Evolving Fuzzy Systems
A new strategy for rule semantics preservation

Germán L. Osella Massa*, Leonardo Corbalán†,
Laura Lanzarini‡, Armando De Giusti§
Instituto de Investigación en Informática LIDI.¶
Universidad Nacional de La Plata.
{gosella, corbalan,laural,degiusti}@lidi.info.unlp.edu.ar

Abstract

Fuzzy rule-based systems have proved to be a convenient tool for modeling complex systems. This is due to their capacity to capture their typical imprecision, which makes classical methods inefficient. At present, Fuzzy Logic Controllers (FLC) are considered one of the most important applications of fuzzy rule-based systems.

However, the learning process of proper rules for a given problem is still an important research issue. In this direction, different solutions for this problem have been developed, many of them based on Evolutionary Algorithms. Nevertheless, the preservation of fuzzy system rules semantics during the evolving process - more specifically during the recombination - is not always assured.

This paper proposes a codification for fuzzy systems together with the proper genetic operators in order to achieve a balance between the searching process carried out by the evolving algorithm and the preservation of the recombined fuzzy system.

Such codification and the proposed genetic operators have been used in an evolving algorithm, and its behavior in real function approximation has been tested, with successful results.

Keywords: Evolutionary Algorithms, Fuzzy Logic, Unsupervised Learning, Genetic Operators.

*Becario Doctoral de CONICET. Ayudante Dipomado - Facultad de Informática. UNLP
†Becario de Perfeccionamiento de la UNLP. Ayudante Diplomado - Facultad de Informática.
‡Profesor Titular. Facultad de Informática. UNLP
§Investigador Principal del CONICET. Profesor Titular - Facultad de Informática. UNLP
¶Calle 50 y 115 1er Piso, (1900) La Plata, Argentina, TE/Fax +(54) (221) 422-7707. http://lidi.info.unlp.edu.ar

1 Introduction

Fuzzy logic has proved to be an adequate tool for developing solutions to complex problems in several areas, not only within the area of Artificial Intelligence but also as regards decision-making problems [9, 11], specially in control problems [8, 10, 12, 1, 7].

Nevertheless, it is not always simple to design a fuzzy system capable of behaving as expected. The choice of the linguistic concepts in which the range of each variable is divided and their combination into rules constituting the knowledge base may entail a complex task, even when an expert's help is available.

Evolutionary algorithms are searching methods that have been successfully applied into a wide range of problems, being the automatic fuzzy-system generation among them. However, they are general searching methods, reason why they are often strengthened when provided with specific knowledge of the problem to solve. This could be achieved by designing a more appropriate representation for potential solutions or by inserting heuristics typical of the problem into the genetic operators.

An aspect which is not usually taken into account is the use of genetic operators that preserve characteristics [4]. Overlooking such aspect may cause the crossover operator to generate new possible solutions which are rarely connected to those used to produce them, turning it into a macromutation operator [6]. In the specific case of fuzzy systems, this problem becomes evident when trying to combine different fuzzy-system rules barely related among them, but which have been chosen due to the random nature that is typical of this operator.

In this work, a specific representation and genetic operators for fuzzy systems are proposed in order to achieve a balance between the stochastic natured search typical of the evolving algorithm and the preservation of the recombined fuzzy systems, so as to preserve the rule semantics contained in such systems.

This paper is organized in the following sections: Section 2 describes the Takagi-Sugeno type fuzzy systems, which are used in this paper. Section 3 presents the proposed evolving method to counteract the mentioned problems. Section 4 proves the proposed approach through experimental results obtained in a real function approximation problem. Finally, some conclusions are presented in section 5.

2 Takagi-Sugeno Fuzzy Systems

A Takagi-Sugeno system [2] has N input variables $x_1, x_2, \ldots, x_N$, M output variables $y_1, y_2, \ldots, y_M$ and K rules R_k as:

$$R_k: \text{ IF } \mathbf{x}\, is\, S_k \text{ THEN } \mathbf{y}_k = \mathbf{f}_k(\mathbf{x}), \qquad k = 1, 2, ..., K \tag{1}$$

where $\mathbf{x} = (x_1, x_2, ..., x_N)$ is a vector containing the input data, $\mathbf{y}_k = (y_1, y_2, ..., y_M)$ is the output vector, S_k is a fuzzy set, and $\mathbf{f}_k$ is a vector function relating the output variables and the input variables.

