# Fuzzy Genetic Algorithms: Issues and Models*

**F. Herrera, M. Lozano**
Dept. of Computer Science and A.I.
University of Granada, 18071 - Granada, Spain
e-mail: {herrera,lozano}@decsai.ugr.es

## Abstract

*There are two possible ways for integrating Fuzzy Logic and Genetic Algorithms. One involves the application of Genetic Algorithms for solving optimization and search problems related with fuzzy systems. The another, the use of fuzzy tools and Fuzzy Logic-based techniques for modeling different Genetic Algorithm components and adapting Genetic Algorithm control parameters, with the goal of improving performance. The Genetic Algorithms resulting from this integration are called Fuzzy Genetic Algorithms.*

*In this contribution, we tackle Fuzzy Genetic Algorithms by analyzing their definition based on the Zadeh's concept of Fuzzy Algorithms and the two different meanings as Fuzzy Logic may be viewed. We review different approaches, attempt to identify some open issues and summarize a few new promising research directions on the topic.*

**Keywords:** Fuzzy Logic, Genetic Algorithms, Fuzzy Algorithms, Fuzzy Genetic Algorithms.

*This research has been supported by CICYT TIC96-0778.

# 1 Introduction

Recently, numerous papers and applications combining Fuzzy Logic (FL) and Genetic Algorithms (GAs) have become known, and there is an increasing interest in the integration of these two topics (Bäck & Kursawe, 1994; Cordon, Herrera & Lozano, 1997; Herrera & Verdegay, 1996; Sanchez, Shibata & Zadeh, 1996).

The FL and GAs integration has been accomplished by following two different approaches: 1) the application of GAs in optimization and search problems related with fuzzy systems, and 2) the use of fuzzy tools or Fuzzy Logic-based techniques for modeling different GA components or adapting GA control parameters, respectively, with the goal of improving performance. Generally, GAs resulting from the last type of integration are called *Fuzzy* GAs (FGAs). Different GA issues are concerned by this approach:

- *Adaptation of GA control parameters.* Fuzzy Logic Controllers may be used for dynamically computing appropriate GA control parameters using the experience and knowledge of the GA experts in order to induce suitable exploitation/exploration relationships throughout the GA execution for avoiding the premature convergence problem (Arnone, Dell'Orto & Tettamanzi, 1994; Bergmann, Burgard & Hemker, 1994; Herrera, Herrera-Viedma, Lozano & Verdegay, 1994; Herrera & Lozano, 1996a; Lee & Takagi, 1993; Lee & Takagi, 1994; Xu & Vukovich, 1993; Xu, Vukovich, Ichikawa, Ishii, 1994).

- *Crossover operators.* Fuzzy connectives and triangular probability distributions may be considered for designing powerful crossover operators that establish adequate population diversity levels and so attacking the premature convergence problem as well (Herrera & Lozano, 1996b; Herrera, Lozano & Verdegay, 1996; Herrera, Lozano & Verdegay 1997; Voigt, 1995).

- *Representation.* The classical binary representation (genes with values of zero or one) may be generalized into a fuzzy representation (genes with values between zero and one), in order to include some more complex genotype versus phenotype relationships similar to the ones occurring in nature (Voigt, 1992; Voigt, Born & Santibáñez-Koref, 1993).

- *Stopping Criteria.* Belief and uncertainty measures may be taken into account for handling solution predictions in order to force the GA to reach optimal solutions with a user-defined accuracy, stooping the run when this is accomplished (Meyer & Feng, 1994).

In this paper, we attempt to lay the foundations of FGAs by developing three important tasks. First, we study the definition of FGAs based on the Zadeh's concept of Fuzzy Algorithms and the two different meanings as FL may be viewed, a narrow interpretation and its wide sense (Zadeh, 1993), trying to identify the general context of FGAs. From this study, two FGA approaches arise: FGAs concerning to the narrow interpretation of FL and ones concerning to the wide interpretation. The first ones involve the adaptation of GA control parameters by means of Fuzzy Logic Controllers. The second ones include the previous and GAs whose elements are designed using fuzzy tools. Then, we review several FGAs approaches that have emerged in the GA literature. Finally, we attempt to identify some open issues and summarize a few new promising research directions for FGAs.

The paper is set up as follows: in Section 2, we tackle with possible FGA definitions, in Section 3, we survey different GA components designed using fuzzy tools, in Section 4, we deal with Fuzzy Logic Controllers-based adaptive GAs, in Section 5, we discuss future directions and some challenges for FGAs, and finally, in Section 6, we comment on some concluding remarks.

# 2 Fuzzy Genetic Algorithm Definition

In this section, we study the definition of FGA from two different points of view: one following the Zadeh's definition of Fuzzy Algorithm (Subsection 2.1) and another involving the two different meanings as FL may be viewed, a narrow interpretation and its wide sense (Subsection 2.2).

## 2.1 FGA Definition Derived from the Concept of Fuzzy Algorithm

Essentially, a Fuzzy Algorithm (Zadeh, 1968) is an ordered sequence of instructions (like a computer program) in which some of the instructions may contain labels of fuzzy sets, e.g.:

Reduce $x$ *slightly* if $y$ is *large*

Increase $x$ *very slightly* if $y$ is *not very large and not very small*

If $x$ is *small* then stop; otherwise increase $x$ by 2.

By allowing an algorithm to contain instructions of this type, it becomes possible to give an approximate fuzzy-algorithmic characterization of a wide variety of complex phenomena (Zadeh, 1973).

According to this classical definition of Fuzzy Algorithm, an FGA will correspond to a GA incorporating this type of information, and basically, the FGA concept will include adaptive GAs based on the use of fuzzy rule-based systems for controlling GA control parameters in order to adjust them dynamically, under the consideration of accepting the use of a fuzzy inference system for managing a fuzzy rule base. Therefore, the main idea would be centered in the use of Fuzzy Logic Controllers (FLCs) (see Appendix A for a short introduction) whose inputs are any combination of GA performance measures or current control parameters and whose outputs are GA control parameters. Current performance measures of the GA are sent to the FLCs, which compute new control parameters values that will be used by the GA.

## 2.2 FGA Definitions Under the Narrow and Wide Interpretations of FL

FL may be viewed from two different meanings as was expounded in Zadeh (1993), a *narrow* and a *wide* interpretation.

> *"(a) a narrow interpretation, FLn, in which fuzzy logic is basically a logic of approximate reasoning; and (b) a wide interpretation, FLw, in terms of which fuzzy logic is coextensive with the theory of fuzzy sets, that is, classes of objects in which the transition from membership to nonmembership is gradual rather than abrupt".*

The FGA definition proposed in Subsection 2.1 corresponds to a narrow interpretation of FL, as we may observe with the following definition (Zadeh, 1993):

> *"In this narrow sense, fuzzy logic may be viewed as a generalization and extension of multivalued logic. But the applicability of fuzzy logic is far than that of multivalued logic because FLn provides many concepts and techniques which are not a part of multivalued logic. Among such concepts and techniques - which play a key role in the applications of fuzzy logic - are those of the linguistic variable; the concepts of possibility and necessity; concepts of truth and usuality-qualification; fuzzy quantification and cardinality. Furthermore, the agenda of FLn is quite different from that of classical multivalued logical systems".*

Considering the FL in its wide sense (Zadeh, 1993),

> *"In its wide sense, FLw, fuzzy logic is a very broad theory with many branches, among them fuzzy sets, fuzzy arithmetic, fuzzy mathematical programming, fuzzy pattern recognition, fuzzy control, fuzzy probability theory, fuzzy topology, the calculi of fuzzy rules and fuzzy graphs, and fuzzy logic, FLn, in its narrow sense. It should be noted that there is a growing trend to interpret fuzzy logic in its wide sense since the label fuzzy logic is more euphonious and more self-explanatory than fuzzy set theory. Regardless of its interpretation, the role model for fuzzy logic is the human mind".*

and according to an extension of the basic Fuzzy Algorithm definition based on the FL in its wide sense, the definition of FGA may be extended. An FGA may be defined as an ordering sequence of instructions in which some of the instructions or algorithm components may be designed with tools based on FL, such as, fuzzy operators and fuzzy connectives for designing genetic operators with different properties, Fuzzy Logic control systems for controlling the GA parameters according to some performance measures, fuzzy stop criteria, representation tasks, etc.

Hence, we present the following FGA definition:

> *An FGA is considered in the wide sense of FL as a GA that uses Fuzzy Logic-based techniques or fuzzy tools to improve the GA behavior modeling different GA components.*

On other hand, in the specialized literature some publications have appeared concerning FGA concepts (for example see Buckley & Hayashi (1994)), which present the use of GAs in imprecise environments, managing fuzzy information. In this way, FGAs are GAs for producing approximate solutions to fuzzy optimization problems where the problem variables translate imprecision or ambiguity into the measurement of the chromosome.

We understand that these GAs are not FGAs, they manage fuzzy objectives and the fitness associated to the chromosomes is defined by a fuzzy number. Therefore, it is necessary to define a method to obtain the selection probabilities from the fuzzy fitness of the chromosomes, for example as the ones proposed in Buckley & Hayashi (1994) and Herrera, Lozano & Verdegay (1994), and then, they may manage the fuzzy optimization problems.

