Automatic Synthesis of Associative Memories by Genetic Programming, a First Approach

Juan Villegas-Cortez[1], J. Humberto Sossa A.[1], Carlos Avilés-Cruz[2] and Gustavo Olague Caballero[3]

[1] Centro de Investigación en Computación - IPN, Mexico D.F.
jvillegas@sagitario.cic.ipn.mx, hsossa@cic.ipn.mx
[2] Universidad Autonoma Metropolitana, Unidad Azcapotzalco, Mexico D.F., Mexico,
caviles@correo.azc.uam.mx
[3] Centro de Investigación Cientfica y de Educación Superior de Ensenada, BC., Mexico.
olague@cicese.mx

Abstract. Associative memories (AMs) are a particular case of artificial neural networks (ANNs), whose main purpose is a faster pattern association than traditional ANN models. This positive attribute has set a new area of research consisting in the application of AMs to specific pattern recognition problems, such as pattern recognition under mixed noise, or real valued patterns recall. Unfortunately every problem has its own difficulties and receive a different treatment, considering the pattern kind or its approach into the specific application scope considering the limited capacity of the AM classic model. Genetic Programming (GP) has proven great success for developping automatic programming, thus creating specialized solutions through the composition of terminals and functions using theory of evolution in order to generate a design based system for problem solving. This work uses GP for the automatic production of AMs in the solution of the classic AM operator for binary patterns problem. We present experimental results on two well-known problems that have been studied by the pattern recognition community (orthogonal and non orthogonal pattern sets association). Our methodology allows us to create novel designs that are different to traditional human-based designs. This work could be considered, as far as we know, as the first approach that applies GP to the synthesis of automatic AM.

1 Introduction

An associative memory (AM) is a special kind of ANN that allows to recall one output pattern from a given input one, such that it can work as a key that may be altered by some kind of noise (salt, pepper or mixed). Several models of AM have been developed during the last years; see: [11], [12], [17], [4], [19], [1], [18], most of them work out fine considering one kind of noise, and only one model works well for the case of mixed noise [23].

Thus, an association between input pattern, x, and output pattern, y, is denoted as (x^k, y^k) where k is the corresponding association. The associative

© G. Sidorov, M. Aldape, M. Martínez, S. Torres. (Eds.)
Advances in Computer Science and Engineering.
Research in Computing Science 42, 2009, pp. 91-102

Received 18/03/09
Accepted 27/04/09
Final version 06/05/09

memory $\boldsymbol{M}$ is represented by a matrix whose components m_{ij} can be seen as the synapses of a neural network. The operator $\boldsymbol{M}$ is generated from an *a priori* set of finite known associations, named as the fundamental set of associations and is represented as $\{(x^k, y^k)|k = 1, ..., p\}$, where p is the number of associations. If $(x^k = y^k)\forall k = 1, \ldots, p$, then $\boldsymbol{M}$ is considered auto-associative, otherwise is hetero-associative. A distorted version of a pattern x to be restored will be denoted as $\widetilde{x}$. If $\boldsymbol{M}$ is fed with a distorted version of x^k and the output obtained is exactly y^k, the recalling feature is perfect. Regarding the components of m_{ij} belonging to $\boldsymbol{M}$, these are conformed by simple operations such as add, multiply, max, min and another defined operators. In order to preserve its simplicity the process of creating new structures for a AM turns out to be complex.

Genetic Programming (GP) is a powerful bio-inspired technique, based on biological evolution theory, that pursues the automation of computer programming using Darwinian principles for the solution of real world and complex problems. It is considered as part of machine learning techniques, and it has also been used to optimize a population of programs using a fitness function that rates the goodness of every program within the population with the goal of solving a specific task. Its origins could be traced back to the application of evolutionary simulation by Barricelli in 1954 [2]. In the 1960's Rechenberg developed its applications for optimization methods, and then, in the 70's, together with his research team solved complex problems using evolutionary strategies [14]. Holland's research had great importance in the early 70's [3]. Smith (1980) and Cramer (1985) presented the first results using GP methodology. In 1981 Forsyth [7], reported little programs evolution in forensic science. Koza [6], is one of the most important exponents of GP and he has been a pioneer in GP applications developed for complex optimizations and has been continuously researching in solutions to several problems.

GP performs the evolution on computer programs (individuals) that are represented as tree structures. These trees are easily evaluated in a recursive way, every node has one function operator, and every terminal node has an operating element, as a result the use of GP in computer programming languages is favored. A set of generated individuals is considered as a generation, every individual represents a solution for a given problem that is meant to be solved. During evolution new individuals are achieved by means of both the selection of the best individuals (although the worst ones have also a little chance of being selected), and their mutual combination for creating new solutions using selection, crossover and mutation operators. After several generations or final convergence criterion based in fitness value, it is expected that the population might contain some good solutions for the addressed problem. Nowadays GP has a wide application covering several environments (see: [9], [8], [20], [21], [22]), its consequent great success comes from its capability of being accurately adapted to numerous problems. Although the main and more direct application is the generation of mathematical expressions, it has been also used in other fields such as rule generation, filter designing, 3D reconstruction models, photogrammetric network design, object recognition, interest point detectors, robotics etc., and finally in a more related field very close to our purposes: the automatic design of ANN [10].