In the case of using $\mathbf{f}_k$ lineal functions, rules are obtained with the following structure:

$$R_k: \text{ IF } \mathbf{x}\, is\, S_k \text{ THEN } \mathbf{y}_k = \mathbf{A}_k\mathbf{x} + \mathbf{B}_k \qquad k = 1, 2, ..., K \tag{2}$$

where $\mathbf{A}_k$ and $\mathbf{B}_k$ are matrices of $M \times N$ and $M \times 1$, respectively, containing the coefficients relating the output variables and the input variables. If the rule is expressed in its scalar form instead of using the matrix form, we have:

$$R_k: \text{ IF } \left(\begin{array}{c} x_1\, is\, S_{k1} \\ and\, ...\, and \\ x_N\, is\, S_{kN} \end{array} \right) \text{ THEN } \left\{ \begin{array}{l} y_1^k \;= a_{11}^k x_1 + \ldots + a_{1N}^k x_N + b_1^k \\ \vdots \\ y_M^k = a_{M1}^k x_1 + \ldots + a_{MN}^k x_N + b_M^k \end{array} \right. \tag{3}$$

In order to represent such a fuzzy system, it is enough to specify the fuzzy sets S_j^k of the antecedent and the real coefficients a_{ji}^k and b_j^k ($1 \leq i \leq N$, $1 \leq j \leq M$, $1 \leq k \leq K$) of the consequent of each of the rules of such system.

In this paper trapezoidal-type fuzzy sets (figure 1) are used, which are completely defined with the four values shown in such figure. In addition, there are cases in which the trapezoids turn into triangles, as figure 2 shows.

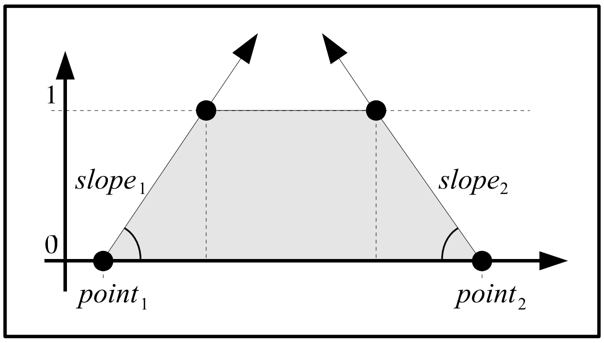

Figure 1: Values defining the trapezoidal fuzzy set: $point_1$ and $point_2$ define the ends of the trapezoid, while $slope_1$y $slope_2$ determine the slope of the corresponding sides.

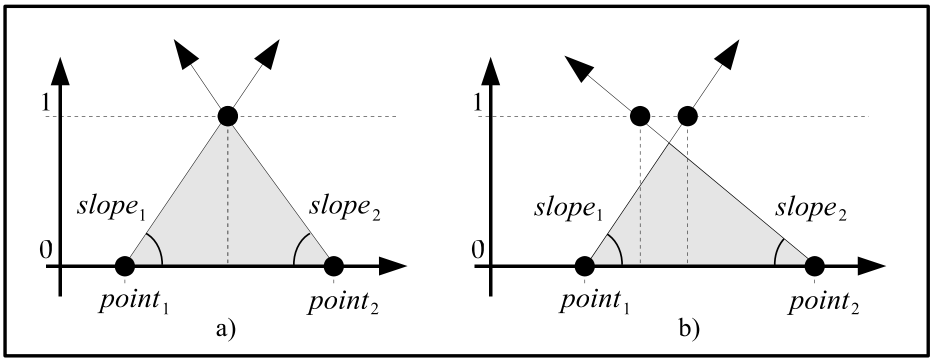

Figure 2: Particular cases allowed by the used fuzzy sets. a) The trapezoid is reduced to a fuzzy triangular set. b) 1 is not necessarily the maximum value.

