Improving Parameters of the Gamma Associative Classifier using Differential Evolution

Antonio Ramírez-Ramírez[1], Itzamá López-Yáñez[2],
Yenny Villuendas-Rey[3], and Cornelio Yáñez-Márquez[4].

1, 4 CIC, Instituto Politécnico Nacional, México
2 CIDETEC, Instituto Politécnico Nacional, México
3 Facultad de Informática de la Universidad de Ciego de Ávila, Cuba.

Abstract. This paper presents an effective method to improve some of the parameters in an associative classifier, thus increasing its performance. This is accomplished using the simplicity and symmetry of the differential evolution metaheuristic. The Gamma associative classifier, which is a novel associative model for pattern classification, is formed by some parameters, that when modified, it has been found to be more efficient in the correct discrimination of objects; experimental results show that applying evolutionary algorithms models, the desired efficiency and robustness of the classifier model is achieved. In this first approach, improving the Gamma associative classifier is achieved by applying the differential evolution algorithm.

Keywords: pattern classification, metaheuristics, Gamma Associative Classifier, differential evolution

1 Introduction

The standard paradigm of computing, which peaked with countless algorithms for almost all computing tasks has been overtaken by new metaheuristic hybridization techniques. Thus, neural networks, cellular automata, evolutionary strategies and quantum computation [1] [2]; all unconventional computing tasks, have gained new strength to be enhanced with the benefits of metaheuristics. In applying these algorithms in the automated search for values that are most suitable for the parameters of Gamma Associative Classifier (CAG, for its acronym in Spanish), it could maximize it to obtain more accurate classification results. Today the computer systems are facing a major problem is access to large data sets, therefore, the application of new non-traditional techniques of performing the tasks of programming and measurement is required. Such is

the case of several evolutionary algorithms, in particular, the differential evolution algorithm (DE). The latter is an algorithm recently used in unconventional computing area, and has proven to be a good solution for some of the optimization problems facing the paradigms of traditional computing, such as processing large volumes of information, as chemical and biological systems of our environment [3] [4] [5]. DE is a metaheuristic developed by R. Storn and K. Price [6], this algorithm is based on vectors and is cataloged within evolutionary algorithms, since it makes use of cross and mutation operators [7].
DE is used especially in problems involving large search spaces. Unlike genetic algorithms, DE is based on simple addition and subtraction operations, enabling you to simply and exploration of the search space more efficiently. Because of this and their own features as vector components, result in a better solution to specific problems [8].
On the other hand, in the area of pattern recognition (RP) they have handled four major tasks traditionally: clustering, regression, recovery and classification, the latter being one of the best known and addressed subjects [9] - [12]. In academia, have been studied these four tasks in different approaches and areas of application, such as the probabilistic-statistical approach [13], the deterministic approach or based on metric [14], the structural syntactic approach [15], focus or neuronal area [16] and the associative approach [10] [17] [18], among some others. Within the associative approach, approximation models have been developed that allow not only make recovery and pattern classification, but also achieved good results in the regression task, and even address some of the problems of prediction. Recently, the CAG has been applied successfully in some relevant issues like time series prediction in the oil wells production context [19] and the hybrid associative classifier with translation (CHAT by its acronym in Spanish) in medical applications [20].
This article addresses the one-parameter setting, of one of the best associative classifiers, the CAG and the results of the application of differential evolution algorithm (DE) over the parameters of CAG, are also presented [6], in order to adjust these parameters automatically and thus verify the improved performance of model. The increased performance of the classifier was demonstrated when was adjusted its parameters with the metaheuristic.

The rest of the paper is structured as follows: At the Associative Gamma Classifier section, is described the origin and basis of the classifier. In the Parameters subsection of CAG, it briefly describes its operation. In CAG parameter adjustment section it explains and exemplifies a way to manually look for different values for one of its parameters, in order to improve their performance. The central proposal of this work is presented in section looking for the solution vector with (DE). The algorithm description and some

examples are presented on Experimental results section. Finally conclusions are presented.