In this paper we try to show the possibility of applying the GP methodology for an AM development and describe the problems inherent to the achievement of our final goal. It is organized as follows. In section 2 a brief description of the representative models of AM is presented. In section 3, we present our GP methodology for automatic synthesis of AM by means of GP. Section 4 shows our results. Finally, conclusions and suggestions for further research are presented in section 5.

2 Representative Models of AM

An AM can be classified into three basic models:

- The Linear Associator (based on the Kohonen correlation matrix [5]).
- The Hopfield Model [4] (based on recurrent type networks where the the units resulting as output are fed back as input units).
- The Bi-directional model (similar to the Linear Associator, but with bidirectional connections, i.e., $w_{ij} = w_{ji}$ for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$).

Based on these primary proposals come other type of models that use mixed ideas in order to get better results for two problems: i) the salt or pepper noise case (the $\alpha\beta$ AM [24], and the Morphological Independent method [12]), and ii) the mixed noise case (The Median AM [1] and The Dynamic AM [23]). These last two proposals have covered gray levels and color images recall problems too. All the above models have one common feature, they try to reach the association with the use of simple operators, mixing the order of application of operators between patterns, or computing new characteristics describing the association. As a final result, the redesigning of new operators of AM is achieved through a process developed in two phases (association and recalling).

3 AM development with the use of GP

The development of AM has been a subject of investigation carried out by human experts, similar to the development of ANN. Further than that, our purpose is to face the problem of automatic synthesis of AM using GP. So far, there is only just one robust design of ANN using GP for a specific problem, Rivero [10], and a preliminary work about the automatic development of kernel pattern design for morphological AM [16], using Genetic Algorithms. Our proposal points towards to an automatic generation of AM, as opposed to the non automatic cases appearing in the reviewed works mentioned in section 2.

As a general rule, the task of AMs generation using evolutionary algorithms could follow the ANNs generation rule. First, the weight evolution starts from an ANN with an already determined topology [25], but in this case we know only one, simple association, using elementary arithmetic operators (having special care with the use of multiply operator with vectors); this is our starting point. Second, the evolution of architectures includes the generation of the topological structure, this means establishing the connectivity for every synapse, in our case, between the components of the input vector and the corresponding output vector. There is a one-to-one correspondence. The key is to keep in mind that the AM holds all the "aspects" of the association, having as input both pattern sets (source and target), during the association phase, and as output just only the target entity resulting from one source pattern, during the recalling phase.

3.1 Model

Our first proposal for a GP-development of AMs, takes into account the main feature of the Linear Associator model [5], that is, the *times*$\{*\}$ operator for performing local associations and the *sum*$\{+\}$ operator for global association; both operations conform the first stage for producing the association matrix M. The *times* operator between the M matrix and an input vector for the recalling process, comprises the second stage. We considered the possibility of taking new evolutionary operators in the local association in order to study all the possible aspects which had not been considered.

In this case, the nodes to be used are the following:

- Op_k. The evolutionary operator (which defines a M_i AM network behavior). It is generated as an individual in the form of a tree. This is the coded genotype.
- x_j and y_j. Nodes that belongs to the *Terminal Set* T. These input nodes are one entry of every pattern-vector of X and Y; so T is defined as: $T = \{x_j, y_j\}$, this is so in order to have a coded correlation input.
- The arithmetic operators, *Functions Set*, (F). They have been constructed considering the possible structures of individuals similar to reviewed models of AMs: $F = \{+, -, min, max, times\}$

The evolutionary process demands the assignation of a *fitness* value to every genotype. Such value is the result obtained after the evaluation of the network with the pattern set representing the problem to be carried out.

For the *Fitness Function*, first we considered the normalized correlation coefficient between the goal (g) and the source processed image (f) [13]. The objective fitness function f_A, known as *similarity* $(0 \leq f_A \leq 1)$, is defined as:

$$f_A = \frac{f \cdot g}{\sqrt{f \cdot f}\sqrt{g \cdot g}} \tag{1}$$

where f and g are two digital images of size $N \times N$, and $f \cdot g$ is given by the equation:

$$f \cdot g = \frac{1}{N} \cdot \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} f(i,j) \cdot g(i,j) \tag{2}$$

This function f_A intends to maximize the number of matching black pixels inside the images f and g, so this seems a reasonable choice for the fitness function. The optimum is found when $f_A = 1$, corresponding to the matching of all pixels. The worst case takes place for $f_A = 0$, when none of the pixels match.