3 Proposed Evolving Method

The evolving process is based on the Genetic Algorithm proposed in [3], with modifications in the codification of solutions and in the employed genetics operators. The pseudocode is shown in the algorithm 1. The execution begins with a population of fuzzy systems that are generated at random but it is also possible to generate them by making slight changes to a solution designed by a human expert, if there is any. Then, the performance of each possible solution is tested in the resolution of the problem in question, after which a new population is created from the solutions of the existing population applying genetic operators specifically designed for this method. The process goes on until a maximum number of generations is achieved or until the behavior of the best solution outperforms a pre-established minimum.

Algorithm 1 Evolutionary algorithm pseudocode.

```
T = 0
P(T) = Initial population containing N chromosomes
Evaluate P(T)
While (T < Max_Generation) and (Best_Fitness < Minimun_Fitness)
    P(T + 1) = Population with the M best chromosomes from P(T)
    While size(P(T + 1)) < N
        Select Parent1 and Parent2 from P(T)
        Child1, Child2 = Crossover(Parent1, Parent2)
        Add Mutation(Child1) to P(T + 1)
        Add Mutation(Child2) to P(T + 1)
    Evaluate P(T + 1)
    T = T + 1
```

3.1 Classification

In order to preserve the rule semantics when applying the crossover operator, it is necessary to determine the characteristics common to both fuzzy systems. This is not a trivial task at all. To achieve this, it is necessary to observe the fuzzy sets and their participation in each of the rules. The proposed method makes use of a mechanism of fuzzy set classification into *concepts* which eases the comparison of both systems.

A concept represents a group of fuzzy sets expressing truth-values over similar intervals within the range of a variable. Intuitively, if both sets belong to a same concept, they express truths over a similar fact.

The classification mechanism keeps a list of concepts per each input variable. Each concept within each list is characterized by a fuzzy set $\mathbf{S_i}$ representing it.

Every fuzzy set generated within the evolutionary algorithm is compared to each of the representatives $\mathbf{S_i}$ until finding a similar one. If this happens, the set is marked with the concept identifier i. If the concepts are exhausted, without finding a proper one, a new one is created using such fuzzy set as its representative. At the beginning of the classification process, all the concept lists are empty. The lists will grow as fuzzy sets not able to be classified within existing concepts are discovered.

In order to determine whether the fuzzy sets X and Y are similar, the classification process makes use of the function $\delta(X, Y)$, defined in equation (4), in order to compute the difference between them. Such function computes the distance between the points corresponding to the ends of the compared fuzzy sets. These points determine the interval within the range of the variable about which the set has information. If the value computed by $\delta(X, Y)$ is inferior to a certain threshold μ, it is assumed that both sets are similar and belong to a same concept.

$$\delta(X, Y) = |Y_{punto_1} - X_{punto_1}| + |Y_{punto_2} - X_{punto_2}| \tag{4}$$

The threshold μ establishes a tolerance level to the differences between two fuzzy sets of a same variable.

3.2 Fuzzy System Coding

Chromosomes submitted to evolution have a structure as that appearing in figure 3, containing all the necessary information to specify a complete fuzzy system. Such structure is of variable size.

Fuzzy sets are codified in a chromosome, potentially one per each concept of each variable discovered throughout the evolving process. Each coded set belongs to a different concept and is not necessarily equal to the representative of each concept. There can be concepts for which fuzzy sets are not coded and, if they are used in any rule, they are simply ignored. Fuzzy sets are coded through four real values, such as shown in figure 1. As the classification mechanism discovers new concepts, all the chromosomes are automatically expanded to make room for the fuzzy sets associated to such concepts.

The chromosome also has a set of rules, each of them divided in antecedent and consequent, as detailed in equation 3.