# 3 GA Components Based on Fuzzy Tools

In this section, we review different GA components built using fuzzy tools appeared in the GA literature. Most of them are proposals for genetic operators. This is the case of the Fuzzy Connective-Based Crossover operators (Subsection 3.1), the Dynamic FCB-Crossover operators (Subsection 3.2), the *Soft* genetic operators (Subsection 3.3), and the *Fuzzy Crossover Based on Fuzzy Templates* (Subsection 3.4), which were presented for real-coded GAs (RCGAs) (see Herrera, Lozano & Verdegay (1998)). The remaining approaches include the Fuzzy Representation (Subsection 3.5) and the Fuzzy Stop Criterium (Subsection 3.6).

## 3.1 Fuzzy Connective-Based Crossover Operators

GA behavior is strongly determined by the balance between exploiting what already works best and exploring possibilities that might eventually evolve into something even better. This balance between the creation of diversity and its reduction, by focusing on the individuals with greater fitness, is essential in order to achieve a reasonable behavior for GAs in the case of complicated optimization problems (Bäck, 1994). When this balance is disproportionate, the *premature convergence problem* (a premature stagnation of the search) will probably appear, causing a drop in the GA efficacy, since the search is likely to be trapped in a local optimum before the global optimum is found.

The power of GAs arises from their crossover operator. It may be considered to be one of the algorithm's defining characteristics, and it is one of the components to be borne in mind to improve the GA behavior (Liepins & Vose, 1992). The crossover operator is a determinant element in the exploration/exploitation balance kept by GAs, thus it may be taken into account for solving the premature convergence problem. In this way, solutions to this problem have been found by designing crossover operators that allow suitable levels of exploration and exploitation to be established.

In Herrera, Lozano & Verdegay (1997), the development of such crossover operators was attempted. It was presented the *Fuzzy Connective-Based crossovers* (FCB-crossovers), crossover operators for RCGAs based on the use of fuzzy connectives: t-norms, t-conorms and average functions. Furthermore, a set of offspring selection mechanisms were proposed, which choose the chromosomes (produced by the FCB-crossover operators) that will be the population members. Different exploration and exploitation degrees may be introduced with the FCB-crossover operators and the offspring selection mechanisms establish a relationship between these properties so that they induce different diversity levels in the population and therefore the premature convergence problem may be eradicated.

Next, we describe the FCB-crossover operators. In order to do this, we follow two steps: in Subsection 3.1.1, we define functions for the combination of genes, and in Subsection 3.1.2, we use these functions to define crossover operators between two chromosomes.

### 3.1.1 Functions for the Combination of Genes

Let us assume that $X = (x_1 \dots x_n)$ and $Y = (y_1 \dots y_n)$ ($x_i, y_i \in [a_i, b_i] \subset \Re$, $i = 1 \dots n$) are two real-coded chromosomes that have been selected to apply the crossover operator to them. In short, the action interval of the genes $x_i$ and $y_i$, $[a_i, b_i]$, may be divided into three intervals, $[a_i, x_i]$, $[x_i, y_i]$ and $[y_i, b_i]$, that bound three regions to which the resultant genes of some combination of $x_i$ and $y_i$ may belong. Figure 1 shows this graphically.

These intervals may be classified as exploration or exploitation zones as is shown in the Figure 1. The interval with both genes being the extremes is an exploitation zone and the two intervals that

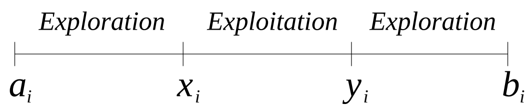


**Figure 1.** Action interval for $x_i$ and $y_i$

| Family | T-norm | T-conorm | Averaging Function. $(0 \leq \lambda \leq 1)$ |
|---|---|---|---|
| Logical | $T_L(x,y) = \min(x,y)$ | $G_L(x,y) = \max(x,y)$ | $P_L(x,y) = (1-\lambda)x + \lambda y$ |
| Hamacher | $T_H(x,y) = \frac{xy}{x+y-xy}$ | $G_H(x,y) = \frac{x+y-2xy}{1-xy}$ | $P_H(x,y) = \frac{1}{\frac{y-y\lambda-xy+x\lambda}{xy}+1}$ |
| Algebraic | $T_A(x,y) = xy$ | $G_A(x,y) = x + y - xy$ | $P_A(x,y) = x^{1-\lambda}y^{\lambda}$ |
| Einstein | $T_E(x,y) = \frac{xy}{1+(1-x)(1-y)}$ | $G_E(x,y) = \frac{x+y}{1+xy}$ | $P_E(x,y) = \frac{2}{1+(\frac{2-x}{x})^{1-\lambda}(\frac{2-y}{y})^{\lambda}}$ |

**Table 1.** Families of fuzzy connectives

remain on both sides are exploration zones. Therefore, exploration and/or exploitation degrees may be assigned to any crossover operator for RCGAs with regard to the way in which these intervals are considered for generating genes.

Under this idea, in Herrera, Lozano & Verdegay (1997), three monotone and non-decreasing functions are proposed: $F$, $S$ and $M$, defined from $[a,b] \times [a,b]$ into $[a,b]$, $a, b \in \Re$, and which fulfill:

$$\forall c, c' \in [a,b],\ F(c,c') \leq \min\{c,c'\},\ S(c,c') \geq \max\{c,c'\} \text{ and } \min\{c,c'\} \leq M(c,c') \leq \max\{c,c'\}.$$

Each one of these functions allows us to combine two genes giving results belonging to each one of the aforementioned intervals. Therefore, each function will have different exploration or exploitation properties depending on the range being covered by it.

Fuzzy connectives, t-norms, t-conorms and averaging functions (Mizumoto, 1989a; Mizumoto, 1989b), were used to obtain $F$, $S$ and $M$ functions. They are defined from $[0,1] \times [0,1]$ into $[0,1]$ and fulfill: 1) t-norms are less than the minimum, 2) t-conorms are greater than the maximum and 3) averaging functions are between the minimum and the maximum. $F$ was associated to a t-norm $T$, $S$ to a t-conorm $G$, and $M$ to an averaging operator $P$. In order to do so, a transformation of the genes to be combined is needed, from the interval $[a,b]$ into $[0,1]$ and later, the result into $[a,b]$. Four families of fuzzy connectives were used for obtaining $F$, $S$ and $M$ functions, which are shown in Table 1. These fuzzy connectives accomplish the following property:

$$T_E \leq T_A \leq T_H \leq T_L \leq P_H \leq P_A \leq P_E \leq P_L \leq G_L \leq G_H \leq G_A \leq G_E.$$

### 3.1.2 $F$-, $S$- and $M$-Crossover Operators

Now, if $Q \in \{F, S, M\}$, we may generate the offspring $Z = (z_1 \ldots z_n)$ as

$$z_i = Q(x_i, y_i), \quad i = 1 \ldots n.$$

This crossover operator applies the same $F$, $S$ or $M$ function for all the genes in the chromosomes to be crossed. For this reason, they were called $F$-crossover, $S$-crossover and $M$-crossover, respectively. Four families of FCB-crossover operators may be obtained using the families of fuzzy connectives in Table 1. Each one of them is called the same as the related fuzzy connective family.

These crossover operators have different properties: the $F$-crossover and $S$-crossover operators show exploration and the $M$-crossover operators show exploitation. According to the associated property of the families of fuzzy connectives in Table 1, the degree in which each crossover operator shows its related property, depends on the fuzzy connective on which it is based. On the one hand, the Einstein $F$- and $S$-crossover operators show the maximum exploration, whereas the Logical ones represent the minimum exploration. On the other, the Logical $M$-crossover operator shows the maximum level of exploitation since it uses the maximum level of information from both genes, i.e., it is not biased towards either of them. The effects of these crossover operators along with their associated exploration or exploitation degrees may be observed in Figure 2.

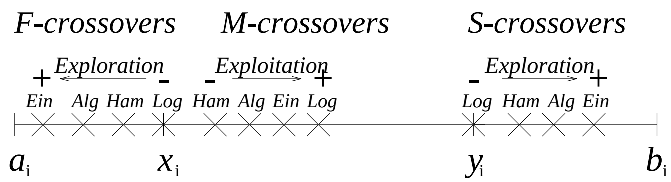


**Figure 2.** FCB-crossover operators

The application of the FCB-crossover operators to the population was made through different strategies. The most suitable one was the following: for each pair of chromosomes from the total population that undergoes crossover, four offspring are generated, the result of applying an $F$-crossover, an $S$-crossover, and two $M$-crossovers to them. Both M-crossovers are based on the same averaging operator, however the parameter $\lambda$ of one of them is $1 - \lambda$ of the other. The offspring selection mechanism chooses the two most promising offspring of the four to substitute their parents in the population.

Different experiments were carried out comparing these proposals with other crossover operators existing in the literature. The results showed a good performance of the proposals. In particular, a suitable behavior was achieved by using the Logical FCB-crossovers along with the application strategy presented above. However, other FCB-crossovers introducing higher diversity levels stood out for very complex functions. For these functions, high diversity levels are fundamental for avoiding the premature convergence problem.