Thus, we will use Eq. (1) as our fitness function, and it will be utilize for the evaluation of every generated individual m_i, applying the association between the source set $\bar{X}$, and the goal set $\bar{Y}$, both of them conform our fundamental set. The evaluation is carried out in three steps. First, we apply the *association* between $\bar{X}$ and $\bar{Y}$. Second, we perform the recalling taking the $\bar{X}$ set while keeping the recalled patterns as $\hat{\bar{Y}}$. Third and final, the computation of Eq. (1) between $\bar{Y}$ and $\hat{\bar{Y}}$, for all the local associations, is carried out in order to have a global fitness. This process is shown in Fig. 1.

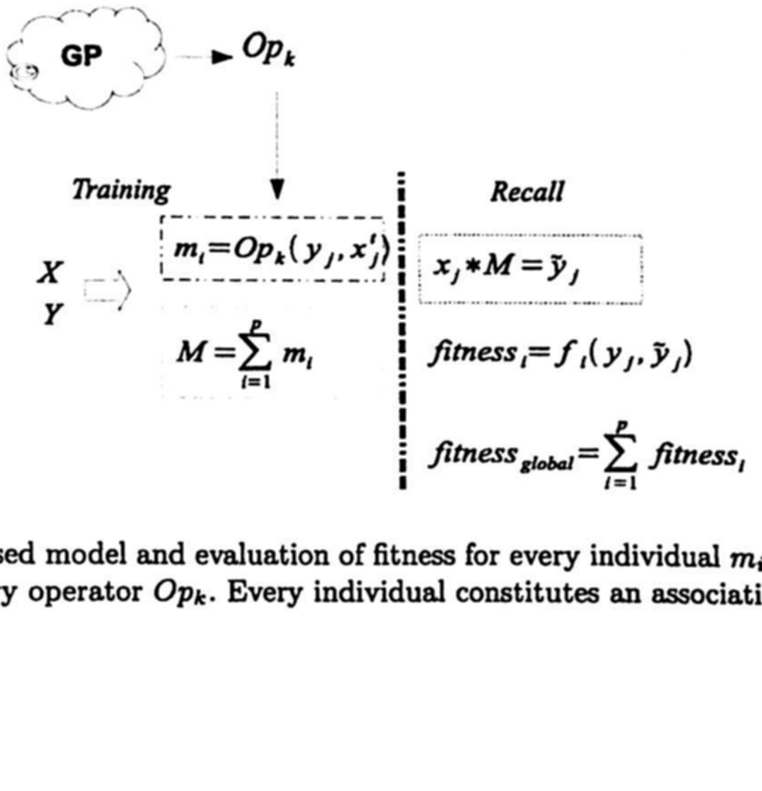


Fig. 1. Proposed model and evaluation of fitness for every individual m_i, created from its evolutionary operator Op_k. Every individual constitutes an associative rule.

3.2 GP Setup

All the experiments were implemented using Matlab with a GPLab toolbox ver. 3.0 [15]. According to the simplicity of the function set we performed several batch of runs consisting of 50 generations, each of them with 70 individuals. After every run we had one possible evolutionary solution for our problem.

The GP parameters used in our experiments are similar to those suggested by Koza [6], taking values of 0.7 for the crossover rate and 0.3 for the mutation rate, respectively. Mutation was based on the *ramped-half-and-half* initialization method, which was also used to initialize the population.

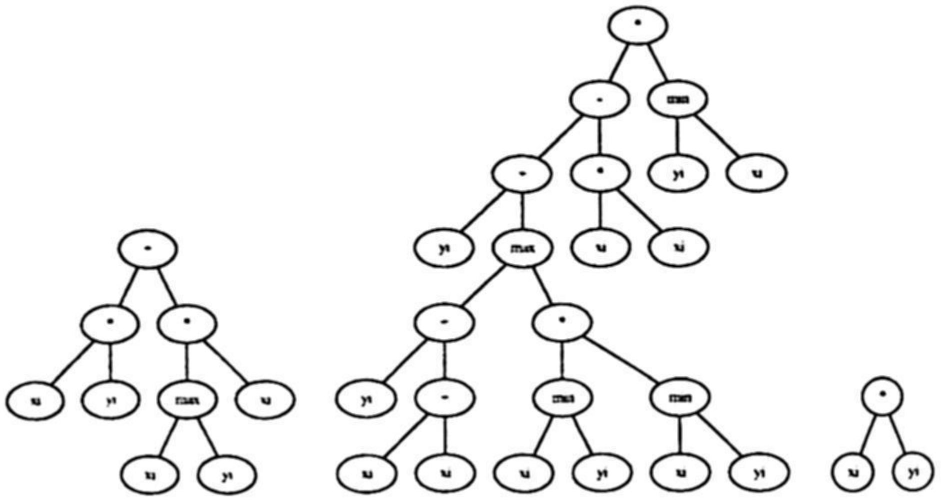
4 Results and Analysis

We applied the above methodology using simple vectors sets in binary sets, converting them to bi-polar notation ($0 = -1$, in order to reduce zeros for *times* operator) as:

$$X = \begin{bmatrix} -1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 \end{bmatrix}, \quad Y = \begin{bmatrix} 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

where X and Y are the source and the goal vector sets, respectively. We tried to find an association between these sets in a hetero-associative memory rule. Every line per matrix is a vector such as $x_1 = [-1\ -1\ -1\ 1]$, it is associated with $y_1 = [1\ -1\ 1\ 1]$ and so on x_2 with y_2, etc.

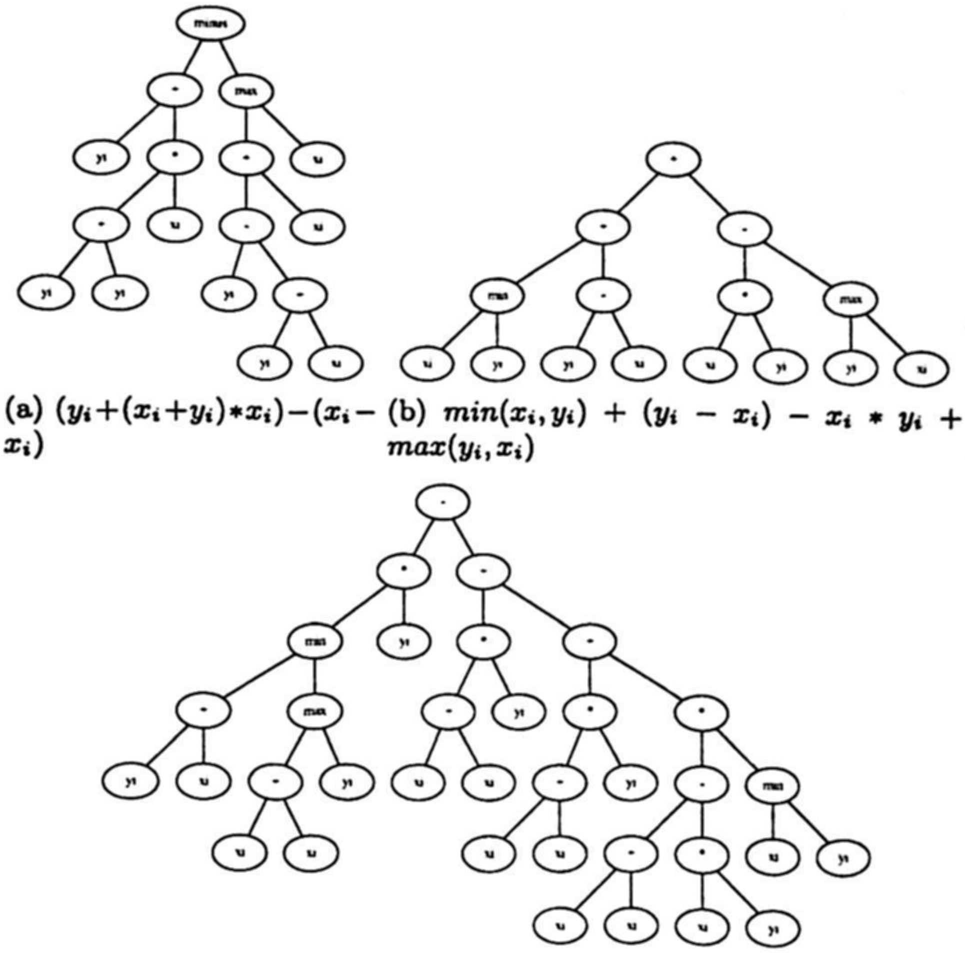
After the accomplishment of the batches described at the beginning of the previous section, under the described conditions, we got several individuals, as those shown in Fig. [2(a), 2(b)], with fitness values of 1. This last value is in concordance with the known result of the classic AM model depicted in Fig. 2(c), which is also one of the various evolutionary results of the same batch process. The selected individuals, shown in the Fig. 2, are just some of the total.



(a) $(x_i * y_i) + (max(x_i, y_i) * x_i)$ (b) $(y + max(y_i + 2x_i, min(x_i, y_i)^2) - {x_i}^2) * min(y_i, x)$ (c) $x_i * y_i$

Fig. 2. Synthesized individuals [(a), (b)] by GP model, and the classic operator (c).