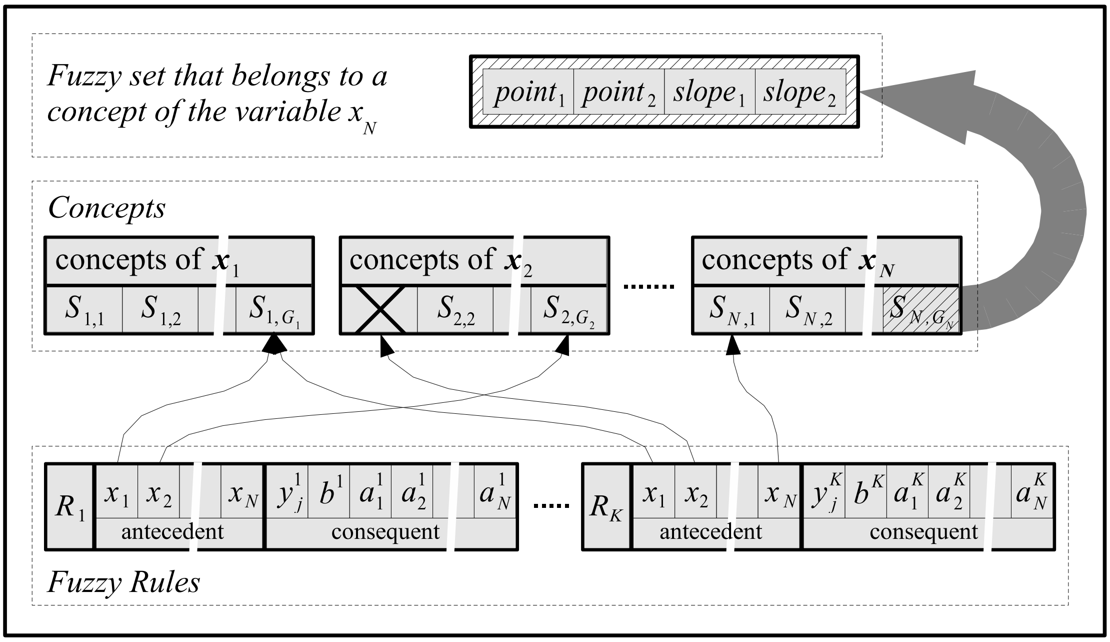

Figure 3: Structure of a chromosome representing a fuzzy system.

The antecedent contains, per each input variable, a reference to a fuzzy set within the list of concepts of each variable. This reference is codified as an integer with sign $i \in [-1, L)$, being L the quantity of fuzzy sets existing in the list. For the case in which $i = -1$, it is considered that there is no reference to any fuzzy set and the variable associated to this reference is eliminated from the antecedent of that rule. During the generation of rules, special care is taken in order to avoid the existence of a rule whose references within the antecedent are all equal to -1. This scheme allows evolving fuzzy sets, while easing at the same time their re-utilization in several rule antecedents, thus preventing the evolutionary algorithm from producing the same set several times.

In the rule consequent it is indicated which output variable y_j is being referenced and which the real coefficients corresponding to the lineal function associating such variable to the input variables $x_1, x_2, \ldots, x_N$ are.

Rules are ordered according to the references to the fuzzy sets, keeping a lexicographical order. The existence of two or more rules with identical references is avoided. These conditions are imposed in order to avoid the existence of redundant rules and to ease the application of the genetic operators, which will be defined in the next section.

The chromosome is of variable length, since both the quantity of represented fuzzy sets and the number of defined rules are free of being modified during the execution of the evolutionary algorithm.