2 Gamma Associative Classifier

2.1 Gamma operators

The CAG emerged as a need to combine the power and versatility of the Alpha and Beta operators, developed at the Center for Computer Research since more than ten years ago and are the source and foundation of Alfa-Beta associative memories, which have been implemented in several issues of interest in the recovery pattern and pattern classification [21]. These two operators are defined in tabular form, considering the *A={0,1}* and *B={0,1,2}* sets, as shown in **Table 1**.

Table 1. Definition of the Alpha and Beta operators.

α: A×A → B

x	y	α(*x*,*y*)
0	0	1
0	1	0
1	0	2
1	1	1

β: B × A → A

x	y	β(*x*,*y*)
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

These operators, combined with the association model that is based on patterns coding method, known as modified Johnson-Möbius code [23], gave rise to Alfa-Beta bidirectional associative memories [10], the new association method, based one-hot vector classifier [20], as well as the CHAT-OHM method [20]. All these techniques are associative models, along with the CAG, with which they share common foundational elements: both they are based on associative memories and they operate thanks to the Alpha and Beta operators.

In recent years the CAG has been used in significant research as predicting of production of oil wells [19] and in the prediction of atmospheric pollutants [22], among others. This was made possible by its simple but efficient construction, as discussed in [22], along with the original operators Alpha and Beta, the design and operation of the CAG two

additional original operators was involved: the u_β operator, and Gamma operator of similarity generalized.

The unary u_β operator, receives as input an n-dimensional binary vector and delivers an integer number using the following expression:

$$u_\beta = \sum_{i=1}^{n} \beta\,(x_i, x_i) \tag{1}$$

The generalized similarity Gamma operator γ_g receives as input tow binary vectors $\mathbf{x} \in A^n$ and $\mathbf{y} \in A^m$, where $n, m \in \mathbb{Z}^+, n \leq m$, as well a non-negative number θ and delivery as result a binary digit, which is calculated as:

$$\gamma_g(x, y, \theta) = \begin{Bmatrix} 1 & if \quad m - u_\beta[\alpha(x, y) mod 2] \leq \theta \\ 0 & otherwise \end{Bmatrix} \tag{2}$$

That is, γ_g outputs 1 if both vectors differ by a max θ bits, and outputs 0 otherwise. This means that the CAG works on a set of patterns associated with a class called the fundamental set. These patterns $x^i \neq x^j$ meet ∀ i, j ∈ *{1, 2, ..., p}, i ≠ j,* which means that each of these patterns, it is unique and is associated with a single class, depending on the definition of problem addressed.

2.2 Gamma parameters

CAG algorithm consists of several parameters, among which can be highlighted a weights vector called *w*, this vector indicate how much each variable contributes to the decision to discriminate between input patterns. By modifying the values of this array, changes are obtained in the classification results. Other classifier parameters susceptible of modification is θ_0 (initial theta) representing the value that starts to evaluate how different can be two patterns and that gamma operator considers similar instead. This parameter is subject to ρ parameter (stop), which refers to the maximum value allowed to θ and able the algorithm to continue the search for a disambiguation near a border patterns; when theta reaches the value set to stop$(\theta = \rho)$, the CAG will cease iterating, assigning or placing in a class to the input pattern. In addition to the parameter ρ unemployment, CAG also has a parameter ρ_0 (pause), which helps the program to undertake an assessment of the pattern to classify, to determine their belonging to the unknown class.

Other parameters are the input variable d, which is evaluated to decide whether to classify the pattern belongs to an unknown class or to any of the known classes and finally the parameter u, which is the threshold against which d parameter is compared to decide which class the pattern belongs to class. Currently, there are empirical recommendations by the author of the CAG, to assign values to some of their parameters, without there so far exist, some automated method to determine the appropriate values for these parameters. In the flow diagram of Figure 1, we can appreciate the CAG algorithm model, which has highlighted in a different color, the elements (parameters) that may be adjusted.

