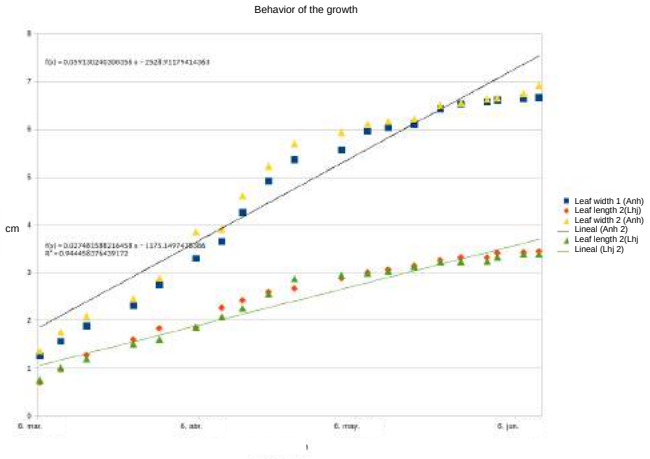


Fig. 3. Behavior of the growth rate of the variables leaf width and fruit width of the Rigel genotype and their relationship as a function of temperature

the apical dominance that is in the top section, its development is faster than the bottom of the plant.

The germination average of the habanero pepper leaf of genotype Rigel is 4.71 cm, considering that can reach the beginning of the growth of the leaf development, this variable has a variant value of 3.71, this reply is because the data of the variables was scattered and with more variability. The behavior of the width leaves in the same genotype was an average of 2.53 cm, which was found in the same days of the length germination, the variance is 0.74 lower due to the data is closer in relation to the average value of the width leaf (see Table 2).

A factor that influenced the growth rate and the production of the habanero pepper crop was the temperature. The range of temperature values in the greenhouse is between 48.75ºF and 91.85 ºF.

In Figure 4 the increase of the leaf growth and the fruit in temperature function have been observed. On the axis of the independent variables, the values of the average temperature have been considered and it is relational with the growing rate of the leaf length at that moment, and the fruit length is presented at the same time.

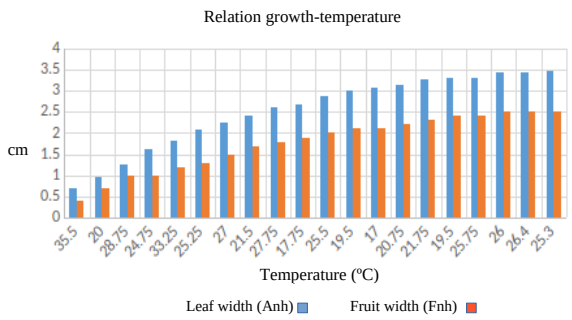


Fig. 4. Behavior of the growth rate for the variables width of the plant leaf and the width of the fruit of the Rigel genotype and their relationship as a function of temperature

Table 2. Determination coefficients

Groups	Rep	SC	$\overline{x}$	S^2
Large(LH)	20	94.24	4.71	3.71
Width(AnH)	20	50.67	2.53	0.74
Fruit	20	50.67	2.53	0.74

Table 3. Coefficients of determination and standard deviation to variables fruit width and plant leaf from chilli Gliese genotype and Riegel

Study factor	Betelgeuse	Riegel	Gliese
Coef. of det. R^2	0.91	0.95	0.95
Adjusted R-squared	0.91	0.95	0.95
Error	0.2 1	0.18	0.17

The increase in each variable is distinguished, even though the fruit length is slower.

In the linear regression analysis between the variables length leaf width (AnH) and fruit width (AF) a highly significant correlation was obtained, where the sum of squares was 9.88 and the critical value of F was $9.1898E-12$, regarding the length of the fruit and the leaf, the sum of squares is 12.18 and the critical value of F is $2.54E-14$. Factors study genotypes can be seen in Table 3.

The results of the multiple linear regression technique showed a significant association in those

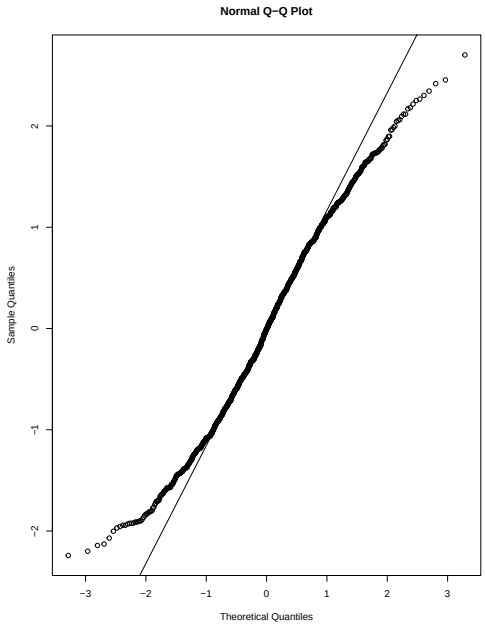


Fig. 5. Residual diagnostics in the data set

predictive variables such as height, temperature and type of pepper genotype, Figure 6.

On the other hand, the estimated line obtained from the regression diagnosis can be observed in the equation 6, the $Leaf2L$ variable was eliminated because it does not present significant values in the response variable.

$$R' = -1.04 - 0.048 * Higth - 0.18925 * Leaf1L + 0.493 * Leaf1w - 0.447 * Leaf2w - 0.03805 * Temperature + 0.212 * Plants + \epsilon. \quad (6)$$

Furthermore, these results showed a growth trend with respect to yield for each treated genotype (see Figure 7). The normality of the studied data set can be observed in Figure 5, this represents a long-tail behavior in the residuals, which means that the proposed model must be studied.

Although a statistical model was found and this represents a proposed solution, the problem is complex as can be seen in Figure 7, where it only shows the correlation of two variables.

It is necessary and advisable to continue with the study to find the information of answers to questions as; If there is a relationship between the determining variables type of plant and yield, or also pre-write which variable could identify the type of plant in question, before having the fruit, that is, without having to invest in time.