In Herrera & Lozano (1996b) and Herrera, Lozano & Verdegay (1996), $F$, $S$ and $M$ functions were used for building *Heuristic FCB-crossovers*. They produce a child each whose components are closer to the corresponding component of its fitter parent. Two types were presented: *Dominated Heuristic crossovers* that attempt to introduce a useful diversity into the GA population, and *Biased Heuristic crossovers* that induce a biased convergence led towards the best elements. An important property of these operators is that they seems to induce a similar *crowding* effect (Mahfoud, 1992) on multimodal functions, since they generate offspring very similar to one of the parents (the best).

## 3.2 Dynamic FCB-Crossover Operators

A promising idea to avoid the premature convergence problem consists of allowing the exploration in the beginning of the search process and the exploitation at the end of it. Taking as a base the results of the FCB-crossovers regarding to the diversity levels using different fuzzy connectives, in Herrera, Lozano & Verdegay (1996), *Dynamic FCB-crossovers* were proposed based on the use of parameterized fuzzy connectives. These operators keep a suitable sequence between the exploration and the exploitation along the GA run: *"to protect the exploration in the initial stages and the exploitation later"*.

In order to build Dynamic FCB-crossover operators, two steps are followed: in Subsection 3.2.1, we define function families for the combination of genes and in Subsection 3.2.2, we use these families to design dynamic crossover operators.

### 3.2.1 Function Families for the Combination of Genes

Let's consider $x_i, y_i \in [a_i, b_i]$ ($x_i \leq y_i$) two genes to be combined during a generation $t$ and $g_{max}$ be the maximum number of generations. With regard to the three associated intervals, $[a_i, x_i]$, $[x_i, y_i]$ and $[y_i, b_i]$, in Herrera, Lozano & Verdegay (1996), three families of functions were proposed: a family of $F$ functions, $\mathcal{F} = (F^1, ..., F^{g_{max}})$, a family of $S$ functions, $\mathcal{S} = (S^1, ..., S^{g_{max}})$, and a family of $M$ functions $\mathcal{M} = (M^1, ..., M^{g_{max}})$, which for $1 \leq t \leq g_{max} - 1$ and $\forall c, c' \in [a, b]$ fulfill:

$$F^t(c, c') \leq F^{t+1}(c, c') \text{ and } F^{g_{max}}(c, c') \cong min(c, c'),$$

$$S^t(c, c') \geq S^{t+1}(c, c') \text{ and } S^{g_{max}}(c, c') \cong max(c, c'),$$

$$M^t(c, c') \geq M^{t+1}(c, c') \text{ or } M^t(c, c') \leq M^{t+1}(c, c') \ \forall t,$$

$$\text{and } M^{g_{max}}(c, c') \cong M_{lim}(c, c'),$$

| Types | Par. t-norms | Par. t-conorms |
|---|---|---|
| *Frank* $(q > 0)$, $(q \neq 1)$ | $T_F^q(x,y) = log_q[1 + \frac{(q^x-1)(q^y-1)}{q-1}]$ | $G_F^q(x,y) = 1 - log_q[1 + \frac{(q^{1-x}-1)(q^{1-y}-1)}{q-1}]$ |
| *Dombi* $(q > 0)$ | $T_{Do}^q(x,y) = \frac{1}{1+\sqrt[q]{(\frac{1-x}{x})^q + (\frac{1-y}{y})^q}}$ | $G_{Do}^q(x,y) = 1 - \frac{1}{1+\sqrt[q]{(\frac{x}{1-x})^q + (\frac{y}{1-y})^q}}$ |
| *Dubois* $(0 \leq q \leq 1)$ | $T_{Du}^q(x,y) = \frac{xy}{x \vee y \vee q}$ | $G_{Du}^q(x,y) = 1 - \frac{(1-x)(1-y)}{(1-x)\vee(1-y)\vee q}$ |

**Table 2.** Parameterized t-norms and t-conorms

where $M_{lim}$ is an $M$ function called $M$ limit function. We shall denote $\mathcal{M}^+$ or $\mathcal{M}^-$ an $\mathcal{M}$ function family fulfilling the first and the second part of the last property, respectively.

$\mathcal{F}$ and $\mathcal{S}$ functions families may be built using a parameterized t-norm $T^q$ converging on the Logical t-norm and a parameterized t-conorm $G^q$ converging on the Logical t-conorm, respectively. Three families of parameterized fuzzy connectives were used for obtaining $\mathcal{F}$ and $\mathcal{S}$ function families, which are shown in Table 2.

$\mathcal{M}$ function families may be obtained using parameterized averaging operators. An example of these operators is:

$$\forall x, y \in [0,1],\, P^q(x,y) = \sqrt[q]{\frac{x^q + y^q}{2}} \quad -\infty \leq q \leq \infty. \tag{1}$$

### 3.2.2 $\mathcal{F}$-, $\mathcal{S}$- and $\mathcal{M}$-Crossover Operators

Now, let's consider $X^t = (x_1^t \dots x_n^t)$ and $Y^t = (y_1^t \dots y_n^t)$ two chromosomes that were selected, in the generation $t$, to apply the crossover operator to them. If $O^t$ is the $t$-th function belonging to an $\mathcal{F}$ or $\mathcal{S}$ or $\mathcal{M}$ function family, then we could generate an offspring $Z^t = (z_1^t \dots z_n^t)$ as

$$z_i^t = O^t(x_i^t, y_i^t), \quad i = 1, ..., n.$$

If, during the GA run, we use the functions $Q^s$ for $s = 1, ..., g_{max}$ belonging to an $\mathcal{F}$ or an $\mathcal{S}$ or an $\mathcal{M}$ family, as described earlier, then we shall call this type of crossover operator $\mathcal{F}$-crossover or $\mathcal{S}$-crossover or $\mathcal{M}$-crossover, respectively. The $\mathcal{F}$-crossovers and the $\mathcal{S}$-crossovers show exploration properties, whilst the $\mathcal{M}$-crossovers show exploitation properties. We may observe in Figure 3 the effects of these operators when they are applied to the genes $x_i$ and $y_i$ and $M_{lim} = \frac{x_i + y_i}{2}$.

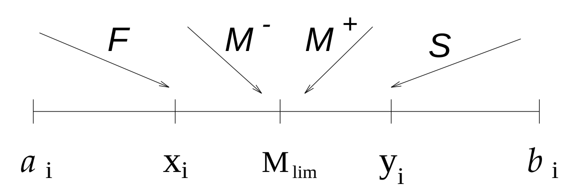


**Figure 3.** Dynamic FCB-Crossover operators

Three families of $\mathcal{F}$- and $\mathcal{S}$-crossovers may be obtained using the parameterized t-norms and t-conorms in Table 2: the Frank, Dombi and Dubois ones, furthermore, an $\mathcal{M}$-crossover may be defined using (1).

Different applications for these crossover operators were presented. The most suitable one was the following: for each pair of chromosomes from the total population that undergoes crossover, four offspring are generated, the result of applying an $\mathcal{F}$-crossover, an $\mathcal{S}$-crossover, an $\mathcal{M}^+$-crossover and an $\mathcal{M}^-$-crossovers to them. The offspring selection mechanism choose the two most promising offspring of the four to substitute their parents in the population.

An empirical study of these operators was made. In general, the best results were returned using the Dubois Dynamic FCB-crossovers. An important property of these operators is that their convergence speed towards the Logical FCB-crossovers is the fastest, which are very suitable, as mentioned in Subsection 3.1.2.

In Herrera & Lozano (1996b) and Herrera, Lozano & Verdegay (1996), $\mathcal{F}$, $\mathcal{S}$ and $\mathcal{M}$ function families were used for designing *Dynamic Heuristic FCB-crossovers.* These operators put together the heuristic properties and the features of the Dynamic FCB-crossover operators. They showed very good results as compared with other crossover operators proposed for RCGAs, even than the FCB-crossover operators and the Dynamic ones.

## 3.3 Soft Genetic Operators

In Voigt (1995), Voigt & Anheyer (1994) and Voigt, Mühlenbein & Cvetković (1995), crossover and mutation operators were presented, which are based on the use of triangular probability distributions. These operators, called *Soft Modal* crossover and mutation, are a generalization of the discrete crossover operator and the BGA mutation, respectively, proposed for the *Breeder* GA (Mühlenbein & Schlierkamp-Voosen, 1993). The term Soft is gleaned from fuzzy set theory only to grasp the main idea, since probability distributions are considered instead of membership functions. Next, we describe their main features.

### 3.3.1 Soft Modal Crossover Operator

In Soft Modal crossover (Voigt, Mühlenbein & Cvetković, 1995) the probability that the $i$-th gene in the offspring has the value $z_i$ is given by the distribution $p(z_i) \in \{\phi(x_i), \phi(y_i)\}$, where $\phi(x_i)$ and $\phi(y_i)$ are triangular probability distributions having the following features ($x_i \leq y_i$ is assumed):

| Triangular Prob. Dist. | Minimum Value | Modal Value | Maximum Value |
|---|---|---|---|
| $\phi(x_i)$ | $x_i - d \cdot \lvert y_i - x_i \rvert$ | $x_i$ | $x_i + d \cdot \lvert y_i - x_i \rvert$ |
| $\phi(y_i)$ | $y_i - d \cdot \lvert y_i - x_i \rvert$ | $y_i$ | $y_i + d \cdot \lvert y_i - x_i \rvert$ |

where $d \geq 0.5$ under its initial formulation (in Herrera & Lozano (1998), its range is constrained to $[0, 1]$). Figure 4 outlines an example of applying this crossover operator for the case of $d = 0.5$. It is clear that the greater the $d$ value is, the higher the variance (diversity) introduced into the population.