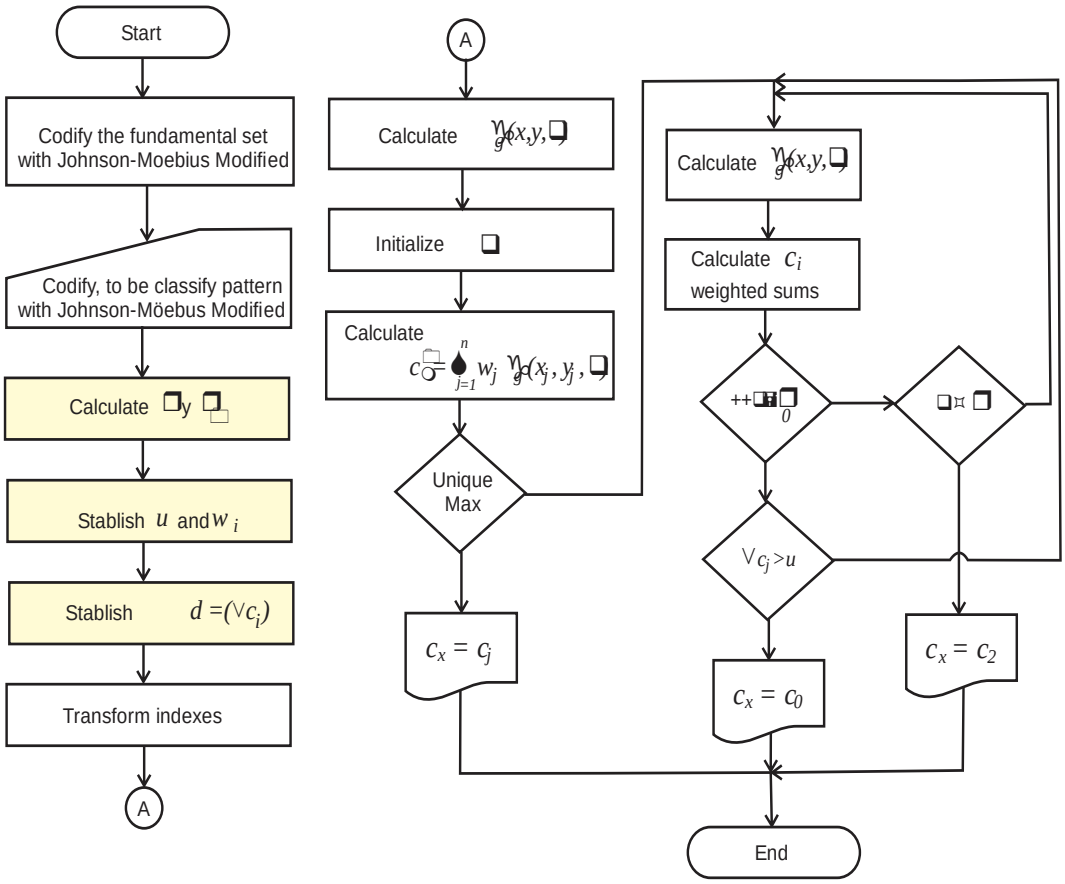

Fig. 1. Flowchart of the learning phase of the CAG.

This model widely discussed in [35], was modified to achieve an improved performance of it. It was decided to apply the metaheuristic on search of better w_i parameter weighting.

2.3 CAG parameters adjustment

In an initial experiment, a series of initial values assigned to the parameters w, θ, ρ, d and u was proposed; for better performance CAG like in [35]. The defaults that have been used for these parameters and have delivered good results in the various problems that have been successful using the CAG are shown in Table 2.