These individuals present two different ways of association between two pattern sets and they make a perfect recall on the fundamental set. In order to test their robustness, we tested them over unknown pattern sets (random and orthogonal patterns generated for auto-associative and hetero-associative cases), taking the number of bits as the total number of vectors per test (ranging from 2 by 2 to 100 by 100, bits and vectors, respectively); all the former with the purpose



(a) $(y_i+(x_i+y_i)*x_i)-(x_i-x_i)$

(b) $min(x_i,y_i)+(y_i-x_i)-x_i*y_i+max(y_i,x_i)$

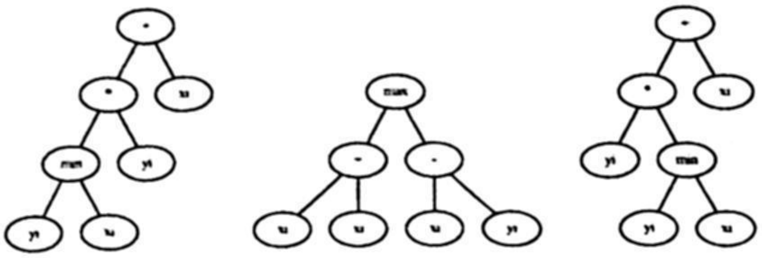
(c) $(min(y_i+x_i,max(x_i+x_i,y_i)))*y_i-((x_i+x_i)*y_i+(x_i+x_i)*y_i+(x_i+x_i-x_i*y_i)*min(x_i,y_i))$

Fig. 3. Synthesized individuals generated from orthogonal pattern sets under auto-associative relationship.

of reviewing the behavior on both cases, auto-associative and hetero-associative. The results are shown in Fig. 5.

Other two tests were performed with this methodology but this time applied for the synthesis of individuals of two more cases: i) the generation of individuals to perform a perfect recall over orthogonal pattern set in auto-associative mode (Fig. 3); and ii) another for random non orthogonal pattern set (Fig. 4). We tested these generated individuals with the same conditions and the same cases described in the previous paragraph. The results are shown in Fig. [6] and Fig. [7].

As we can see, these individuals fit perfectly for their corresponding training pattern sets, but when a new pattern set is associated with them, they do not work in a good manner. On the other hand, when the patterns are evolutionary orthogonal based on hetero-associative AM the individuals work perfectly (Fig. 7), as the classical model does.



(a) $(min(y_i, x_i) * y_i) + x_i$ (b) $max(x_i + x_i, x_i - y_i)$ (c) $(y_i * min(y_i, x_i)) + x_i$

Fig. 4. Synthesized individuals generated from orthogonal pattern sets under hetero-associative relationship.

5 Conclusions

The technique described in this report is suitable to developing simple AMs resulting from GP. These solutions are the outcome of an evolutionary process that explores all the possibilities of association using the described functions and terminals sets. All of this work was carried out by just one single evolutionary process for the local pattern pair association (x_i, y_i).

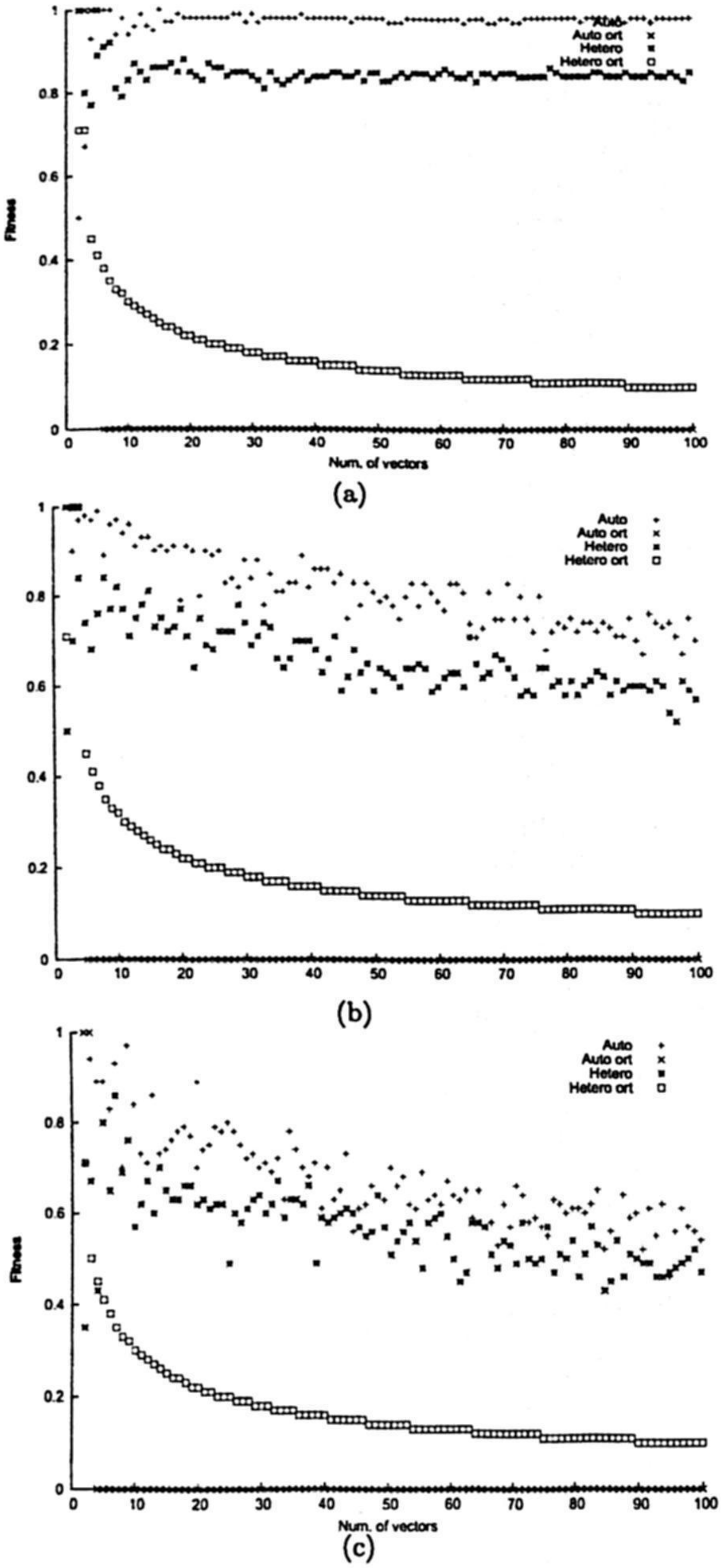
This system shows the following advantage: it is possible the development of AMs through the use of GP. With this model a great amount of possibilities could be explored given the same complexity to the proposed functions and terminal sets. Contrary to most AMs models working with two processes, the association and the recalling, this proposed model works only with the association stage, this could be compared to the Linear Associator AM branch.