One of the first Artificial Neural Network experiments was to identify if the value of the length of the leaf can help predict the height of the plant. Among the parameters that he included in the algorithm were the variable $h1$ and the response variable $height$, the error function was also used the equation 7 and the activation function applied was 8. The results showed a correspondence between both variables:

$$err(x, y) = \frac{1}{2} * (y - x)^2, \tag{7}$$

$$f(x) = \frac{1}{1 + exp(-x)}. \tag{8}$$

Another interesting piece of data is that the Garson [11] method of the Artificial Neural Network paradigm was used in the search for response variables among all the variables that were included in the study. The method made a significant connection between all nodes, giving a weighting value to each connection and, later, the technique considered a disconnection between the evaluation of weights on variables that do not have a significant relationship that is evaluated by the paradigm, since the connections were scaled and disarticulated, those variables that were not relevant to find the response variable were eliminated.

Figure 8 shows the variables of the data set that obtained a significant level of importance (x's axis) in relation to the plant genotype (y's axis), this predictive analysis will allow the expert to decipher the type of habanero chili without waiting for the plant to expose the fruit. In the same way, the study was considered very significant because the power of a neural network was verified in a prescriptive-level data analysis. Another example that showed its importance was the identification of an explanatory variable for performance, the result of the Artificial Neural Network can be seen in Figure 9.

```
Residuals:
    Min      1Q  Median      3Q     Max
-5.1585 -1.7914 -0.0087  1.8129  6.2121

Coefficients:
             Estimate   Std. Error  t value   Pr(>|t|)
(Intercept) -1.03911    1.05112    -0.989    0.323115
No_data      1.34144    0.02048    65.497    < 2e-16 ***
lants        0.21267    0.03386    6.281     5.05e-10 ***
Leaf1L      -0.18925    0.08247    -2.295    0.021957 *
Leaf1w       0.49331    0.12257    4.025     6.14e-05 ***
Leaf2L      -0.05921    0.07744    -0.765    0.444726
Leaf2w      -0.44678    0.12866    -3.473    0.000538 ***
Temp        -0.03805    0.01313    -2.897    0.003854 **
Height      -0.04789    0.01519    -3.153    0.001667 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.307 on 979 degrees of freedom
Multiple R-squared:  0.9169,  Adjusted R-squared:  0.9162
F-statistic:  1350 on 8 and 979 DF,  p-value: < 2.2e-16
```

Fig. 6. Results of data analysis

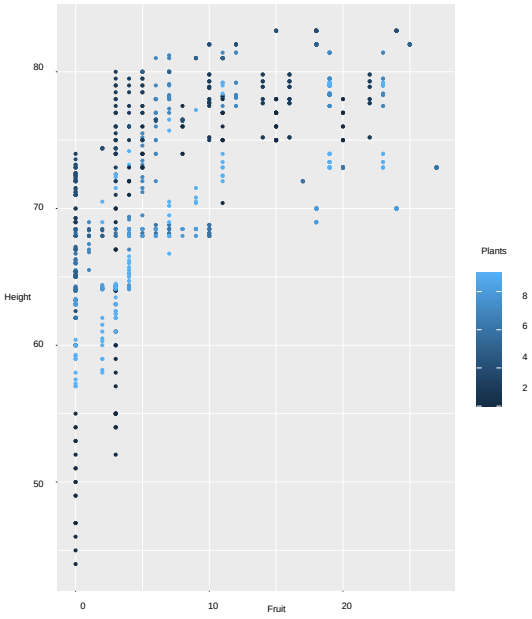


Fig. 7. Graphical representation of the dispersion between variables

ISSN 2007-9737

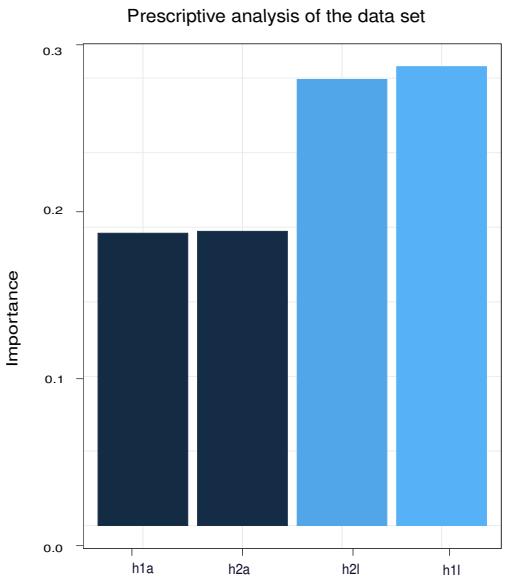


Fig. 8. ANN paradigm in the data set

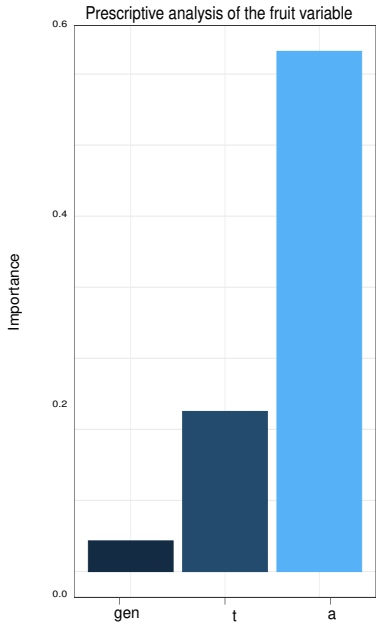


Fig. 9. Importance of the explanatory variable for the response variable Fruits

4 Conclusion and Future Work

The study of experimental data in the habanero pepper crop was performed by a multi-disciplinary team, where team members whose area of expertise is computer science, interacted with experts in agricultural science. In this research, the experimental data obtained was analyzed by statistical and data mining techniques with the main purpose of finding interesting information beyond the prediction that a multiple regression analysis could produce.

In fact, the linear regression model appears to be insufficient for finding an explanatory variable that fully determines the response variable. We found that the Artificial Neural Network approach was quite valuable for carrying out our analysis. This work is a process that we find attractive for the analysis of multivariable linear and not linear data, which are required for obtaining a diagnostic that is both, predictive and prescriptive.