Table 2: Default values of CAG parameters in [35]

$w_i = 1, i = 1, 2, \ldots, n$ θ initialized on 0. $\rho = \bigvee_{j=1}^{n}\left(\bigvee_{i=1}^{p} x_j^i\right)$	$\rho_0 = \bigwedge_{j=1}^{n}\left(\bigvee_{i=1}^{p} x_j^i\right)$ $d = \vee c_i$ $u = 0$

In this initial recommendation proposed by the author of the CAG in [35], it raises a number of initial values that can be checked increased efficiency classifier for some problems. In recent years has been used successfully CAG and there have been some case studies, which have emerged, some empirical rules for the allocation of initial values for these parameters, This values definitely depend on the characteristics of data set to be evaluated. In developing these experiments, it was found that changes improve the performance or in the worst cases, revealed the same conclusions, which had no changes. In Table 3 it can be seen that when modifying the weights of two different data sets (Data Set Iris Wisconsin Breast Cancer and Machine Learning UCI Repository [38]), these changes happen.

Table 3: case 1: Default values, case 2: suggested values in [35].

Data sets	Cases	Weights	Errors	Performance
Iris	1	[1.0,1.0,1.0,1.0]	11	92.67%
	2	[0.5 0.5 1.5 1.0]	7	95.33%
Breast Cancer	1	[1.0,1.0,1.0,1.0, 1.0,1.0,1.0,1.0,1.0]	59	91.55%
	2	[0.5,1.0,1.0,1.0, 1.0,1.5,1.0,1.0,1.5]	59	91.55%

Although apparently it's a very simple example, you can observe that in fact, when the weight vector *w* from CAG, you apply small changes, resulting in a different outcome and check if that change does not improve the original solution, at least retains the level performance. But the question then arises: will exist, and if so, could be automatically find a combination of weights such that depending on the characteristics of the data set needed to evaluate, provide better performance CAG?

3 Looking for the solution vector with DE

The assumptions underlying the central proposal of this article is that by applying a metaheuristic in the search for more precise w_i parameter from CAG, it can automate that search in order to find the most appropriate weighting to achieve better results in pattern classification problems where the AGC is applied.

A metaheuristic that of a simple and low computational cost, would find solutions naturally vectors is Differential Evolution (DE) [6], which has been applied to solve various problems of pattern classification in different tasks [29-34]. This metaheuristic is a direct search method that is robust, easy implementation and has fast convergence [24] [25] and it is highly parallelizable [26-28]. In [36] for example, dramatic results that demonstrate the effectiveness and efficiency of DE, when applied in 18 different problems arise. In these results some variants of the original algorithm, as proposed in [37], wherein the algorithm is improved harmonic DE include search. In the flow chart in Figure 2, it shows the basic operation model of DE algorithm.

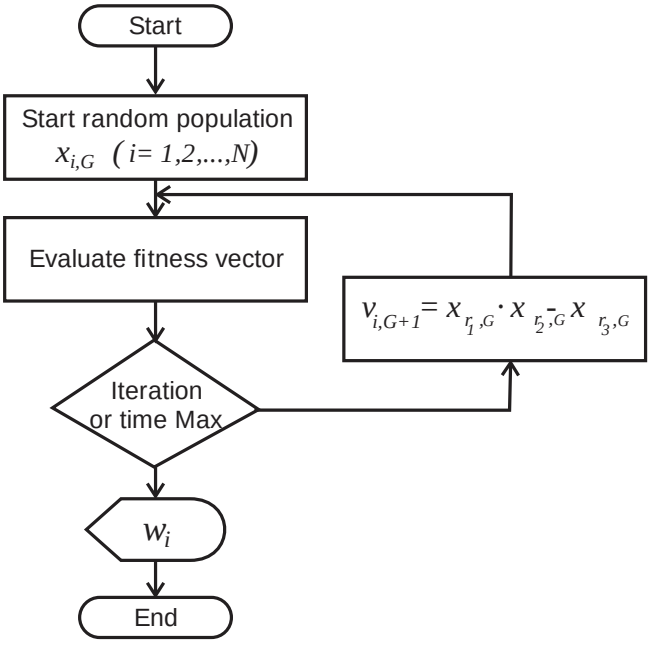

Fig. 2. Flowchart of basic Differential Evolution model.