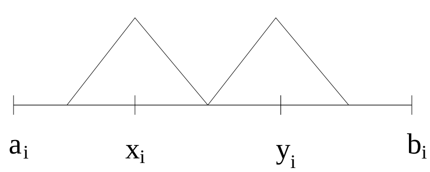


**Figure 4.** Soft Modal crossover operator ($d = 0.5$)

### 3.3.2 Soft Modal Mutation Operator

If $x \in [a, b]$ $(a, b \in \Re)$ is a real-coded gene to be mutated, then Soft Modal mutation (Voigt & Anheyer, 1994) generates a gene, $x'$, as $x' = x + \Delta$, where $\Delta$ is randomly chosen from the following set of probability distributions:

$$p(\Delta) \in \pm\{\phi(b_m^{\pi}) \cdot A_m, \phi(b_m^{\pi+1}) \cdot A_m, \ldots, \phi(b_m^{0}) \cdot A_m\},$$

where $A_m = R_m \cdot (b - a)$ ($R_m$ is set usually to 0.1) and $\pi = \lfloor \frac{\log(R_{min})}{\log(b_m)} \rfloor$, with $b_m > 1$ being a parameter called the base of the mutation and $R_{min}$ the lower limit of the relative mutation range. $\phi(z_k)$ is a triangular probability distribution with

$$\frac{b_m^k - b_m^{k-1}}{2} \leq z_k \leq \frac{b_m^{k+1} - b_m^k}{2}.$$

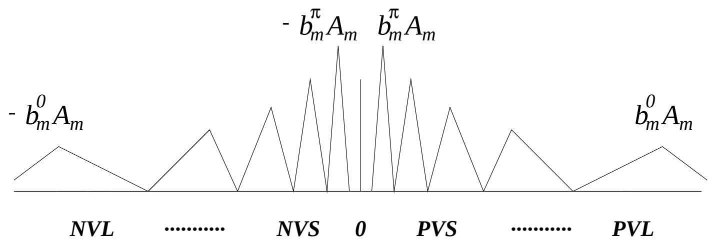


**Figure 5.** Soft Modal mutation operator

Figure 5 shows the possible change produced by this operator. NVS and PVS denote negative and positive very small mutations, respectively, and NVL and PVL negative and positive very large mutations, respectively.

In Voigt (1995), a performance comparison was made between an Evolutionary Algorithm based on Soft genetic operators and the Breeder GA. With respect to robustness and performance the former was superior.

## 3.4 Fuzzy Crossover Using Fuzzy Templates

In Sanchez (1993), a fuzzy crossover was proposed for working with chromosomes consisting of strings of real numbers in the interval $[0, 1]$, which were used for problems where fuzzy sets need to be represented. For presenting such a crossover operator first it was suggested that the classical one-point crossover operator under binary coding may be defined by means of *templates*. A binary chromosome has associated a set whose elements are the loci of the genes with value one. For example, the chromosome $C = (0\ 1\ 1\ 1\ 0\ 1)$ has associated the set $SC = \{2, 3, 4, 6\}$. Supposing two binary chromosomes, $C_1 = (b_1^1 \dots b_n^1)$ and $C_2 = (b_1^2 \dots b_n^2)$ to be crossed, with $SC_1$ and $SC_2$ being their associated sets, an instance of the application of the one-point crossover operator on these chromosomes, e.g. $H_1 = (b_1^1 \dots b_i^1\ b_{i+1}^2 \dots b_n^2)$ and $H_2 = (b_1^2 \dots b_i^2\ b_{i+1}^1 \dots b_n^1)$, $i \in \{1, ..., n-1\}$, may be expressed in terms of the associated sets, $SH_1$ and $SH_2$, by means of another set, $T = \{i + 1, ..., n\}$, which was called *template*, as follows:

$$SH_1 = (SC_1 \cap \bar{T}) \cup (SC_2 \cap T) \text{ and } SH_2 = (SC_1 \cap T) \cup (SC_2 \cap \bar{T}),$$

with $\bar{T}$ being the complement of $T$ in all the possible loci set.

The idea is to generalize these issues for the case in which all sets considered are fuzzy, i.e., the chromosomes $C_1$ and $C_2$ are vectors of real numbers in [0, 1] and the templates are also fuzzy sets of loci, i.e., *fuzzy templates*. Under this condition the previous computation of $SH_1$ and $SH_2$, considering that operators $\cup$ and $\cap$ are fuzzy ones, becomes a *Fuzzy Crossover Operator*. The effect of using fuzzy templates, instead of crisp ones, is to smooth the fuzzy set represented in the fuzzy chromosomes generated through crossover. Other ways to define fuzzy operators may be considered, such as $SC_1 \cup SC_2$ and $SC_1 \cap SC_2$.

## 3.5 Fuzzy Representation

Classical Evolutionary Algorithms, such as GAs and Evolution Strategies (Schwefel, 1995), do not take into account the *development* of an individual or organism from the gene level to the mature phenotype. There are no one-gene, one-trait relationships in natural evolved systems. The phenotype varies as a complex, non-linear function of the interaction between underlying genetic structures and current environmental conditions. Nature follows the universal effects of *pleiotropy* and *polygeny*. Pleiotropy is the fact that a single gene may simultaneously affect several phenotype traits. Polygeny is the effect when a single phenotypic characteristic may be determined by the simultaneous interaction of many genes (Fogel, 1994). An attempt to deal with more complex genotype/phenotype relations in Evolutionary Algorithms was presented in Voigt (1992) and Voigt, Born & Santibáñez-Koref (1993). A *fuzzy representation* is proposed for the case of tackling optimization problems of parameters with variables on continuous domains. Each problem parameter has associated a number of $m$ *fuzzy decision variables* belonging to the interval $[0, 1]$. The chromosomes are formed by the link of the values of the decision variables in each parameter. For each parameter, the decoding process is carried out using a function $g : [0,1]^m \to [0,1]$, and a lineal transformation from the interval [0,1] to the corresponding parameter domain. As an example of such a function the authors presented the following:

$$\forall d = (d_1, ..., d_m) \in [0,1]^m, \quad g(d) = \frac{1}{2^{m-1} - 1} \sum_{j=1}^{m} d_j 2^{j-1}.$$

When $m > 1$, this coding type breaks the one-to-one correspondence between genotype and phenotype (assumed by the classical Evolutionary Algorithms), since two different genotypes may induce the same phenotype. So, it is impossible to find inferences from phenotype to genotype, i.e., the mapping from genotype to phenotype is not *isomorphic*.

We should point out that this attempt is very similar to the one presented in the previous subsection. In Sanchez (1993), chromosomes are considered as sets. Under the present approach they are considered as binary decisions. Both works generalize these ideas to fuzzy tools, i.e, fuzzy sets and fuzzy decisions, respectively.

In Voigt (1992), a crossover operator, called *fuzzy min-max-recombination*, was presented for working under the fuzzy representation. It uses the union and intersection operators of fuzzy sets for defining all possible children. Given two arrays of fuzzy decision values, $D_1 = (d_1^1 \dots d_n^1)$ and $D_2 = (d_1^2 \dots d_n^2)$, first, the min- and max-fuzzy rules are applied for computing the fuzzy subsets $D_{min} = (\min\{d_1^1, d_1^2\} \dots \min\{d_n^1, d_n^2\})$ and $D_{max} = (\max\{d_1^1, d_1^2\} \dots \max\{d_n^1, d_n^2\})$, finally, the child shall be defined at random by using a uniform probability distribution applied to the range of all fuzzy min and max values. The mutation of a fuzzy decision gene is carried out by choosing a random value from the interval $[0,1]$. In Voigt, Born & Santibáñez-Koref (1993), other types of genetic operators were considered as well, some of them were adapted from the Breeder GA (Mühlenbein & Schlierkamp-Voosen, 1993).

Different experiments carried out in Voigt, Born & Santibáñez-Koref (1993) with $m = 1$ and $m = 2$ showed that the use of fuzzy representation allows a robust behavior to be obtained. In some cases, a better performance than the Breeder GA was achieved. Furthermore, another important conclusion was stated: for a small population size the performance for $m = 2$ is slightly better than for $m = 1$, whereas the opposite is true for large population sizes.

### 3.6 Fuzzy Stop Criterium

Due to the possibility of premature convergence, GAs do not guarantee that the optimal solution shall be found. Therefore, if the optimal solution is not known, GA performance is difficult to measure accurately. In Meyer & Feng (1994), a fuzzy stop criterium mechanism (FSCM) is developed to provide a useful evaluation of the GA real time performance. FSCM is based on achieving a user-defined level of performance for the given problem. In order to do so, it includes a predicting process based on statistics for estimating the value of the GA optimal solution, then it compares the current solution to this optimal one by checking if an acceptable percentage (specified by the user) of the latter is reached. If so, the GA stops and returns belief and uncertainty measures that provide a reliability measure for the GA chosen solution.