The studies carried out show that determining variables such as plant height, leaf length, leaf width and fruit have a positive and significant correlation, as did Hernandez et al. [14] with they study of the cucumber plant.

On the other hand, Estrada [9] in his study of tomato genotype found a correspondence relationship between the values of the fruit and the plant with respect to the data of the clusters, we found a strong relationship between the fruit and the values of the plant's leaf $r^2 = 0.94$.

The study of the data of a habanero pepper crop was carried out in a multidisciplinary study, the data obtained was analyzed by statistical and data mining techniques in order to find interesting information beyond a multiple regression study.

The regression model does not seem sufficient to find an explanatory variable that determines the response variable, being the ANN technique very helpful for this objective.

Acknowledgments

We would like to thank to TecNM/IT Roque for the support to carry out the project in the greenhouse and the facilities. Almost, thanks to the MC.

Chablé for his advice and support in conducting the machine learning study. On the other hand, to Román García for the support in the translation. This work is funded by agronomy department of TecNM/IT Roque.

References

1. **Ait, H., Aoudjit, R., Rodrigues, J. (2019).** A comprehensive review of data mining techniques in smart agriculture. Engineering in Agriculture, Environment and Food, Vol. 12.
2. **Antoniadis, A., Carfora, M. F., Colandrea, P., Cuomo, V., Franzese, M., Pignatti, S., Serio, C., Amato, U. (2013).** Statistical classification for assessing prisma hyperspectral potential for agricultural land use. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 6.
3. **Badii, M., Araiza, L. A., Guillén, A. (2010).** Esenciales de la estadística: Un acercamiento descriptivo (essentials of statistics: A descriptive approach). df, pp. 4.
4. **Ben, D. (2017).** Big data and data science: A critical review of issues for educational research. British Journal of Educational Technology.
5. **Cai, Y., Guan, K., Lobell, D., Potgieter, A., Wang, S., Peng, J., Xu, T., Asseng, S., Zhang, Y., You, L., Peng, B. (2019).** Integrating satellite and climate data to predict wheat yield in Australia using machine learning approaches. Agricultural and Forest Meteorology, Vol. 274, pp. 144–159.
6. **Cao, L. (2017).** Data science: A comprehensive overview. ACM Comput. Surv., Vol. 50, No. 3.
7. **CastellónMartínez, E., ChávezServia, J., Carrillo-Rodríguez, J., Vera-Guzmán, A. (2017).** Preferencias de consumo de chiles (capsicum annuum l.) nativos en los valles centrales de Oaxaca. Revista fitotecnia, mexicana, Vol. 35, No. 5, pp. 27–35.
8. **Creutzberg, G. (2015).** Agriculture 3.0: A New Paradigm for Agriculture. Nufflier Canada.
9. **Estrada, Y., Lescay, Y., Vázquez, F., Celeiro, F. (2012).** Variabilidad genética y correlaciones fenotípicas en germoplasma de tomate (solanum lycopersicum l). Granma Ciencia, Vol. 16.
10. **Gibson, D., Ifenthaler, D. (2017).** Preparing the Next Generation of Education Researchers for Big Data in Higher Education, chapter 3. Springer, pp. 29–42.
11. **Goh, A. (1995).** Back-propagation neural networks for modeling complex systems. Artificial Intelligence in Engineering, Vol. 9, No. 3, pp. 43–151.
12. **González, R., Camacho, E., Montesinos, P., Rodríguez, J. (2019).** Prediction of irrigation event occurrence at farm level using optimal decision trees. Computers and Electronics in Agriculture, Vol. 157, pp. 180.
13. **Gutierrez, C. (1994).** Filosofía de la estadística. Universitat de Valencia Servei de Publicacions, EU, 1 edition.
14. **Hernandez, Z., Sahagun-Castellanos, J., Espinosa-Robles, P., Colinas-León, M. T., Rodriguez Perez, J. E. (2014).** Efecto del patrón en el rendimiento y tamaño de fruto en pepino injertado. Revista fitotecnia mexicana, Vol. 37, pp. 41–47.
15. **Jones, J. W., Antle, J. M., Basso, B., Boote, K., Conant, R. T., Foster, I., Charles, G., Herrero, M., Howitt, R., Janssen, S., Keating, B., Munoz-Carpena, R., Porter, C., Rosenzweig, C., Wheeler, T. (2017).** Toward a new generation of agricultural system data, models, and knowledge products: State of agricultural systems science. Agricultural Systems, Vol. 155, pp. 269–288.
16. **Khanal, R., M, A., Lambert, D., Paudel, K. (2019).** Modeling post adoption decision in precision agriculture: A Bayesian approach. Computers and Electronics in Agriculture, Vol. 162, pp. 466–474.
17. **Kumar, M., Nagar, M. (2017).** Big data analytics in agriculture and distribution channel. pp. 384–387.
18. **Lecona-Guzmán, C. (2017).** Mejoramiento genético de chile habanero: selección y registro de variedades mejoradas. Revista fitotecnia, mexicana, Vol. 35, No. 5, pp. 27–35.
19. **López-Puc, G., Canto-Flick, A. (2009).** El reto biotecnológico del chile habanero. Ciencia, Vol. 60, No. 3, pp. 30–35.
20. **Majumdar, J., Naraseeyappa, S., Ankalaki, S. (2017).** Analysis of agriculture data using data mining techniques: application of big data. Journal of Big Data, Vol. 4, No. 20.

21. **Minsky, M., Papert, S. (1969).** Perceptrons: An Introduction to Computational Geometry. MIT Press, Cambridge, MA, USA.

22. **Muthurasu, N., Sahithyan, S., Aravind, M. T., RamanagiriBharathan, A. (2018).** A prediction system for farmers to enhance the agriculture yield using cognitive data science. Journal of Advanced Research in Computer Science.