Right now we are working in a new model of association taking into account the two processes of the training phase (local and global association), and the recalling process. Our future purpose is to introduce evolutionary operators for every substage (the local evolutionary association covered in this work, plus the global association, and the recalling), in order to have a common solution as the output of the cooperative co-evolution, considering three fundamental aspects of the co–evolutionary process: the individual evaluation, the population selection or variation, and the fitness kind per process. These aspects are shown in the dotted blocks of Fig. 1.

References

1. R. Barrón. Memorias asociativas y redes neuronales morfológicas para la recuperación de patrones. Tesis doctoral. CIC-IPN. 2006.
2. N. Barricelli, Esempi numerici di processi di evoluzione, Methodos, pp. 45-68. 1954
3. J. H. Holland, Adaptation in natural and artificial systems, University of Michigan Press, 1975.
4. J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences. 79: 2554-2558. 1982.

5. T. Kohonen. Correlation Matrix Memories. IEEE Transactions on Computers, Vol. C-21, pp. 353-359. 1972.
6. J. R. Koza. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press. 1992.
7. R. Forsyth, "BEAGLE A Darwinian Approach to Pattern Recognition", Kybernetes, vol. 10, pp. 159–166, 1981.
8. B. Hernández, G. Olague, R. Hammoud, L. Trujillo and E. Romero. Visual learning of texture descriptors for facial expression recognition in thermal imagery. Comput. Vis. Image Underst., Vol. 106, pp. 258–269, 2007.
9. C. Perez B and G. Olague. Learning Invariant Region Descriptor Operators with Genetic Programming and the F-Measure. International Conference on Pattern Recognition. ICPR., 2008.
10. D. Rivero, J. Rabuñal and Alejandro Pazos. Automatic Desing of ANNs by Means of GP for Data Mining Tasks: Iris Flower Classification Problem. ICANNGA 07. Part I, Springer-Verlag LNCS 4431, pp. 276-285. 2007.
11. G. X. Ritter et al. Morphological associatives memories. IEEE Transactions on Neural Networks, vol. 9, no. 2, pp. 281-293, 1998.
12. G. X. Ritter, G. Urcid et al. Reconstruction of patterns from noisy inputs using morphological associative memories. International Journal of Mathematical Imaging and Vision. 19(2), pp. 95-111. 2003.
13. M. Quintana, R. Poli, E. Claridge. Morphological Algorithm Design for Binary Images Using Genetic Programming. Genetic Programming and Evolvable Machines. Vol 7, Issue 1. Pag: 81 - 102. 2006 .
14. I. Rechenberg. Evolutionsstrategie Optimierung technischer Systeme nach Prinzipien der biologischen Evolution (PhD thesis). Reprinted by Fromman-Holzboog. 1973.
15. S. Silva, J. Almeida. GPLAB - A Genetic Programming Toolbox for MATLAB. In Gregersen L (ed), Proceedings of the Nordic MATLAB Conference (NMC-2003), 273-278.
16. A. R. Silva Lavalle. Un Método de Algoritmos Genéticos para Optimización de Memorias Asociativas Morfológicas. Tésis. Univesidad de Puerto Rico. 2006.
17. H. Sossa, R. Barron, R. A. Vazquez. New associative memories for recall real-valued patterns. LNCS, 3287:195-202. 2004.
18. H. Sossa, R. Barrón. Extended $\alpha\beta$ associative memories. Revista Mexicana de Física. 53(1):10-20.
19. P. Sussner. Generalizing operations of binary auto-associative morphological memories using fuzzy set theory. Journal of mathematical imaging and vision, 19(2):81-93. 2003.
20. L. Trujillo, G. Olague. Automated Design of Image Operators that Detect Interest Points. Evolutionary Computation. MIT Press., vol. 16, no. 4, pp. 483-507., 2008.
21. L. Trujillo, G. Olague, F. Fernndez, E. Lutton. Behavior-based Speciation for Evolutionary Robotics. GECCO. Atlanta, GA, USA., 2008.
22. G. Olague, C. Puente. Honeybees as an Intelligent based Approach for 3D Reconstruction. ICPR. Hong Kong, China. 2006.
23. R. A. Vázquez and H. Sossa. A New Model of Associative Memories Network. ANNIP. Angers, France, May 9-12, 2007.
24. C. Yáñez M., J. L. Diaz de León S. Memorias asociativas basadas en relaciones de orden y operaciones binarias. Ph. D. Thesis abstract. Computación y Sistemas, Vol. 6, No. 4 pp. 300-311. 2003.
25. X. Yao. Evolving artificial neural networks. Proceedings of the IEEE, 87(9): 1423-1447, September 1999.