The acceptable percentage optimal solution defined by the user represents a fuzzy stop criterium for stopping GA if an appropriate solution is reached. The predicting process is invoked every 40 iterations and uses performance values such as the minimum solution value, average solution value and belief and plausibility values, all obtained during these iterations.

The underlying idea for the FSCM is that the user does not need to find the global solution, but rather an approximate solution that is close to the optimal one, i.e., the GA is used for solving a *fuzzy goal* instead of a crisp one because of the vagueness of the term approximate. This term is quantitatively measured by the user through the acceptable percentage of the optimal solution that he requires in the final solution.

Results obtained on a 25-city TSP problem indicate this approach is preferable to a simple GA, in term of cost/performance and in decreasing the amount of time the GA searches for acceptable solutions.

## 4 FLCs-Based Adaptive GAs

The GA control parameter settings such as mutation probability ($p_m$), crossover probability ($p_c$)

and population size ($N$) are key factors in the determination of the exploitation versus exploration tradeoff. It has long been acknowledged that they have a significant impact on GA performance (Grefenstette, 1986). If poor settings are used, the exploration/exploitation balance may not be reached in a profitable way; the GA performance shall be severely affected due to the possibility of premature convergence. However, finding robust control parameter settings that allow the premature convergence to be avoided in any problem is not a trivial task, since their interaction with GA performance is complex and the optimal ones are problem-dependent (Bäck, 1992). Furthermore, different control parameter values may be necessary during the course of a run for inducing an optimal exploration/exploitation balance. For these reasons, *Adaptive* GAs (AGAs) have been built that dynamically adjust selected control parameters or genetic operators during the course of evolving a problem solution in order to offer the most appropriate exploration and exploitation behavior (see Angeline (1995), Herrera & Lozano (1996a), Hinterding, Michalewicz & Eiben (1997) and Smith & Fogarty (1997)).

A promising way followed for building AGAs involves the application of FLCs (see Appendix A) for adjusting GA control parameters. FLCs implement an expert operator's approximate reasoning process in the selection of a control action. An FLC allows one to qualitatively express the control strategies based on experience as well as intuition. These control strategies may be expressed in a form that permits both computers and humans to share them efficiently. FLCs are useful for adapting GA control parameters, following control strategies underlying human expertise and knowledge on GAs.

The goal of this section is to report an extensive study of the FLCs-based AGAs. We begin, in the next Subsection, by explaining their main idea and summarizing the steps in their design.

We should point out that the FLCs have been considered as well for producing adaptation on other Evolutionary Algorithms. As an example, in Reynolds & Chung (1997), an FLC determines the number of individuals which will impact the current beliefs used by a Cultural Algorithm.

## 4.1 Design of FLCs-Based AGAs

FLCs-based AGAs are to be found in Arnone, Dell'Orto & Tettamanzi (1994), Bergmann, Burgard & Hemker (1994), Herrera, Herrera-Viedma, Lozano & Verdegay (1994), Herrera & Lozano (1996a), Lee & Takagi, (1993), Tettamanzi (1995), Voget (1996), Xu, Vukovich, Ichikawa & Ishii (1994), Wang, Wang & Hu (1997) and Zeng & Rabenasolo (1997). Their main idea is to use an FLC whose inputs are any combination of GA performance measures or current control parameters and whose outputs are GA control parameters. Current performance measures of the GA are sent to the FLC, which computes new control parameter values that will be used by the GA. Figure 6 shows this process.

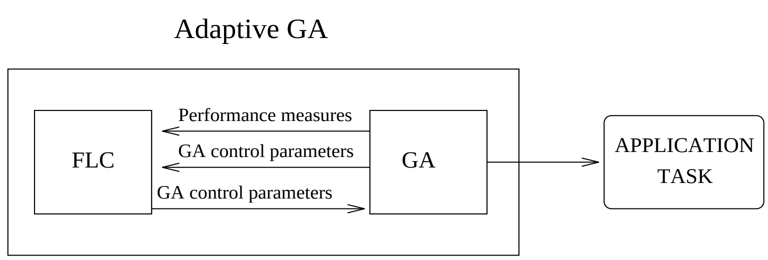


**Figure 6.** FLCs-based AGA model

In general, the following steps are needed in order to design an FLCs-based AGA (Herrera & Lozano, 1996a):

**Defining the inputs and outputs.** Inputs should be robust measures that describe GA behavior and the effects of the genetic setting parameters and genetic operators. In Tettamanzi (1995), some possible inputs were cited: diversity measures, maximum, average and minimum fitness, etc. In Lee & Takagi (1993), Xu & Vukovich (1993) and Xu, Vukovich, Ichikawa, Ishii (1994), it is suggested that current control parameters may also be considered as inputs. Outputs indicate values of control parameters or changes in these parameters (Lee & Takagi, 1993). In Tettamanzi (1995), the following outputs were reported: mutation probability, crossover probability, population size, selective pressure, the time the controller must spend in a target state in order to be considered successful, the degree to which a satisfactory solution has been obtained, etc.

**Defining the data base.** Each input and output should have an associated set of linguistic labels. The meaning of these labels is specified through membership functions of fuzzy sets. So, it is necessary that every input and output have a bounded range of values in order to define these membership functions over it.

**Obtaining the fuzzy rule base.** After selecting the inputs and outputs and defining the data base, the fuzzy rules describing the relations between them should be defined. Although, the experience and the knowledge of GA experts may be used for deriving fuzzy rule bases, many authors have found difficulties to do this. For example, in Lee & Takagi (1993), the following was stated:

> *"Although much literature on the subject of GA control has appeared, our initial attempts at using this information to manually construct a fuzzy system for genetic control were unfruitful."*

In Tettamanzi (1995), a related reflection is quoted:

> *"Statistics and parameters are in part universal to any evolutionary algorithm and in part specific to a particular application. Therefore it is hard to state general fuzzy rules to control the evolutionary process."*

Finally, in Herrera & Lozano (1996a), a similar conclusion is suggested as well:

> *"The behavior of GAs and the interrelations between the genetic operators are very complex. Although there are many possible inputs and outputs for the FLCs, frequently fuzzy rule bases are not easily available: finding good fuzzy rule bases is not an easy task."*

Automatic learning mechanisms for obtaining fuzzy rule bases have been introduced for avoiding this problem. By using these mechanisms, relevant relations and membership functions may be automatically determined and may offer insight for understanding the complex interaction between GA control parameters and GA performance (Lee & Takagi, 1993). Two different fuzzy rule base learning techniques have been presented: the *a priori learning technique* (Lee & Takagi, 1993; Lee & Takagi, 1994) and *Coevolution with Fuzzy Behaviors* (Herrera & Lozano, 1998).

- The a priori learning mechanism is based on the same idea as the meta-GA of Grefenstette (1986). It works considering a set of test functions that have nothing to do with the particular problem to be solved, which may limit the robustness of the fuzzy rule bases returned. The mechanism is executed once, however it has associated a high computational cost.

- Coevolution with Fuzzy Behaviors uses an additional GA that coevolves with the FLCs-based AGA. Fuzzy rule bases are learnt taking into account only the particular problem being solved, which may allow the fuzzy rule bases obtained to be adapted enough for it.

Next, we review different approaches of FLCs-based AGAs, categorized according to the way in which the fuzzy rule bases that they use are obtained. First, in Subsection 4.2, we explain the ones using fuzzy rule bases based on the knowledge and the experience of the GA experts, then, in Subsection 4.3, the one associated with the a priori learning mechanism, and finally, in Subsection 4.4, the one including Coevolution with Fuzzy Behaviors.

## 4.2 Using Fuzzy Rule Bases Derived from the Knowledge of GA Experts

Next, we survey different FLCs-based AGA models that use fuzzy rule bases built through the expertise, experience and knowledge on GAs, which have become available as a result of empirical studies conducted over a number of years.

### 4.2.1 Approach of Xu & Vukovich (1993) and Xu, Vukovich, Ichikawa & Ishii (1994)

In Xu & Vukovich (1993) and Xu, Vukovich, Ichikawa, Ishii (1994), the use of FLCs to control GAs is considered for solving two problems to which a standard GA may be subjected: very slow search speed and premature convergence. For the authors these problems are due to: 1) control parameters not well chosen initially for a given task, 2) parameters always being fixed even though the environment in which the GA operates may be variable and 3) difficulties resulting from selection of other parameters such as population size and in understanding their influence, both individually and in combination, on the GA performance. They proposed to use FLCs for controlling GAs in order to: 1) choose control parameters before GA run, 2) adjust the control parameters on-line to dynamically adapt to new situations and 3) assist the user in accessing, designing, implementing and validating a GA for a given task.