23. **Nieto, R., Andre, J., Barrientos-Priego, A., Martínez-Damián, M., González-Andrés, F., Segura, S., Gallegos-Vázquez, C. (2003).** Variación morfológico de la hoja del chirimoyo. Revista Chapingo, Serie Ciencias Forestales y del Ambiente, Vol. 10, pp. 103–110.

24. **Perero, M. (1995).** Historia e historias de matemáticas. Grupo Editorial Iberoamericano, México, 1 edition.

25. **Pinedo, M. (2012).** Correlation and heritability analysis in the genetic improvement of camu-camu. Scientia agropecuaria, Vol. 3, pp. 23–28.

26. **Pivoto, D., Waquil, P., Talamini, E., Finocchio, C. P. S., Dalla Corte, V. F., de Vargas Mores, G. (2018).** Scientific development of smart farming technologies and their application in brazil. Information Processing in Agriculture, Vol. 5, No. 1, pp. 21–32.

27. **Rajeswari, S., Suthendran, K., Rajakumar, k. (2018).** A smart agricultural model by integrating iot, mobile and cloud-based big data analytics. International Journal of Pure and Applied Mathematics, Vol. 118, pp. 365–369.

28. **Ramon, C. (1899).** Textura del sistema nervioso del hombre y de los vertebrados. Moya, Madrid, 1 edition.

29. **Rozete, M. (2019).** Caracterización fitoquímica y sensorial de variedades de chile habanero (Capsicum chinense Jacq.) Yucatán. Ph.D. thesis, Centro de Investigación Científica de Yucatán, A.C. Posgrado en Ciencias Biológica, Yucatán.

30. **Sagarpa (2020).** Planeación agrícola nacional 20172030 chiles y pigmentos mexicanos.

31. **Santana-Buzzy, N., Canto-Flick, A., Barahona-Pérez, F., Montalvo-Peniche, M., Zapata-Casillo, P., Gutierrez-Alonso, O. (2005).** Regeneration of habanero pepper (capsicum chinense jacq.) via organogenesis. HortScience: a publication of the American Society for Horticultural Science, Vol. 40, pp. 1829–1831.

32. **Stigler, S. (1986).** The History of Statistics: The Measurement of Uncertainty before 1900. The Belknap Press of Harvard University Press, EU, 1 edition.

33. **Tukey, J. W. (1962).** The future of data analysis. Ann. Math. Statist., Vol. 33, No. 1, pp. 1–67.

34. **Tukey, J. W. (1977).** Exploratory Data Analysis. Behavioral Science: Quantitative Methods. Addison-Wesley, Reading, Mass.

35. **Vasquez, B. A. (1999).** Ciencia de datos para gente sociable.

36. **Wolfert, S., Ge, L., Verdouw, C., J, B. M. (2017).** Big data in smart farming – a review. Agricultural Systems, Vol. 153, pp. 69–80.

Article received on 11/12/2021; accepted on 12/04/2022. Corresponding author is Blanca C. López-Ramírez.

ISSN 2007-9737

Synthesis and Characterization of ZnS Nanoparticles and Effects of Nanoparticle Size on Optical Properties

Maria Elena Aguilar Jauregui[1,3], José Abraham Balderas López[1], Eduardo San Martín-Martinez[2]

[1] Instituto Politécnico Nacional,
Unidad Profesional Interdisciplinaria de Biotecnología,
México

[2] Instituto Politécnico Nacional
Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada-Legaria
México

[3] Instituto Politécnico Nacional
Centro de Investigación en Computación
México

maguilar@cic.ipn.mx,
{jbalderasl, esanmartin, esanmartin}@ipn.mx

Abstract. This research presents a methodology for obtaining ZnS nanoparticles (NZnS). Doped with transition metal ions ($MnCl_2$) and use of rare earth $EuCl_3$ to improve their photoluminescence properties. Semiconductor nanoparticles were synthesized using the nanoprecipitation technique. The characterization of the ZnS nanoparticles was carried out using Dynamic Light Scattering (DLS), Powder X-Ray Diffraction (XRD), Scanning Electron Microscopy (SEM), High-Resolution Transmission Electron Microscopy (HRTEM), and Luminescence Spectroscopy (PL) technique. A cubic zinc crystalline structure could be observed considering the position and intensity of the characteristic peaks in the XRD pattern. Also, taking the width at half height of the main peak and using the Debye-Scherrer equation, it is possible to obtain the average size of the nanoparticles, which for the samples analyzed was between 2-4 nm. SEM micrograph revealed a morphology with aggregated nanoparticles. High-Resolution Transmission Electron Microscopy showed the crystalline structure of the doped nanomaterial. PL characterization showed an increase in the luminescent intensity of doped NZnS compared to NZnS doped with the lowest concentration of the coating agent. These results confirm the effectiveness of the methodology for possible application in bioimaging diagnosis of PL in nanomedicine.

Keywords. Nanomaterials, nanoparticles, quantum dots, transition metal, lanthanide ions, passivation agent, luminescence.

1 Introduction

The study of nanoparticles is a relevant area of research for the scientific community due to their magnetic, catalytic, electronic, and optical properties, which are very different from those of bulk materials. Nanomaterials composed of semiconducting nanoparticles, or quantum dots (QDs) are very promising in various applications, particularly ZnS QDs.

ZnS is an inorganic compound that is used as a pigment or as a semiconductor material [1] and is considered a suitable host matrix to form doped phosphors [2]. Great advances have been made in research and technological development to obtain luminescent materials in the different regions of the visible spectrum. ZnS nanoparticles doped with transition metals such as Manganese (Mn) [2] have been used as optical readers, diodes, handheld computers, personal assistants and optical storage

[3].They are also suitable for applications as biological probes [4] and biosensors [1, 5] due to their luminescent characteristics in the visible spectrum, as well as their high biocompatibility [6].

Luminescence intensity is an important factor which have been improved modifying experimental conditions of synthesis, such as concentrations of reactants, pH, and temperature, among others [7- 9].

In the quest to obtain high luminescence intensity and near-infrared shift for possible medical and photonic applications, studies of semiconductors co-doped with transition metals and rare earth have also been carried out.