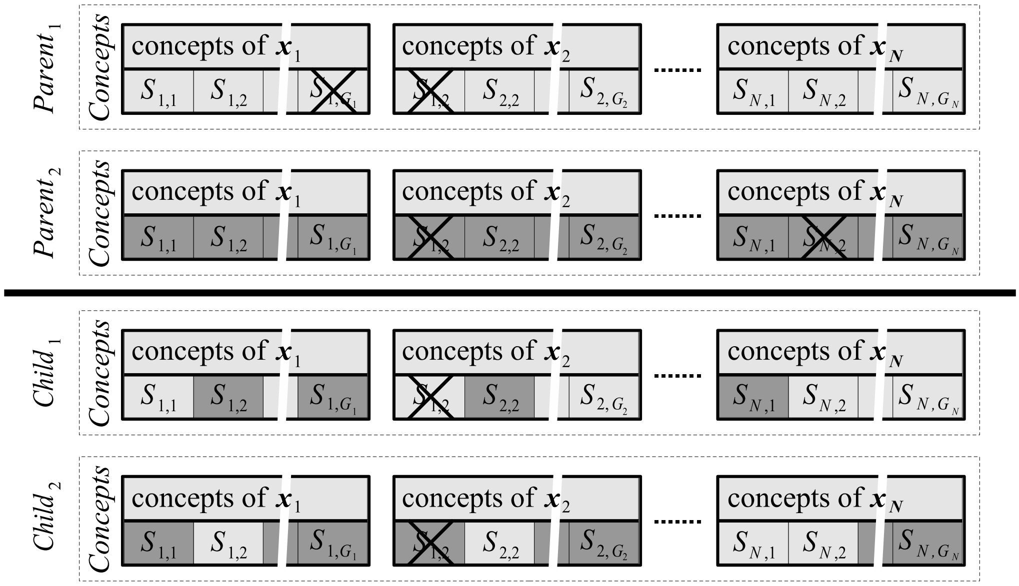

Figure 4: First stage of the crossover operator

However, each rule of the fuzzy system is of fixed length because the quantity of references in the antecedent and the number of real coefficients of the consequent are both values depending on the problem and previously known before the execution of the evolutionary algorithm.

3.3 Genetic Operators

Specific operators have been defined so as to be applied into the chromosomes described in Section 3.2, also making use of the list of concepts in which fuzzy sets are classified.

3.3.1 Crossover

The crossover operator takes two chromosomes as parents and produces two new chromosomes, resulting from the combination of their parents. The crossing is carried out in two stages.

In the first stage (figure 4), an iteration is carried out simultaneously on each parent chromosome over the lists of fuzzy sets. Stochastically, it is determined which son will receive which fuzzy set of each parent. If some parent does not have any fuzzy set defined, the fuzzy set of the other parent is assigned.

In the second stage (figure 5), an iteration is performed simultaneously over the rules of each of the chromosomes. A rule of each parent's is added to each son, determining at random which parent gives its rule to which son. If two rules with the same antecedent are added to a same son, the second overwrites the first one. If the rules of one of the parents are exhausted, the remaining rules from the other parent are added, randomly alternating the son receiving them.

The process finishes when all the rules from both parents are exhausted, giving as a result two new chromosomes. This operator tries to exploit the genetic material of each of the parents, trying to preserve at the same time their common characteristics [4].