Experiments were carried out with an FLCs-based AGA that controls $p_c$ and $p_m$ using two FLCs. Both of them had the same inputs: generation and population size. The fuzzy rule bases considered are shown in Table 3. The authors claimed that part of the mechanism for creating fuzzy rules for adapting $p_m$ is that this one should increasingly diminish when the GA approaches convergence to the best fitness.

| | **Population Size** | | |
|---|---|---|---|
| **Generation** | *Small* | *Medium* | *Big* |
| *Short* | Medium | Small | Small |
| *Medium* | Large | Large | Medium |
| *Long* | Very Large | Very Large | Large |

| | **Population Size** | | |
|---|---|---|---|
| **Generation** | *Small* | *Medium* | *Big* |
| *Short* | Large | Medium | Small |
| *Medium* | Medium | Small | Very Small |
| *Long* | Small | Very Small | Very Small |

**Table 3.** Fuzzy rule bases for the control of $p_c$ and $p_m$, respectively

The FLCs-based AGA stood out as the most efficient algorithm against a standard GA in solving the TSP and other optimization problems.

### 4.2.2 ARGAF

In Herrera & Lozano (1996a), an FLCs-based adaptive RCGA, called *ARGAF*, was proposed. Its principal features are described below.

**Crossover Operator Application.** ARGAF applies two different crossover operators; one with exploitation properties and another with exploration properties. A parameter, denoted as $p_e$, defines the frequency of application of the exploitative operator. Its value strongly influences the exploration/exploitation balance induced by crossover; if $p_e$ is low, ARGAF shall generate diversity and so exploration takes effect, whereas if it is high, the current diversity shall be used for generating best elements and so exploitation comes into force.

Different crossover operator types were considered for building versions of ARGAF. For example, the FCB-crossover operators (Subsection 3.1) were used as follows:

**For** each pair of chromosomes from a total of $p_c \cdot N$ **Do**
    **If** a random number $r \in [0, 1]$ is lower than $p_e$ **Then**
        Generate two offspring, the result of applying two $M$-crossovers.
    **Else**
        Generate two offspring, the result of applying an $F$-crossover and an $S$-crossover.
    The two offspring substitute their parents in the population.

**Selection Mechanism.** ARGAF uses the *linear ranking* selection mechanism (Baker, 1985). In this selection mechanism the selective pressure is determined by the parameter $\eta_{min} \in [0, 1]$. If $\eta_{min}$ is low, high pressure is achieved, whereas if it is high, the pressure is low.

**Adaptation of $p_e$ y $\eta_{min}$.** The $p_e$ and $\eta_{min}$ parameters are adjusted using two FLCs. The inputs of the FLCs are a genotypical diversity measure and a phenotypical diversity measure. The first measure determines the *quantity* of diversity in the population and the second the *quality* of this diversity.

Adapting the $p_e$ parameter, ARGAF controls the effects of crossovers, i.e., either generating diversity or using diversity, whereas adjusting the $\eta_{min}$ parameter, it controls the effects of selection, i.e., either keeping diversity or eliminating diversity. The joint management of these parameters allows

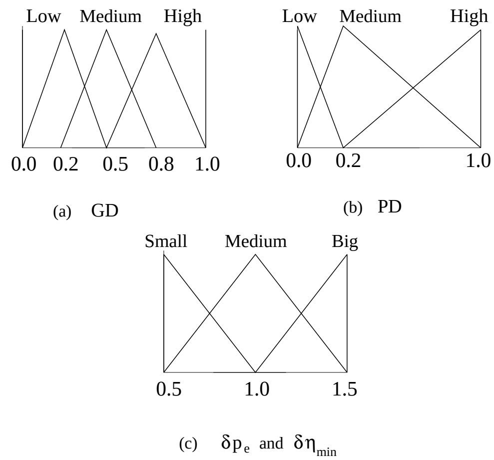


**Figure 7.** Meaning of the linguistic terms associated with the inputs and the outputs

ARGAF to administer the diversity in a suitable way. For example, if useful diversity is detected by ARGAF, then it sets selection for keeping diversity and crossover for using it. If the level of diversity is high and its quality is not good, then ARGAF increases the selective pressure and tries to obtain better elements by increasing exploitation by means of crossover. All these considerations are included in the fuzzy rule bases of the FLCs used by ARGAF.

Next, we discuss the different steps in the design of the FLCs used by ARGAF.

1. *Inputs*

   Two diversity measures were considered as inputs. One is a genotypical diversity measure based on the Euclidean distances of the chromosomes in the population from the best one. Its definition is $GD = (\bar{d} - d_{min})/(d_{max} - d_{min})$, where $\bar{d}$, $d_{max}$ and $d_{min}$ are the average, maximum and minimum distances of the chromosomes in the population from the best one, respectively. The range of $GD$ is $[0, 1]$. If $GD$ is low, most chromosomes in the population are concentrated around the best chromosome and so convergence is achieved. If $GD$ is high most chromosomes are not biased towards the current best element.

   The other input is a phenotypical diversity measure (Lee & Takagi, 1993) defined as: $PD = \frac{f_{best}}{\bar{f}}$, where $\bar{f}$ and $f_{best}$ are the average and best fitness, respectively. $PD$ belongs to the interval $[0, 1]$. If it is near to 1, convergence has been reached, whereas if they are near to 0, the population shows a high level of diversity.

   The linguistic label set of these inputs is $\{Low, Medium, High\}$.

2. *Outputs*

   The outputs are variables that control the variation on the current $p_e$ and $\eta_{min}$ parameters, which are kept within the range $[0.25, 0.75]$. These variables, noted as $\delta p_e$ and $\delta\eta_{min}$, represent the degree to which the current $p_e$ and $\eta_{min}$ should vary, respectively. The variations shall be carried out by multiplying the $\delta p_e$ and $\delta\eta_{min}$ values, obtained by the FLCs, by the current $p_e$ and $\eta_{min}$ values, respectively. The action interval of $\delta p_e$ and $\delta\eta_{min}$ is $[0.5, 1.5]$ and their associated linguistic labels are $\{Small, Medium, Big\}$.

3. *Data Base*

   The data base is shown in Figure 7. The meaning of the linguistic terms associated with $GD$ is depicted in (a), the ones for $PD$ in (b) and finally, the ones for $\delta p_e$ and $\delta\eta_{min}$ in (c). For each linguistic term, there is a *triangular fuzzy set* that defines its semantic, i.e, its meaning.

4. *Fuzzy Rule Base*

   The fuzzy rules describe the relation between the inputs and outputs. Table 4 shows the fuzzy rule bases used by the FLCs of ARGAF.

| | PD | | |
|---|---|---|---|
| **GD** | *Low* | *Medium* | *High* |
| *Low* | Medium | Small | Small |
| *Medium* | Big | Big | Medium |
| *High* | Big | Big | Medium |

| | PD | | |
|---|---|---|---|
| **GD** | *Small* | *Medium* | *Large* |
| *Low* | Small | Medium | Big |
| *Medium* | Small | Big | Big |
| *High* | Small | Small | Big |

**Table 4.** Fuzzy rule bases for the control of $p_e$ and $\eta_{min}$, respectively

Experiments on different optimization problems of parameters with variables on continuous domains were carried out in order to study the efficiency of ARGAF. Its results were compared with the ones of some algorithms like ARGAF, but with fixed $p_e$ and $\eta_{min}$ values. Different combinations of these parameters were considered. The main conclusion was that ARGAF is a very robust GA since it adapts the $p_e$ and $\eta_{min}$ parameters to the settings that return the best results (which were different from one function to others).

#### 4.2.3 Fuzzy Government

In Arnone, Dell'Orto & Tettamanzi (1994), it is claimed that Evolutionary Algorithms require human supervision during their routine use as practical tools for the following reasons: 1) for detecting the emergence of a solution, 2) for tuning algorithm parameters and 3) for monitoring the evolution process in order to avoid undesiderable behavior such as premature convergence. It is advised as well that any attempt to develop artificial intelligence tools based on Evolutionary Algorithms should take these issues into account. They proposed to use FLCs for this task. They called *Fuzzy Government* the collection of fuzzy rules and routines in charge of controlling the evolution of the GA population.

Fuzzy Government was applied to the symbolic inference of formulae problem. Genetic programming (Koza, 1992) was used to solve the problem along with different FLCs, which dynamically adjusted the maximum length for genotypes, acted on the mutation probability, detected the emergence of a solution and stopped the process. The results showed that the performance of the fuzzy governed GA was almost impossible to distinguish from the performance of the same algorithm operated directly with human supervision.

#### 4.2.4 FLCs with Variance Based Genotypical Diversity Measures

In Herrera, Herrera-Viedma, Lozano & Verdegay (1994), the use of FLCs was propose for monitoring the population diversity and the exploration/exploitation balance of an RCGA. It was proposed to build two FLCs that control $p_c$ and $p_m$, depending on the following two genotypical diversity measures:

- *The variance average chromosomes* VAC $= \frac{\sum_{i=1}^{N}(\bar{S}_i - \bar{S})^2}{N}$.
- *The average variances alleles* AVA $= \frac{\sum_{j=1}^{N}\sum_{i=1}^{n}(S_{ij} - \bar{S}_j)^2}{n \cdot N}$.

where $S_i$, i = 1, ..., N, denotes the chromosome $i$, $S_j$, j = 1, ..., n, denotes the vector of genes with position $j$ in the population, $S_{ij}$, denotes the gene with position $j$ in the chromosome $i$, and

$$\bar{S}_i = \frac{\sum_{j=1}^{n} S_{ij}}{n}\ ,\ \bar{S} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{N} S_{ij}}{n \cdot N} \text{ and } \bar{S}_j = \frac{\sum_{i=1}^{N} S_{ij}}{N}.$$

The range for VAC and AVA was established as follows: VAC$_{min}$ = 0 and VAC$_{max}$ as the maximum value obtained at the first generation, for some value higher than it, the maximum value is automatically interchanged; and similarly for AVA, with AVA$_{min}$ = 0. The properties of these diversity measures are the following: they are indifferent to mutual exchange of two chromosomes in a population; and when all the chromosomes in a population are almost identical, the VAC and AVA measures take low values. The fuzzy fuzzy rules suggested were the following:

**If** VAC is low **then** $p_c$ should be adjusted upwards slightly.
**If** VAC is high **then** $p_c$ should be forced downwards slightly.