Lanthanide elements, also called rare earths, have relevant properties, which allows that semiconductor materials doped with these elements present convenient properties for their use in assay tests in biotechnology, bio imaging, bio-detection, diagnosis and treatment of diseases [10].

These properties include chemical stability [11], adjustable color emission [12], low toxicity, penetration into tissues with less damage, and allow selectivity [13]. From the reported research on rare earth, Europium (Eu) is a lanthanide material recognized as an efficient luminophore [14]; it presents a narrow and sharp bandwidth due to the transitions of the electrons in $4f^6$ orbitals, which are “protected” by the electrons in $5S^2$ and $5P^6$ orbitals, which means that the host keeps Eu without perturbing the emission states [15].

This exhibits an emission spectrum ranging from the early visible to the near-infrared. For these reasons, Eu-doped nanoparticles are an excellent alternative for many applications where long wavelength emission is required [16].

In this work, we present a method of synthesis and characterization of zinc sulfide nanoparticles doped with Mn and Eu, to obtain nanoparticles with sizes smaller than 100 nm and that produce high luminescence.

In the experimental process, two synthesis processes are carried out, increasing the concentration of the coating agent to evaluate the effect of the variation of the concentration on the size of the nanoparticles and the photo luminescent emission.

2 Experimental Procedure

2.1 Synthesis

The synthesis method used to obtain luminescent semiconductor nanoparticles is based on precipitation reactions by controlled-release processes of the precipitating cations or anions at room temperature. This is a simple and low-cost methodology [17]. In the present study, the precursors used to prepare the samples were: Zinc acetate (Zn $(CH_3COO)_2$ $2H_2O$) (99.99%, Sigma Aldrich), Na_2S 9H2O (99.98%, Sigma Aldrich), $MnCl_2$ (99.99%, Sigma Aldrich), $EuCl_3$ (99.99%, Sigma Aldrich) and Polysorbate 80 (Tween 80), Ethanol and Deionized Water. All materials were used without further purification.

2.1.1 ZnS Nanoparticles

For the synthesis of zinc sulfide nanoparticles, two different solutions were prepared: a solution of 2.195 g of Zinc Acetate ($(Zn(CH_3COO)_2$ $2H_2O)$), Polysorbate 80 as a capping agent or passivating agent with a concentration of 0.6% w/v, in 50 ml of ethanol. Another solution of 2.451 g of Sodium Sulfide (Na_2S) in 50 ml of deionized water. The solutions were shaken vigorously for 30 min and the Na_2S solution was incorporated dropwise into the zinc acetate solution with Polysorbate 80 and kept in agitation for 1 hr to obtain a white precipitate of ZnS nanoparticles (Fig 1).

2.1.2 Doped ZnS Nanoparticles

Two syntheses were carried out by the precipitation method. For each synthesis two solutions were prepared. A solution of 2.195 g ($Zn(CH_3COO)_2$ $2H_2O$), Polysorbate 80 as a capping agent or passivating agent with a concentration of 0.6% w/v, 0.05 g $MnCl_2$ and 0.292 g $EuCl_3$ as impurities, mixed in 50 ml ethanol. Another solution of 2.451 g Na_2S in 50 ml of deionized water.

Both solutions were stirred vigorously for 30 min. Subsequently, the Na_2S solution was added dropwise to the first mixture and stirred for 1 hr., obtaining a white precipitate of ZnS:Mn,Eu nanoparticles (Figure 2).

In the second synthesis, the same procedure and quantities of the precursors were carried out varying the concentration of the capping agent to 1% w/v.

After obtaining the precipitate from each synthesis, this was subjected to centrifugation and washed several times, then left to dry at room temperature and the resulting powder was milled to obtain the impurified zinc sulfide nanoparticles. The overall synthesis process is schematized in Fig. 2.

2.2 Characterization of Nanoparticles

The obtained nanoparticles were characterized by means of Dynamic Light Scattering (DLS), X-Ray Dispersion (XRD), Scanning Electron Microscopy (SEM), Photoluminescence (PL), and High-Resolution Transmission Electron Microscopy (HRTEM) techniques.

DLS was used to obtain the hydrodynamic diameter of the nanoparticles in dispersion and their Z-potential, thus obtaining their average size in suspension and the degree of repulsion between adjacent nanoparticles in the sample allowing at this way to determine the stability of the colloidal solution [17].

Peak broadening analysis by XRD is a basic tool for characterization to obtain information regarding the crystalline structure, its size, and lattice deformation.

The calculation of nanoparticle size was obtained from the main X-ray diffraction peak by measuring the non-defective region or coherent scattering zone of the peak.

In the SEM micrograph, it can be observed how the morphology of the crystalline structure is and if agglomeration and polydispersity are present. In this technique, the nanoparticles can be observed individually in some way and measured directly.

Whereas in HRTEM microscopy the crystal structure of the sample can be observed at the atomic scale. By means of photoluminescence spectroscopy we measured the electromagnetic radiation emission of doped and undoped ZnS nanoparticles.

Fig. 1. Nanoparticles coated with Polysorbate 80 as passivating agent

Fig. 2. Synthesis of ZnS Nanoparticles doped with Mn and Eu

3 Results and Discussion

3.1 DLS Analysis

Table 1 shows three measurements for the same sample with 0.6% w/v of the coating concentration where the average particle size is 6 nm hydrodynamic particle diameter. As the sample presents a single peak (monomodal) of narrow distribution amplitude (monodisperse) the result can be compared with the size measured by other techniques.

Table 1 also shows that the nanoparticles have a zeta potential of 10.9 mV, indicating that the surface charges are positive and the magnitude indicates that the nanoparticles will prevent their aggregation among themselves, making them somewhat stable in storage time.

Being able to crowd. However, it is known that if nanoparticles are sterically protected with a surfactant or capping agent (passivator), they may experience less or no agglomeration and achieve stability at zeta potential values between ±30 mV. The positive or negative zeta potential and its

magnitude allow us to determine the chemical changes on the surface of the nanoparticles.

The pH is an important factor on which the zeta potential depends. In this experiment, a pH of 4 was measured and the zeta potential presented positive values [18], thus having a homogeneous and stable solution. Nanoparticle sizes were smaller than at pH 6 and pH 8.