Basically DE algorithm generates a population of N d-dimensional vectors, considered one generation G; that generation vectors are denoted by:

$$x_{i,G}, i = 1,2,\cdots,N \quad (3)$$

where N do not change across the process and the initial vectors population is automatic selected by randomly. The new vectors of parameters are generated by a process called mutation, it consist on for each element on $G, x_{i,G}, i = 1,2,\cdots,N$ what is an target vector, is determinate one value $F\epsilon(0,2] \subseteq \mathbb{R}$ and three different integer indexes are chosen randomly in pairs, and result on a mutant vector which agree with the expression:

$$v_{i,G+1} = x_{r_1,G} + F \cdot (x_{r_2,G} - x_{r_3,G}) \quad (4)$$

The parameters of the mutated vector is mixed with parameters another predetermined vector (*target vector*) by a process called *cross*, and hence a *vector test*, in which the cost function associated with the problem to be solved is evaluated is obtained. The target, the mutant and test, in the process of crossing, three vectors are involved through a random process. The 'cross' occurs between pairs of real values in the closed interval [0,1] and is determined randomly, evolving a vector of it, generating the weight vector w_i. If the test vector leads to a lower value on the cost function that the target vector, the vector replacing the target test vector in the next generation, thus leading to the *selection*.

There are so many works that offers detailed explanation for implementation of this metaheuristic in literature, as in [6, 26-28, 29-34, 36] and some others. Nevertheless, as commented paragraph above, the metaheuristic is applied on weights vectors population, and then generate an evolved vector that could make better CAG performance. In the Flowchart fragment in Figure 3, change appears on original model that enable apply the metaheuristic to find a better w_i vector and achieve the improved performance of CAG.

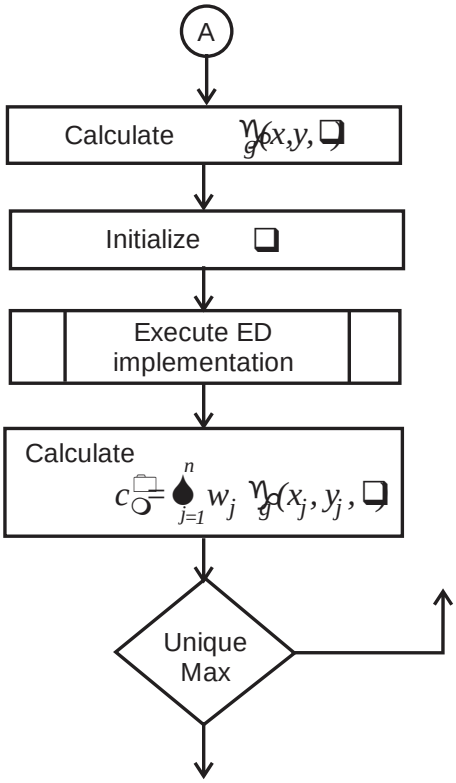

Fig. 3. Flowchart fragment from DE implementation, over CAG model.

4 Experimental Results

When the metaheuristic is performed over one data set, the outputs are a combinations series that conform a vectors list, to be used by CAG and to obtain a better result. The Table 4 shows a vectors list generated in order to apply the CAG on Glass data set.

As can be seen, the values between [0 - 1] for each on vector element, represent the weight of this attribute, that takes to account the CAG to make the classification. That is, on what measure the feature, is considered a weight attribute for this classification.

In the above work it has already demonstrated the feasibility of the algorithm DE, able to find multiple solutions that are at least as good as the best known. In Tables 5 and 6, these results can be seen and check the improvement achieved with relatively low cost, as populations of 100 individuals were used and convergence is reached in less than 50 iterations. In both cases, line 1 and 2 shows the CAG performance with empirical

suggestions as in [35] and the remaining lines, shows performance with w_i vector improved by DE algorithm.

Table 4: Weight vector delivered by DE to apply the CAG.