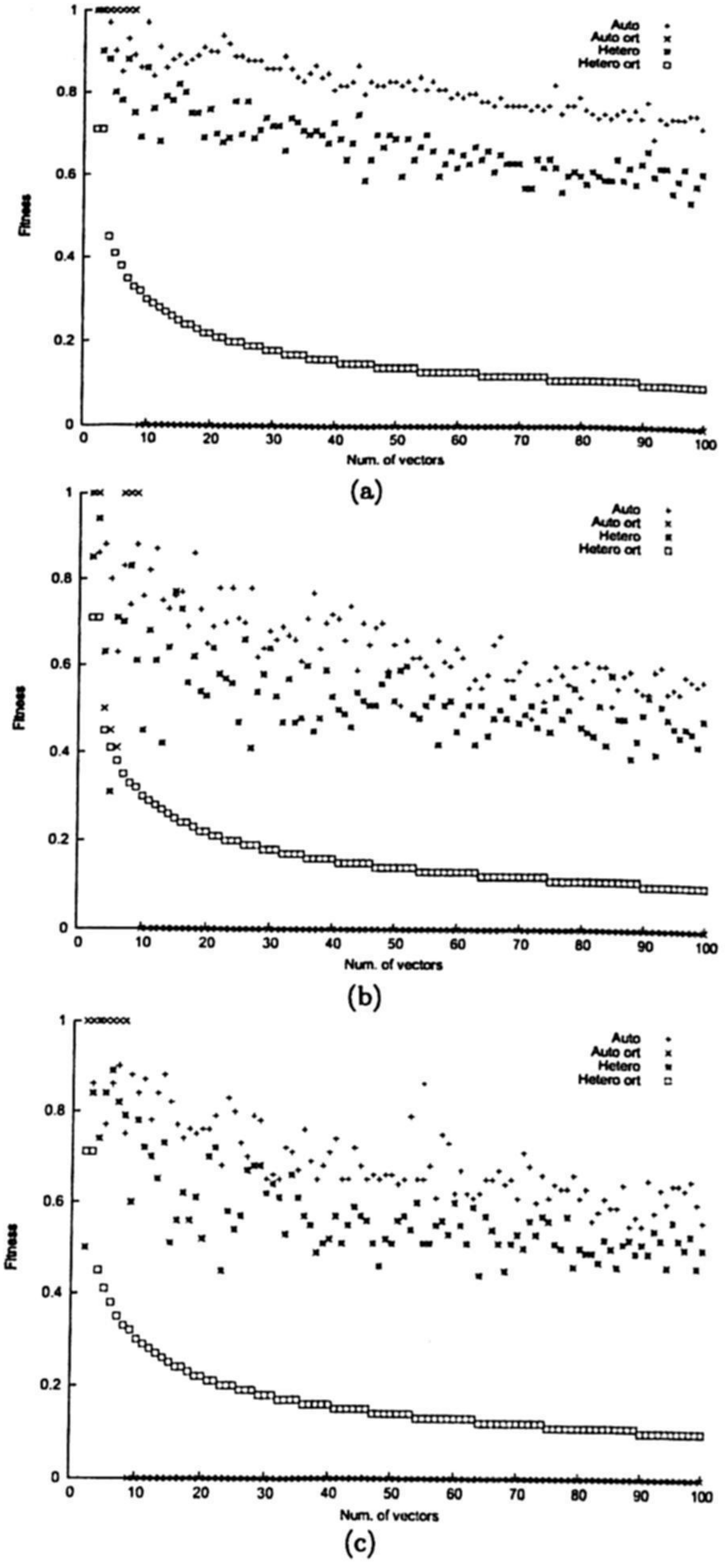


(a)

(b)

(c)

Fig. 5. Test of performance for evolutionary individuals generated from a random non orthogonal pattern sets under hetero-associative association.