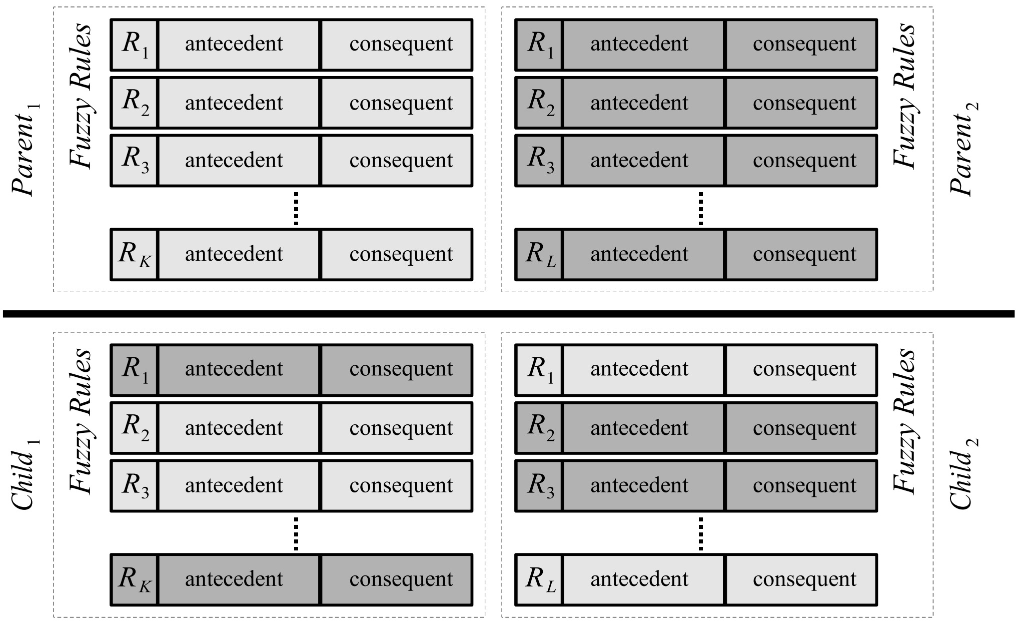

Figure 5: Second stage of the crossover operator

3.3.2 Mutation

The mutation operator produces modifications in fuzzy systems in three different ways and determines the type of change using a uniform probability distribution. This operator acts over the fuzzy sets, over the consequent lineal functions, and over the rules making up the knowledge base.

When it acts over the fuzzy sets, it selects a variable from a rule's antecedent and modifies the referenced fuzzy set, in the following ways:

- It varies the values defining the set (figure 1), altering the slopes or the points that define the ends of the trapezoid. This is carried out in such a way that the modified set is kept in the same concept that included it before undergoing the changes. To achieved this, the operator alters the ends of the set so that the function $\delta(X,Y)$ (equation 4) used in the classification does not surpass the threshold μ when the modified set is compared to the representative of the concept to which it belonged.

- It creates a new set at random for the selected variable, adding it as a concept of that variable. This may replace an existing concept if it was previously defined or it may incorporate it to the concept list if it did not previously exist. The new set is generated in three different ways: Modifying a fuzzy set representative of some of the concepts of the selected variable or creating a new set at random at any interval of the total range of the variable, or at the intervals of the range that are not yet covered by any concept.

When acting over the lineal functions of the consequent, the operator may cause some of these mutations:

- It modifies some of the real coefficients, adding or subtracting random values.

- It replaces the complete function, generating a new one at random. This includes the output variable to which the function is assigned.

Finally, when it acts over the rules, it may

- Add a new rule generated at random, maybe replacing an existing rule if the references of the antecedent of such rule coincide with those of the new produced rule.
- Eliminate some rule of the fuzzy system.
- Alter one of the references of the antecedent of some existing rules, and, once again, it may replace another of the rules contained in the knowledge base if their references coincide.

This operator fulfils the function of introducing new genetic material to the population, thus exploring new areas of the space of possible solutions.

4 Experiments

4.1 Approximation of a two variable function

In the experiments,

$$z = \frac{\sin(x)}{x} . \frac{\sin(y)}{y} \tag{5}$$

was used as test function, selecting 121 points uniformly distributed in the interval $[-3\pi, 3\pi] \times [-3\pi, 3\pi]$. As it is possible that the fuzzy system does not produce a valid response for some of the tested points[1], the system output was computed over the points in which the system generated a valid value. At the same time, the quantity of points in which the system responded and the quantity in which its response was indefinite were counted.

In order to qualify the fitness of the solutions produced by the evolutionary algorithm, we decided to use a variant of the mean square error computation, expressed in the following fitness function:

$$\text{Fitness}(S_k) = c \times \frac{\left(E_{max}^2 - \frac{E}{a}\right)}{1 + b} \tag{6}$$

where c is a scale factor, E_{max} is the maximum square error tolerated in a point, E is the sum of the square of the differences between the fuzzy system response and the correct value rendered by the function, a is the quantity of points in which the systems responded, and b is the quantity of point in which the output was indefinite.

This classification strategy makes the algorithm generate fuzzy systems containing always a response in every of the tested values, rewarding at the same time those which had better approximated the function in question.