3.2 X-ray Diffraction Analysis

XRD spectra indicating the presence of NZnS impurified with transition metal ions and rare earth are shown. The identified planes correspond to cubic-phase ZnS (Fig. 3).

The black spectrum belongs to the undoped NZnS.The blue spectrum corresponds to the ZnS sample doped with Mn and Eu (0.6 % of the capping) and presents diffraction peaks at 28.5°, 48 °, and 57.5° (of 2θ), corresponding to the (111), (220) and (311) crystallization planes.

The spectrum in green color corresponds to the sample of ZnS doped with Mn and Eu (1 % of the capping), and the diffraction peaks are located at 29°, 48 °, and 56.5° (of 2θ), corresponding to the planes (111), (220) and (311), respectively [19].

The estimated size of the synthesized nanoparticles was obtained using the Debye-Scherrer equation (1):

$$D=\frac{k\lambda}{B\,Cos\theta}. \quad (1)$$

D is the average diameter of the nanocrystals in the direction perpendicular to the related planes. The values of the full width at half peak maximum of pure and doped nanoparticles are presented in Table 2.

3.3 SEM Analysis

A SEM micrograph of the NZnS is present, showing their morphology. Nanometer-sized nanoparticles with agglomeration possibly due to the nature of the ZnS.

It is observed that the nanoparticles are almost spherical in shape with average diameters of 10 nm and smooth and uniform surface.

The nanoparticle sizes are consistent with those estimated from the XDR patterns.

Table 1. Size and Zeta Potential

Nanoparticles	Size (nm)	Zeta Potential pH4	Zeta Potential pH6	Zeta Potential pH8
ZnS:Mn,Eu (0.6% w/v)	6	10.9	13.8	12.4
ZnS:Mn,Eu (1% w/v)	11	23	23.2	15.6

Table 2. Estimated size of pure and doped ZnS nanoparticles using the Debye-Scherrer equation

Nanoparticles	FWHM B	K	λ Å(Cu-kα) (nm)	θ	Size (nm)
ZnS	0.07859	0.9	0.1542	0.5061	2.02
ZnS:Mn,Eu (0.6% w/v)	0.06109	0.9	0.1542	0.497	2.57
ZnS:Mn,Eu (1% w/v)	0.04363	0.9	0.1542	0.506	3.63

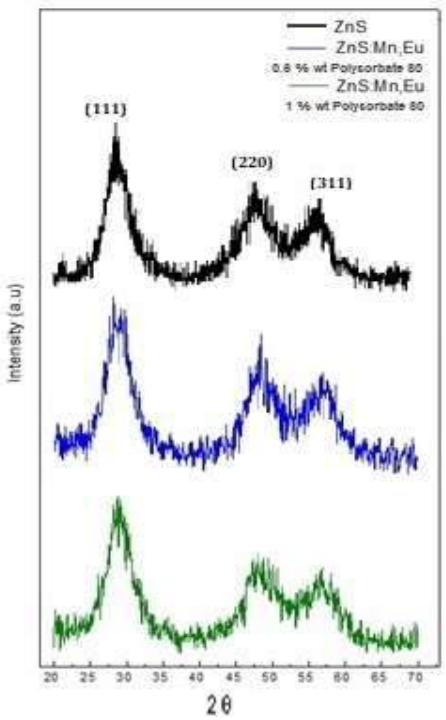


Fig. 3. XRD patterns of ZnS and Mn,Eu doped ZnS nanoparticles (0.6 % w/v, 1 % w/v Polysorbate 80)

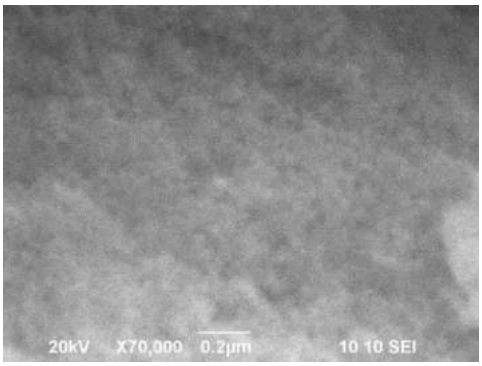


Fig. 4. SEM micrograph of doped nanoparticles

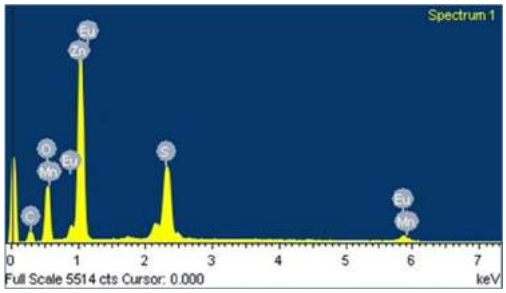


Fig. 5. Elemental chemical composition of the surface of the sample

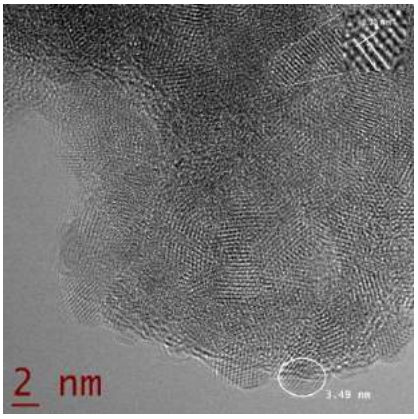


Fig. 6. HRTEM image of doped zinc sulfide nanoparticles

3.4 EDS Analysis

The energy dispersive spectroscopy analysis in Figure 5 shows the elemental chemical composition of the sample surface.

The presence of the precursor elements Zn, S, Mn, and Eu is observed.

A significant amount of oxygen and little carbon was found, possibly corresponding to the coating of the samples.

3.5 HRTEM Analysis

The micrograph obtained by means of HRTEM (Fig. 6) shows the formation of nanoparticles with sizes between 2 and 4 nm. The nanoparticle sizes observed in the image are in agreement with the sizes estimated from XRD spectrometry.