**If** AVA is low **then** $p_m$ should be adjusted upwards slightly.
**If** AVA is high **then** $p_m$ should be forced downwards slightly.

### 4.2.5 FLCs-Based AGAs for Multiobjective Optimization Problems

In Voget (1996), an FLCs-based AGA for multiobjective optimization problems is presented. In each generation an FLC decides what transformation of the cost components into an one-dimensional fitness function is taken.

In Voget & Kolonko (1998), a more complex method, called *Fuzzy Reduction Genetic Algorithm*, is proposed. It attempts to enable a uniform approximation of the *Pareto Optimal* solutions (those that cannot be improved with respect to any cost function without making worse the value of some other). The direction of the selection pressure is adapted to the actual state of the population, forcing it to explore a broad range of so far Pareto Optimal solutions. The adaptation is done by an FLC that controls the selection procedure and the fitness function.

The authors started by explicitly formulating desirable goals for the evolution of the population towards the target Pareto Optimal solutions. These goals could be expressed in vague terms only, and so, the use of an FLC seemed to be appropriate for dealing with them. Then, they defined deviation measures of a population from these goals, which were the inputs to the FLC. Latter, they fix a set of possible actions which could serve as countermeasures to decrease the deviations. These actions are different selection mechanisms based on classical ones presented in the literature for tackling multiobjective optimization problems, which have several disadvantages if used constantly. The FLC determines activation rates for the actions. The action that should actually be taken is decided according to the activation rates found, following different strategies, such as to choose the one with maximal activation rate, or to select an action randomly according to the distribution of the activation rates, etc.

As an application, a timetable optimization problem is presented where the method was used to derive cost-benefit curves for the investment into railway nets. The results showed that the fuzzy adaptive approach avoids most of the empirical shortcomings of other multiobjective GAs by the adaptive nature of the procedure.

### 4.2.6 FLCs-Based Adaptation at Individual-Level

In general, there are three levels where the adaptation may take place in an AGA (Angeline, 1995; Smith & Fogarty, 1997): 1) *population-level* adaptations adjust control parameters that are overall for the entire population, 2) *individual-level* adaptations are centred on the consideration of the individual members of the population rather than the ensemble as a whole, and 3) *component-level* adaptations dynamically alter how the individual components of each chromosome will be manipulated independently from each other. The FLCs-based AGAs presented in the previous subsections involve population-level adaptation.

FLCs-based adaptive mechanisms at individual-level may be interesting for adjusting control parameters associated with genetic operators (Herrera & Lozano, 1996a; Herrera & Lozano, 1998; Zeng & Rabenasolo, 1997). In this way, the control parameters will be defined on samples instead of on the whole population, i.e., in a *distributed way*. Inputs to the FLCs may be central measures and/or measures associated with particular chromosomes or sets of them, and outputs may be control parameters associated with genetic operators that are applied to those chromosomes. A justification for this approach is that it allows for the application of different search strategies in different parts of the search space. This is based on the reasonable assumption that in general search space will not be homogeneous, and that different strategies will be better suited to different kinds of sub-landscapes (Smith & Fogarty, 1997). For instance, a population member currently residing in a relatively flat region of the search space may be handled more severely than a population member in a more complex portion of the search space (Angeline, 1995).

In Zeng & Rabenasolo (1997), an FLCs-based binary-coded AGA is presented in which the crossover probability, the probability of crossover position and the mutation probability are defined on specific samples of the population using several FLCs that take into account fitness values of samples and distances between samples. In particular, the following measures were considered as inputs to the FLCs, where $X$ and $Y$ are two chromosomes to be crossed (maximization is assumed).

- *Variance of fitness values*: $Var = \frac{f_{max}-\bar{f}}{f_{max}-f_{min}}$, where $\bar{f}$ is the average of all fitness values and $f_{max}$, $f_{min}$ the maximal and the minimal fitness values, respectively.
- *Distance between the fitness value of the best parent and $f_{max}$*: $G = \frac{f_{max}-\max\{f(X),f(Y)\}}{f_{max}-f_{min}}$.

- *Distance between X and Y*: $D = d(X, Y)$.
- *Normalized fitness values*: $f_1 = \frac{f(X)}{f_{max}}$ and $f_2 = \frac{f(Y)}{f_{max}}$.

$Var$ is overall for the entire population, and $G$, $f_1$, $f_2$ and $D$ are measures defined on specific samples. All the measures were included in $[0, 1]$. Their set of linguistic labels is $L = \{Small, Big\}$.

Next, we show, as an example, the fuzzy rule base considered for obtaining the crossover probability, $p_c$, for each pair of parents, $(X, Y)$. The set of linguistic labels for $p_c$ is $L = \{Small, Big\}$.

**If** $G$ is Big **then** $p_c$ is Big.
**If** $Var$ is Small and $G$ is Small **then** $p_c$ is Small.
**If** $D$ is Small and $f_1$ is Big and $f_2$ is Big **then** $p_c$ is Big.
**If** $D$ is Small and ($f_1$ or $f_2$) is Small **then** $p_c$ is Small.
**If** $D$ is Big **then** $p_c$ is Big.

These fuzzy rules attempt to implement the following principles (where apparently there are some conflicts):

- *Maintain the diversity in the population;* two distant samples have more chance to be selected for crossover.
- *Enhance the searching in optimal regions;* two near samples with high fitness values have more chance to be selected for crossover.
- *Avoid convergence to local optima;* crossover operations have to be enhanced if the variance of fitness values is very small.
- *Stabilize optimal populations;* crossover operations have to be reduced if the specific fitness values of the samples to be selected are close enough to the maximal fitness value of the current population.

The FLC proposed for adapting the mutation probability was designed so that mutation operations can be enhanced when the GA tends to a local optimum and that they can reduce when the current population is in strong variations or the globally optimal population is obtained.

The results of the FLCs-based AGA proposed on a multimodal test function were compared with the ones of a simple GA. On the one hand, populations in the FLCs-based AGA were more diversified than those in the simple GA, on the other, it was easier for the FLCs-based AGA to lead to the global maximum and its convergence behavior was better than the simple GA.

Another approach involving FLCs-based adaptation at individual-level is the one in Subsection 4.4.

## 4.3 A Priori Learning of Fuzzy Rule Bases

In Lee & Takagi (1993) and Lee & Takagi (1994), an automatic learning process was proposed for obtaining suitable fuzzy rule bases along with their the data bases. The mechanism is very similar to the meta-GA of Grefenstette (1986). It is based on a GA whose chromosomes code possible fuzzy rule bases together with their corresponding data bases. The fitness function value for a chromosome is calculated using the Online and Offline measures (De Jong, 1975) obtained from an FLCs-based AGA that uses the fuzzy rule base coded in such chromosome on each one of the five De Jong's test functions. After the meta-level GA completed 1000 fitness function evaluations, the best fuzzy rule base reached is returned.

In order to study this mechanism, an FLCs-based AGA was developed using three FLCs. All consider the following three inputs:

- Two phenotypical diversity measures: $PD_1 = \frac{f_{best}}{\bar{f}}$ and $PD_2 = \frac{\bar{f}}{f_{worst}}$, where $\bar{f}$, $f_{best}$ and $f_{worst}$ are the average, best and the worst fitness, respectively. $PD_1$ and $PD_2$ belong to the interval $[0, 1]$. If they are near to 1, convergence has been reached, whereas if they are near to 0, the population shows a high level of diversity.

- The change in the best fitness since the last control action.

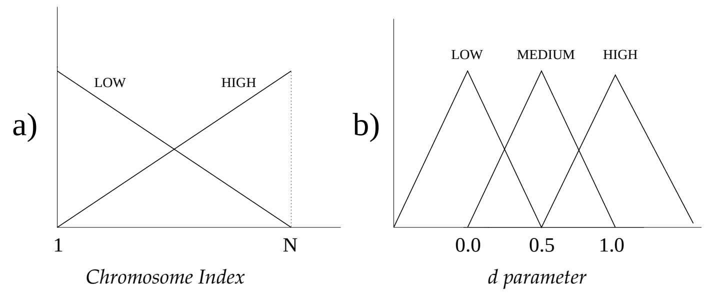


**Figure 8.** Meanings for the linguistic labels considered

The outputs of the three FLCs are variables that control variations in the current $p_m$, $p_c$ and $N$, respectively. For example, such output for the control of $N$, represents the degree to which the current $N$ should vary. The new population size is computed by multiplying the output value by the current $N$.