4.2 Results

Results were obtained carrying out 30 independent runnings of the algorithm with the parameters detailed in table 1. Figure 6 shows, per each generation, the average of the best population fitness together with the maximum and minimum of the 30 executions. The achieved results are a very good approximation to the function in question. If the fuzzy systems produced by the algorithm are analyzed, we can observe in figure 7 how the number of rules of the best systems obtained in

[1] This happens when none of the system rules covers the values taken by the input variables, and this is very noticeable at the beginning of the evolutionary process.

each generation varies. The algorithm produces compact solutions in terms of the number of rules necessary to approximate the function.

Parameter	Value	Parameter	Value
Population size	200	Generations	500
Selection method	Roulette Wheel	Replacement method	Generational
Crossover probability	80%	Mutation probability	5%

Table 1: Parameters

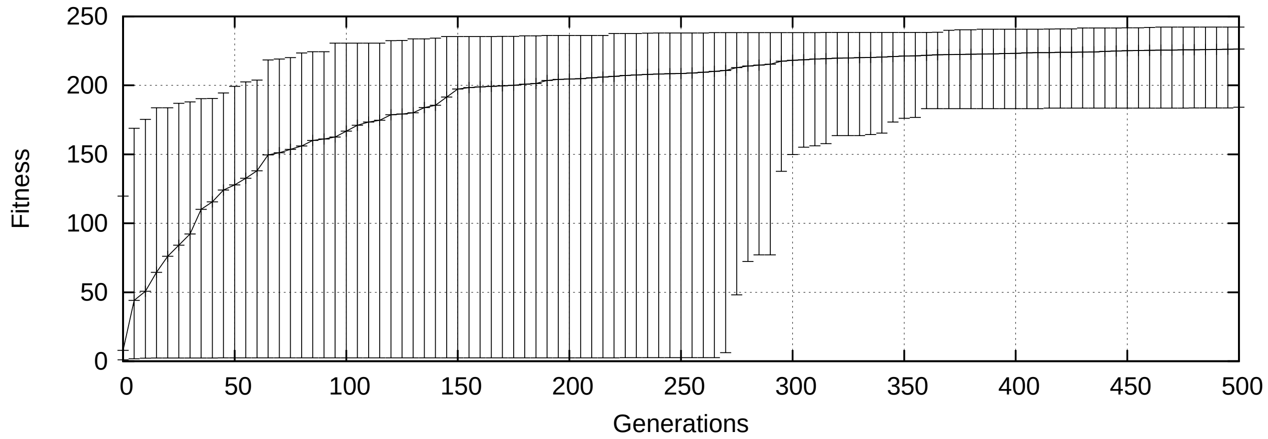

Figure 6: Fitness Evolution.

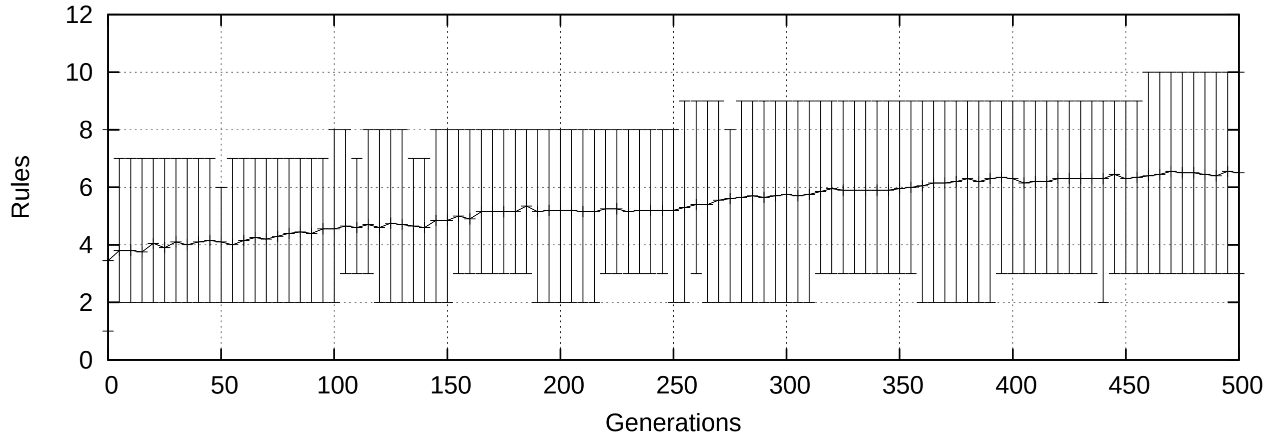

Figure 7: Evolution of the quantity of rules of the best systems of each generation.