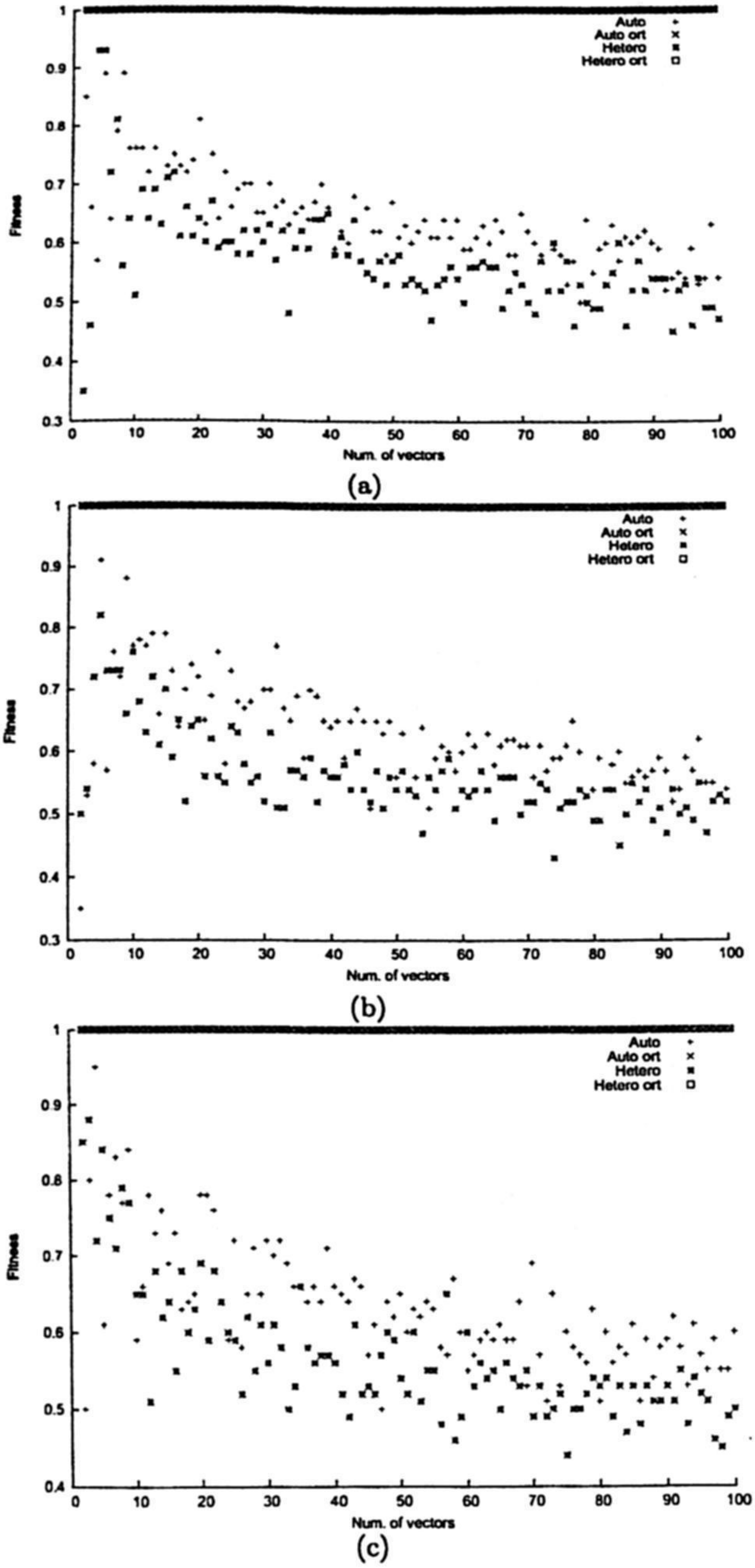


(a)

(b)

(c)

Fig. 6. Test of performance for evolutionary individuals generated from a random non orthogonal pattern sets under auto-associative association.



(a)

(b)

(c)

Fig. 7. Test of performance for evolutionary individuals generated from a random non orthogonal pattern sets under hetero-associative association.