In the nanoparticles, lattice fringes are observed in the nanoparticles with the interplanar space assigned to the (111) planes of Zinc in the cubic phase.

3.3 Photoluminescence Studies

The obtained samples were irradiated with an ultraviolet light lamp, observing an emission in the red.

This is due to the recombination of Mn ions and the concentration of Mn used in the synthesis [20], in addition to the possible recombination of Eu ions with ZnS. (Fig. 7a).

It is observed from the PL spectra (Fig. 7c-d) that the visible light emitted by the doped nanoparticles was obtained with a broad emission peak with wavelengths of 612 nm at room temperature, with an excitation spectrum of 364 nm (Fig 7b).

The amplitude of the peak is characteristic of the Mn^{2+} ion. However, due to the concentration of Mn^{2+} used in the synthesis, an emission in the orange-red was achieved (Fig. 9a) [21]. A significant increase in luminescent intensity was observed in the ZnS:Mn,Eu nanoparticles with lower concentration of the passivating agent of about twofold.

This significant increase in intensity may be due to the passivation of the surface defects of the nanoparticles because this is where non-radiative recombination by the Mn 2+ ion centers [22] and Eu III occurs.

So also, the increase in surface area/volume influences, since the nanoparticle size decreases as the concentration of the coating agent decreases.

4 Conclusions

The nanoparticles of ZnS were obtained by precipitation of a homogeneous suspension of zinc sulfide, doped with transition metals and rare earths, and Polysorbate 80 as a passivation agent.

The synthesized ZnS nanoparticles have an average size of about 6 nm. (hydrodynamic diameter). The measured zeta potential indicates that the sample is homogeneous and stable.

From the technique to obtain the size of the nanoparticles by XRD spectra, it was possible to estimate a size between 2 to 3.63 nm with a cubic structure according to the planes that define the crystal lattice and this is corroborated with what was observed by SEM and the interplanar space of the HRTEM micrograph.

Also, the PL experiments show a strong luminescent intensity, which is approximately doubled due to the lower concentration of the coating agent. As the concentration of the coating agent decreases, the size of the nanoparticle is smaller and the luminescent intensity increases.

In our experimentation, due to the concentration of Mn added in the synthesis, and the incorporation of Eu III ions we obtained an increase in the emission wavelength, towards red, which is very important and favorable for biomedical applications.

ZnS nanoparticles doped with transition metals and rare earths can be considered to form a new unique luminescent material with strong and stable visible emission. With the doping of Mn and Eu in ZnS nanoparticles, an enhancement of luminescence efficiency is observed.

The micrograph results suggest that metal and rare earth ions were incorporated into the network of the new nanomaterial.

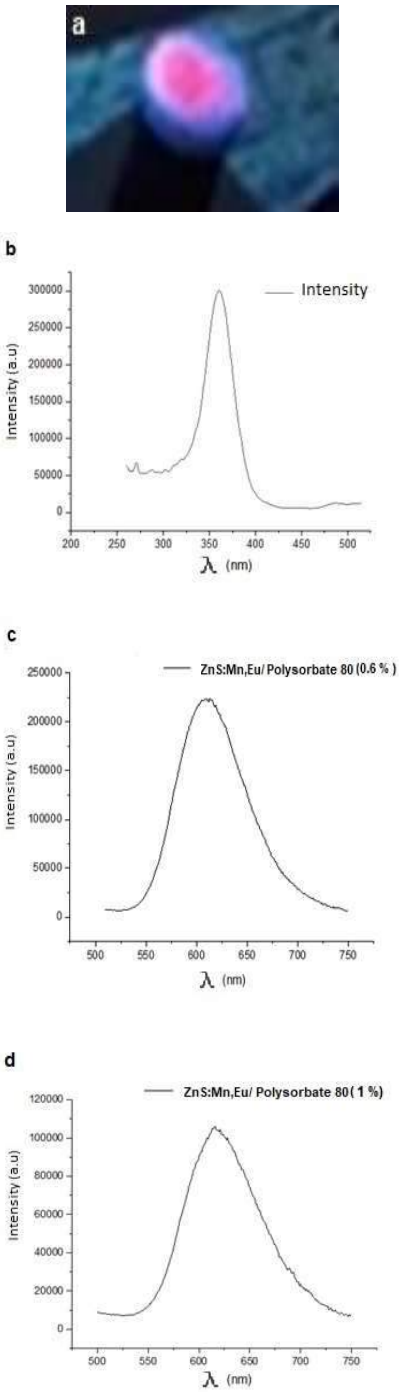


Fig. 7. a) Doped nanoparticle powder under UV light lamp, b) 364 nm excitation spectrum, c) Emission spectrum of ZnS:Mn, Eu nanoparticles with a lower concentration on the shell, d) Emission spectrum of ZnS:Mn, Eu nanoparticles with a higher concentration on the shell

Acknowledgments

The authors are grateful for the support provided by the Instituto Politécnico Nacional.

References

1 Allehyani, S.H.A., Seoudi, R., Said, D.A., Lashin, A.R., Abouelsayed, A. (2015).

Synthesis, characterization, and size control of zinc sulfide nanoparticles capped by poly (ethylene glycol). Journal of Electronic Materials, Vol. 44, No. 11, pp. 4227–4235. DOI: 10.1007/s11664-015-3974-3.

2 **Hossu, M., Schaeffer, R.O., Wei-Chen, L.M., Zhu, Y., Sammynaiken, R., Joly, A.G. (2013).** On the luminescence enhancement of Mn2+ by co-doping of Eu2+ in ZnS:Mn. Optical Materials, Vol. 35, No. 8, pp. 1513–1519. DOI: 10.1016/j.optmat.2013.03.014.

3 **Satya-Kamal, Ch., Mishra, R.K., Patel, D. K., Ramachandra-Rao, K., Sudarsan, V., Vatsa, R.K. (2016).** Effect of structure, size and copper doping on the luminescence properties of ZnS. Materials Research Bulletin, Vol. 81, pp. 127–133. DOI:10.1016/j.materresbull.201 16.05.010.