5 Conclusions

This paper has presented the proper representation and genetic operators for generating Fuzzy Systems by means of an Evolutionary Algorithm preserving the rule semantics making it up.

Its efficiency has been demonstrated by applying it into the resolution of a concrete problem without introducing any previous knowledge conditioning the structure of the fuzzy set to be used. The results obtained were satisfactory.

At present, we are working on the use of this proposal for obtaining fuzzy controllers applicable in control problems, specially in robotics.

References

[1] Bonissone, P. P., "Fuzzy Logic Controllers: An Industrial Reality", In *Computational Intelligence: Imitation Life*, Ed. by J. M. Zurada et al., IEEE Press, pp 316-327, 1994.

[2] Babuška, R., ”Fuzzy modeling: Principles, methods and applications”. In C. Bonivento, C. Fantuzzi, and R. Rovatti, editors, *Fuzzy Logic Control: Advances in Methodology*, pp 187-220. World Scientific, Singapore, 1998.

[3] Goldberg, D. E., ”Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley Pub. Co., 1989.

[4] Radcliffe, N. J., ”Forma Analysis and Random Respectful Recombination”, ICGA 1991, pp. 222-229, 1991.

[5] Takagi, T. & Sugeno, M., ”Fuzzy identification of systems and its application to modeling and control”, IEEE Trans. Systems, Man and Cybernetics 15(1), pp. 116-132, 1985.

[6] Jones, T., ”Evolutionary Algorithms, Fitness Landscapes and Search”, PhD thesis, University of New Mexico, Alburquerque, 1995.

[7] Kosko, B., ”Neural Networks and Fuzzy Systems”, Prentice-Hall, 1991.

[8] Y. Ono, H. Uchiyama, W. Potter, "A mobile robot for corridor navigation: a multi-agent approach". Proceedings of the 42nd annual Southeast regional conference, Huntsville, Alabama, Pages: 379 - 384 , April 2004 , ISBN:1-58113-870-9.

[9] Cox, Earl, "A Fuzzy System for Detecting Anomalous Behaviors in Healthcare Provider Claims", en Goonatilake, Suran & Treleaven, Philip (Eds). (Intelligent Systems for Finance and Business, John Wiley & Sons, 1996, pp. 111-134.

[10] Carlos Dualibe, Paul Jespers, Michel Verleysen. "On Designing Mixed-Signal Fuzzy Logic Controllers as Embedded Subsystems in Standard CMOS Technologies". Proceedings of the 14th symposium on Integrated circuits and systems design. IEEE Computer Society Washington, DC, USA . Page: 194. Year of Publication: 2001 , ISBN:0-7695-1333-6.

[11] Ahmet Çinar. Fuzzy blending of materials. Proceedings of the 2nd international conference on Computer graphics and interactive techniques. Pages: 162 - 166 Year of Publication: 2004 ISBN:1-58113-883-0 ACM Press New York, NY, USA.

[12] Gang Chen, Zhonghua Yang, Hao He, Kiah Mok Goh. Proceedings of the second international joint conference on Autonomous agents and multiagent systems. Pages: 960 - 961 Year of Publication: 2003. ISBN:1-58113-683-8. ACM Press New York, NY, USA.

[13] F. Herrera, M. Lozano and J. L. Verdegay. "A learning process for Fuzzy Control Rules using Genetic Algorithms". Technical Report DECSAI-95108. Universidad de Granada. Spain. 1995.

[14] Tzung-Pei Hong, Chai-Ying Lee. "Learning fuzzy knowledge from training examples". Proceedings of the seventh international conference on Information and knowledge management. Bethesda, Maryland, United States Pages: 161 - 166 . Year of Publication: 1998. ISBN:1-58113-061-9. ACM Press New York, NY, USA.