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
0.677304	0.473165	0.519946	0.020773	0.950655	0.171773	0.581708	0.241777	0.153159
0.407815	0.460470	0.589114	0.689878	0.433206	0.124820	0.694805	0.838947	0.704812
0.003967	0.933099	0.735842	0.963115	0.756653	0.114569	0.874673	0.617319	0.496038
0.958731	0.854590	0.794889	0.766650	0.754062	0.806594	0.468336	0.231333	0.284859
0.689434	0.087731	0.135545	0.320545	0.786423	0.039731	0.336544	0.019812	0.818385
0.419945	0.075037	0.204713	0.989650	0.268974	0.992778	0.449641	0.616982	0.370039
0.762775	0.775271	0.040309	0.610759	0.558764	0.595403	0.558132	0.913882	0.522149
0.105413	0.229669	0.604417	0.347079	0.298745	0.917939	0.911512	0.019474	0.692385
0.836116	0.462810	0.945073	0.900974	0.331106	0.151076	0.779720	0.807953	0.225912
0.432075	0.689603	0.820313	0.289422	0.104743	0.860735	0.204476	0.395016	0.035265

Table 5: Results comparison from table 3 (iris data set) vs results of parameters modified by DE.

Data Set	Try	Weights	Errors	Performance
(previous result with suggested values)	1	[1.0 1.0 1.0 1.0]	11	92.67%
	2	[0.5 0.5 1.5 1.0]	7	95.33%
Iris Data Set Instances: 50 Classes: 3 Dimensions: 4	1	[0.1 0.1 1.5 1.5]	8	94.67%
	2	[0.0 0.0 1.5 1.0]	6	96.00%
	3	[0.0 0.0 2.5 1.0]	5	96.67%
	4	[0.0 0.0 2.5 1.0]	5	96.67%
	5	[0.0 0.0 4.0 2.0]	5	96.67%

Table 6: Results comparison from table 3 (breast cancer Wisconsin data set) vs results of parameters modified by DE.

Data Set	Try	Weights	Errors	Performance
(previous result with suggested values)	1	[1,1,1,1,1,1,1,1,1]	59	91.5594%
	2	[0.5,1,1,1,1,1.5,1,1,1.5]	59	91.5594%
Breast Cancer Wisconsin. Instances: 699 Classes: 2 Dimensions: 9	1	[1,2,2,1.5,1,2,1,1,0.5]	49	92.9900%
	2	[0.5,2,2,1,0.5,2,0.5,0.5,0]	47	93.2761%
	3	[0,2,2,1,0,2,0,0,0]	46	93.4192%
	4	[0.5,4,4,2,0.5,4,0.5,0.5,0]	46	93.4192%
	5	[0,2,2,1,0,2,0,0,0]	46	93.4192%

Although when to apply the same databases that worked in [35], proposal methodology provides good results, the classification was performed improved CAG with on other databases, this way we noted the increased performance of CAG, when combined with the metaheuristic used. In Table 7, one can see the improvement in the classification performed on another three databases by comparing the result of applying the CAG in its original format and with the modified DE.

Table 7: Comparison of results from all five processed data sets with the original CAG and modified by DE

Data set	Features	Instances	Classes	CAG outputs	Weights by DE
iris	4	150	3	0.9533	0.9667
breast-cancer-wisconsin	9	699	2	0.9156	0.9667
diabetes	8	768	2	0.6641	0.6782
glass	9	214	4	0.5318	0.5550
wine	13	178	3	0.8536	0.8592

Conclusions

This paper has presented an analysis that supports the hypothesis that through appropriately modify values of the parameters of the CAG, is possible improve the overall performance of this associative model that is currently at the forefront in terms of recovery methods, classification and regression. The original suggestion, proposed in [35] shows work specifically on these data sets, but nowadays is working in several others data sets with free access to demonstrate the truthiness and performance of this implementation. The substantial contribution lies in the application of a metaheuristic on searching of the correct weight vector instead of manually trying, this improvement was

made possible in this case, through the application of a metaheuristic that continues to work in several worldwide laboratories, who are also at the forefront of research and is known as Differential Evolution.

Acknowledgements The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, COFAA, SIP, CIC, and CIDETEC), the CONACyT, and SNI for their economical support to develop this work, as well as the Informatics Faculty of the University Of Ciego de Ávila Cuba.