4 **Wolfbeis, O.S. (2015).** An overview of nanoparticles commonly used in fluorescent bioimaging. Royal Society of Chemistry, Vol. 44, pp. 4743–4768. DOI:10.1039/C4CS00 392F.

5 **Kaur, N., Kaur, S., Singh, J., Rawat, M. (2016).** A review on zinc sulphide nanoparticles: from synthesis, properties to applications. J. Bioelectronics and Nanotechnology, Vol. 1, No. 1. pp. 1–5.

6 **Kim, J., Park, K., Vazquez-Zuniga, L.A., Kim, H., Han, M., Jeong, Y. (2015).** Optical characteristics of Mn2+ doped ZnS nanoparticles for laser-based bio-sensing. Advanced Photonics, DOI: 10.1364/IPRSN. 2015.JM3A.43.

7 **Bhargava, R.N., Gallaghera, D., Welker, T. (1994).** Doped nanocrystals of semiconductors - a new class of luminescent materials. Journal of Luminescence, Vol. 60-61, pp. 275–280. DOI: 10.1016/0022-2313(94)90146-5.

8 **Bhargava, R.N., Gallagher, D. (1994).** Optical properties of manganese-doped nanocrystals of ZnS. Physical Review Letters, Vol. 72, No. 3, pp. 416–419.

9 **Sotelo, E.G., Roces, L., García-Granda, S., Fernández-Arguelles, M.T., Costa-Fernández, J.M., Sanz-Medel, A. (2013).** Influence of the Mn2+ concentration on Mn2+-doped ZnS quantum dots synthesis: evaluation of the structural and photoluminescent properties. Journal Nanoscale, Vol. 5, No. 13, pp. 9156–9161. DOI: 10.1039/C3NR02422A.

10 DaCosta, M.V., Doughan, S., Han, Y., Krull, U.J. (2014). Lanthanide upconversion nanoparticles and applications in bioassays and bioimaging: A review. Analytica Chimica Acta, Vol. 832, pp. 1–33. DOI: 10.1016/j.aca. 2014.04.030.

11 **Sousa, D.M., Alves, L.C., Marques, A., Gaspar. G., Lima, J.C., Ferreir, I. (2018).** Facile microwave-assisted synthesis manganese doped zinc sulfide nanoparticles. Scientific Report, Vol. 8, No. 15992, DOI: 10.1038/s41598-018-34268-z.

12 **Zuo, M., Qian, W., Li, T., Hu, X.H., Jiang, J., Wang, L. (2018).** Full-Color tunable fluorescent and chemiluminescent supramolecular nanoparticles for anti-counterfeiting links. ACS Appl. Mater. Interfaces, Vol. 10, No. 45, pp. 39214–39221. DOI: 10.1021/acsami.8b14110.

13 **Jain, A., Fournier, P.G.J., Mendoza-Lavaniegos, V., Sengar, P., Guerra-Olvera, F.M., Iñiguez, E., Kretzschmar, T.G., Hirata, G.A., Juárez, P. (2018).** Functionalized rare-earth-doped nanoparticles for breast cancer nanodiagnostic using fluorescence and CT imaging. Journal of Nanobiotechnology, Vol. 16, No. 26, pp. 1–18. DOI: 10.1186/s12951-018-0359-9.

14 **Ferrer, M.M., de Santana, Y.V.B., Raubach C.W., La Porta, F.A., Gouveia, A.F., Longo, E., Sambrano, J.R. (2014).** Europium doped zinc sulfide: a correlation between experimental and theoretical calculations. Journal of Molecular Modeling, Vol. 20, No. 2375, DOI: 10.1007/s00894-014-2375-5.

15 Ekambaram, S.M. (2005). Effect of host-structure on the charge of europium ion. Journal of Alloys and Compounds, Vol. 390, No. 1-3, pp. L2–L3. DOI: 10.1016/j.jallcom. 2004.08.068.

16 Hirata, G., Perea, N., Tejeda, M., Gonzalez-Ortega, J.A., McKittrick, J., (2005), Luminescence study in Eu-doped aluminum oxide phosphors. Optical Materials,

Vol. 27, No. 7, pp. 1311–1315. DOI: 10.1016/j.optmat.2004.11.029.

17 **Hedayati, K., Zendehnam, A., Hassanpour, F. (2016).** Fabrication and characterization of zinc sulfide nanoparticles and nanocomposites prepared via a simple chemical precipitation method. Journal of Nanostructures, Vo. 6, No. 3, pp. 207–212. DOI:10.7508/JNS.2016.03.005.

18 **Pedraza, J., Laverde, D., Mantilla, C. (2008).** Utilización de estudios de potencial zeta en el desarrollo de un proceso alternativo de flotación de mineral feldespático. Dyna, Vol. 75, No. 154, pp. 65–71.

19 **Malvern Instruments Worldwide (2015).** Zeta potential-An introduction in 30 minutes. Technical note.

20 **Hu, Y., Wei Z., Wu B., Dai, Q., Feng, P. (2018).** Photoluminescence of ZnS: Mn quantum dot by hydrothermal method. AIP Advances, Vol. 8, No. 1, pp. 015014. DOI: 10.1063/1.5010833.

21 **Sarkar, R., Tiwary, C.S., Kumbhakar, P., Basu, S., Mitra, A.K. (2008).** Yellow-orange light emission from Mn2+-doped ZnS nanoparticles. Physica E: Low-dimensional Systems and Nanostructures, Vol. 40, No. 10, pp. 3115–3120. DOI: 10.1016/j.physe.2008.04.013.

22 **Lu, S.W., Lee, B.L., Wang, Z.L., Tong, W., Wagner, B.K., Park, W., Summers, C.J. (2001).** Synthesis and photoluminescence enhancement of Mn^{2+}-doped ZnS nanocrystals. Journal of luminescence, Vol. 92, pp. 73–78. DOI: 10.1016/S0022-2313(00)00238-6.

Article received on 14/01/2022; accepted on 27/04/2022.
Corresponding author is Eduardo San Martín-Martinez.