References

1. Yaoyao Hea, Qifa Xua, Shanlin Yanga, Aoyang Hanc, Li Yang d, "A novel chaotic differential evolution algorithm cascaded", International Journal of Electrical Power & Energy Systems, Elsevier, Vol. 61, pp. 455-462, China 2014.
2. Eleanor Rieffel and Wolfgang Polak, "An Introduction to Quantum Computing for Non-Physicists", ACM Computation Surveys, FX Palo Alto Laboratory, USA 2000.
3. Adamatzky, et al., "Unconventional Computing 2007", ISBN-13: 978-1-905986-05-7, Luniver Press, USA 2007.
4. K. L. Du, M.N.S. Swamy, "Neural Networks in a Softcomputing Framework", Springer Verlag London Limited, ISBN 1-84628-303-5, London 2006.
5. Enrique Alba, "Parallel Metaheuristics, a New Class of Algorithms" @Zeicience, A John Wiley & Sons, Inc., Publication, España 2005.
6. Rainer Storn and Kenneth Price, "Differential Evolution - A simple and efficient Heuristic for global optimization", Kluwer Academic Publishers, Journal of Global Optimization, Vol. 11, pp. 341–359, Netherlands 1997.
7. Carlos Coello, et al., "Un nuevo algoritmo evolutivo para la optimización de una o varias funciones objetivo sujetas a restricciones", Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería, Vol. 20, pp. 139-167, España 2004.
8. Vasileios K. Karakasis, and Andreas Stafylopatis, "Efficient Evolution of Accurate Classification Rules Using a Combination of Gene Expression Programming and Clonal Selection", IEEE Transactions on Evolutionary Computation, Vol. 12, No. 6, USA 2008.
9. Vidar V. Vikjord, and Robert Jenssen, "Information theoretic clustering using a k-nearest neighbors approach", Pattern Recognition, Elsevier Vol. 47 pp. 3070–3081, USA 2014.
10. María E. Acevedo-Mosqu eda, Cornelio Yáñez-Márquez, Itzamá López-Yáñez, "Alpha–Beta bidirectional associative memories: theory and applications", Neural Processing Letters, Vol. 26 pp 1–40, USA 2007.
11. Kötschau, et al., "Element Pattern Recognition and Classification in sunflowers (Helianthus annuus) grown on contaminated and non-contaminated soil", Microchemical Journal, Elsevier, Vol. 114 pp. 164–174, USA 2014.
12. Nileshkumar H. Jadhav, "Subset selection in multiple linear regression in the presence of outlier and multicollinearity", Statistical Methodology, Elsevier Vol. 19 pp. 44–59, USA 2014.
13. Richard O. Duda, Peter E. Hart and David G. Stork, "Pattern Classification", 2ª. Ed., John Wiley & Sons, Canada 2001.

14. Thomas M. Cover, Peter E. Hart, "Nearest Pattern Classification", IEEE Transaction Information Theory, Vol. 13, no. 1, pp. 21-27, USA 1967.
15. S. Watanabe, S., "Pattern Recognition: Human and Mechanical", Wiley 1st Ed., ISBN: 978-0471808152, New York 1985.
16. J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", Proceedings of the National Academy of Sciences of the United States of America, Vol. 79, No. 8, 2554-2558, USA 1982.
17. Karl Steinbuch, "Adaptive Networks Using Learning Matrices", Kybernetik Band 2, Germany 1964.
18. Humberto Sossa, R. Barrón, R. A. Vázquez, "New Associative Memories to Recall Real-Valued Patterns", A. Sanfeliu et al. (Eds.): CIARP, LNCS 3287, pp. 195–202, Springer-Verlag, Berlin 2004.
19. Itzamá López Yáñez, L. Sheremetov, C. Yáñez-Márquez, "A novel associative model for time series data mining", Pattern Recognition Letters 41, Elsevier B. V., USA 2014.
20. Abril V. Uriarte A., et al., "One-Hot Vector Hybrid Associative Classifier for Medical Data Classification", Plos One, www.plosone.org, Vol. 9, Issue 4, USA 2014.
21. Yáñez-Márquez, J. Luis Díaz de L., "Memorias Asociativas basadas en relaciones de orden y operaciones binarias" Computación y Sistemas Vol. 6 No.4 pp. 300 - 311, *ISSN 1405-5546,* Centro de Investigación en Computación – Instituto Politécnico Nacional, México 2003.
22. Itzamá López Yáñez, Amadeo J. Argüelles-Cruz, Oscar Camacho-Nieto, and Cornelio Yáñez-Márquez., "Pollutants Time-series Prediction using the Gamma Classifier", International Journal of Computational Intelligence Systems, Vol. 4, No. 4, México 2011.
23. Rolando Flores Carapia, "Memorias asociativas Alfa-Beta basadas en código Johnson Möbius modificado", Tesis de Maestría, Centro de Investigación en Computación del Instituto Politécnico Nacional, México 2006.
24. Javier Apolloni, Enrique Alba, "Empirical evaluation of distributed Differential Evolution", standard benchmarks", Applied Mathematics and Computation, Vol. 236, pp. 351-366, Elsevier Inc., España 2014.
25. Yu SUN, Yuanxiang Li, Jun Liu, Gang Liu, "An Improved Differential Evolution Algorithm with Ensemble of Population Topologies", Journal of Computational Information Systems, 8667–8674, Binary Information Press, China 2012,
26. Marina S. Ntipteni, Ioannis M. Valakos and Ioannis K. Nikolos, "An Asyncronous parallel differential evolution algorithm", Department of Production Engineering and Management, Technical University of Crete, Grece 2007.
27. D. Zaharie, D. Petcu, Parallel implementation of multi-population differential evolution, in: Concurrent Information Processing and Computing, pp. 223–232, USA 2005.
28. Marco Tomassini, "Parallel and Distributed Evolutionary Algorithms: A Review" Institute of Computer Science, University of Lausanne, 1015 Lausanne, Switzerland 1999.
29. Kötschau, et al., "Element Pattern Recognition and Classification in sunflowers (Helianthus annuus) grown on contaminated and non-contaminated soil", Microchemical Journal, Elsevier, Vol. 114 pp. 164–174, USA 2014.
30. Eleanor Rieffel and Wolfgang Polak, "An Introduction to Quantum Computing for Non-Physicists", ACM Computation Surveys, FX Palo Alto Laboratory, USA 2000.
31. John R. Koza, "Genetic Programming on the Programming of Computers by Means of Natural Selection", A Bradford Book of the MIT Press, Cambridge, Massachusetts, England 1998.
32. D. B. Fogel and J. W. Atmar, "Comparing Genetic Operators with Gaussian Mutations in Simulated Evolutionary Processes Using Linear Systems", Biological Cybernetics, Vol. 63, pp. 111-114, Springer-Verlag, USA 1990.
33. J. M. García Nieto, E. Alba Torres, G. J. Luque Polo, "Algoritmos basados en Cúmulos de Partículas Para la Resolución de Problemas Complejos" Tesis doctoral, Universidad de Málaga, España 2006.

34. Xin-She Yang, Mehmet Karamanoglu, Tao Luanb, Slawomir Koziel, "Mathematical modelling and parameter optimization of pulsating heat pipes", Journal of Computational Science, Elsevier Vol. 5 pp. 119–125, China 2014.
35. Itzamá López Yáñez, "Teoría y aplicaciones del Clasificador Asociativo Gamma", Tesis Doctoral, Centro de Investigación en Computación del Instituto Politécnico Nacional, México 2011.
36. Huizhi Yi, et al., "Three improved hybrid metaheuristic algorithms for engineering design" Applied Soft Computing Vol. 13 pp. 2433–2444, Elsevier Ltd. USA 2013.
37. T. Warren Liao, "Two hybrid differential evolution algorithms for engineering design optimization", Applied Soft Computing, Elsevier Vol. 10, pp. 188–1199, USA 2010.
38. Bache, M. Lichman, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. University of California, School of Information and Computer Science, Irvine, CA 2013