The automatic learning mechanism was executed for obtaining both the data base and the fuzzy rule base for this example. An FLCs-based AGA using the resultant FLCs was applied to a particular problem: the inverted pendulum control task. The results obtained exhibited better performance than a simple static GA. Other experiments aimed at isolating the effects of the FLCs-based adaptation of $N$, $p_c$ and $p_m$ showed that the adaptation of the mutation probability contributes most to high performance (Lee & Takagi, 1994).

## 4.4 Coevolution with Fuzzy Behaviors

In Herrera & Lozano (1998), a fuzzy rule learning process is presented for the case of the distributed application of FLCs for adapting genetic operator control parameters (see Subsection 4.2.6). It is called *Coevolution with Fuzzy Behaviors*. A *Fuzzy Behavior* (FB) is a vector with the linguistic values of the fuzzy rule consequent of an FLC, that encodes its fuzzy rule base. The goal of this mechanism is to obtain fuzzy rule bases producing suitable control parameter values for allowing the genetic operator to show an adequate performance.

The basic idea is to introduce a population of FBs that *coevolves* with the population of chromosomes, which represent solutions to the particular problem. During the application of the genetic operator to be controlled, a random assignment is established between each FB and sets of parents. Then, the genetic operator is applied to each set using the control parameter value obtained from an FLC that uses the fuzzy rule represented by the corresponding FB.

The population of FBs will undergo evolution, through the effects of its own selection process and, crossover and mutation operators. The fitness of the FBs will depend on the efficacy induced by them on the genetic operator. Some aspects to be considered may be: whether they generate offspring fitter that the parents, or introduce high diversity levels, etc. In this way, the mechanism is based on observing the relative performance of different strategies (represented by the FBs) on the particular problem to be solved, which appear to be very effective, as is claimed in Smith & Fogarty (1997).

An instance was implemented for the case of the distributed adaptation of the $d$ control parameter associated with Soft Modal crossover (Subsection 3.3.1). In particular, the range of $d$ is constrained to the interval $[0, 1]$. The features considered for each pair of parents, $X$ and $Y$, were their index in the population, $Index(X)$, $Index(Y) \in \{1, ..., N\}$ ($N$ is the population size). The index of the best chromosome is $N$, and the one of the worst chromosome is 1 (the fitter elements have larger indexes). The set of linguistic labels associated with $Index(X)$ and $Index(Y)$ is $L = \{Low, High\}$. The meanings of these labels are depicted in Figure 8.a. The set of linguistic labels for $d$ is $L_d = \{Low, Medium, High\}$. Their meanings are shown in Figure 8.b.

Each FB codes a fuzzy rule base having fuzzy rules whose inputs are $Index(X)$ and $Index(Y)$ and whose output is $d$. Figure 9 shows a pseudocode algorithm integrating the main GA and the additional GA for evolving FBs. $P(t)$ denotes the population of chromosomes at generation $t$ and $P_{FBs}(t)$ is the population of FBs at generation $t$.

Steps 2, 4, 5.2, 5.5 and 5.6 constitute the main GA. Steps 3, 5.7, 5.8 and 5.9 form the additional GA for evolving FBs. Steps 5.3, 5.4 and 5.7, the cooperation between them is carried out. In the case

**Genetic Algorithm**

**1.** $t = 0$;

**2.** *Initialize* $P(t)$;

**3.** *Initialize* $P_{FBs}(t)$;

**4.** *Evaluate* $P(t)$;

**5.** *While* (*not* termination-condition) *do*

**5.1.** $t = t + 1$;

**5.2.** *Select* $P(t)$ *from* $P(t-1)$;

**5.3.** *Assign to each pair of parents in* $P(t)$ *an FB in* $P_{FBs}(t-1)$, *at random*;

**5.4.** *Calculate* $d$ *for each pair of parents through their FB*;

**5.5.** *Perform Soft Modal Crossover (using the* $d$ *values) and Mutation on* $P(t)$;

**5.6.** *Evaluate* $P(t)$;

**5.7.** *Evaluate* $P_{FBs}(t-1)$;

**5.8.** *Select* $P_{FBs}(t)$ *from* $P_{FBs}(t-1)$;

**5.9.** *Perform Crossover and Mutation on* $P_{FBs}(t)$;

**Figure 9.** GA with an Adaptive Soft Modal crossover operator by coevolution with FBs

of step 5.7, the cooperation lies in the use of the information about the chromosomes generated for evaluating the FBs.

The fitness function associated with the FBs should take into account the performance of Soft Modal crossover when it is applied to the parents with the $d$ value obtained from them. The performance was judged according to the following criterion:

> *At first, it seems reasonable to consider that the ability of the operator to produce offspring of increased fitness may be what is required (Tuson & Ross, 1996). After all, the aim of a GA is to maximize the progress of the average fitness throughout the search. However, this is not enough. An efficient crossover operator should introduce the right portion of variance into the offspring population. If the variance is too large then the GA does not converge at all, whereas if it is too small then it converges prematurely (Voigt, Mühlenbein & Cvetković, 1995).*

Taking into account this biobjective, the following fitness function for each $FB_i$ in $P_{FBs}(t)$ was proposed (minimization is assumed):

$$Fit(FB_i) = \begin{cases} 0 & \text{if } \bar{f}_O < f(X) \leq f(Y), \\ 2 - d_i & \text{if } f(X) \leq \bar{f}_O < f(Y), \\ 3 & \text{if } f(X) \leq f(Y) \leq \bar{f}_O, \end{cases}$$

where $\bar{f}_O$ is the average of the fitness of the two offspring generated, $f(X)$ and $f(Y)$ are the fitness function values of the parents ($f(X) \leq f(Y)$ is assumed), and $d_i$ is the $d$ value calculated from $FB_i$ and $Index(X)$ and $Index(Y)$.

This function induces the aforementioned biobjective for reaching crossover operator performance in the following ways:

1. $Fit(\cdot)$ rewards FBs that produce offspring fitter than the parents.

2. $Fit(\cdot)$ penalizes FBs that produce offspring worse than the parents.

3. When the fitness of the offspring is between the ones for the parents, FBs introducing more diversity (those using greater $d$ values) are preferred. Anyway, the fitness of these FBs will be better than the ones assigned to the FBs in case 2, and worse than the ones in case 1.

An empirical study of the Adaptive Soft Modal crossover was made using test functions with different difficulties from two points of view, one of performance and one of learning behavior (based on the distributions of FBs appearing during the runs). Its results were compared with the ones from Soft Modal crossover with fixed $d$ values ($d = 0$, $d = 0.5$ and $d = 1$) and with the ones from adapting $d$ in a distributed way, through an FLC with a fixed fuzzy rule base (different ones were tried). The main conclusion was that the learning ability associated with the Adaptive Soft Modal crossover allowed suitable distributions of FBs to be produced for introducing a *robust* operation. This means that a significant performance was obtained for each one of the test functions, which achieved their best results through fuzzy rule bases or fixed $d$ values that might be different (as was observed in the experiments).

# 5 Future Work on FGA Research

In this section, we report on some challenges for future work on the field of FGAs. In Subsection 5.1, we introduce newcomers in the research on GA components based on fuzzy tools, and in Subsection 5.2, we set out possible variations and extensions of FLCs-based AGAs.

## 5.1 Future Research on GA Components Based on Fuzzy Tools

Next, we describe some future research areas concerning GA components based on fuzzy tools.

**Fuzzy selection mechanisms.** Selection mechanisms may be built allowing to select chromosomes by their fitness and by other properties that help to guide the GA towards promising regions of the search space. Some aggregation operators based on fuzzy connectives may be used by this mechanism. This model may be extended for tackling properties whose importance changes through the GA run.

**Crossover operators.** Crossover operators for RCGAs may be designed using triangular probability distributions or fuzzy connectives integrating ideas of recent crossover operators, such as BLX-$\alpha$-$\beta$-$\gamma$ (Eshelman, Mathias & Schaffer, 1997) and *Unimodal Normal Distribution Crossover* (Ono & Kobayashi, 1997), which have shown promising results.

**Fuzzy decision processes for the application of the genetic operators.** For example, it would be interesting to constrain when the crossover is carried out; the probability of crossing two low fitness parent chromosomes should be low, whereas this should not occur when any parent show good fitness. This approach would be appropriate for the application of *Mate Selection* mechanisms (Ronald, 1993), in which chromosomes effect the choice of mate for crossover.

## 5.2 Challenges for the FLCs-Based AGA Research

Despite the recent activity and the associated progress in FLCs-based AGA research, there remain many directions in which they may be improved or extended. Next, we outline some of them.

**Relevant inputs.** Research on determining relevant input variables for the FLCs controlling GA behavior should be explored (Lee & Takagi, 1994). These variables should describe either states of the population or features of the chromosomes so that control parameters may be adapted on the basis thereof for introducing performance improvements. Coevolution with FBs may be useful for discovering this type of variables (for the case of distributed adaptation). Different FBs, coding fuzzy rules with a distinct number and type of inputs, may evolve together in the same population. The learning process associated with this approach will proportionate the most significant inputs along with the fuzzy rules concerning them.

The nature of the possible input variables should be study as well, i.e., whether they are universal to any GA or particular to a given problem. This would be useful for determining the fuzzy rule base learning procedure that may be applied: the a priori learning mechanism (Subsection 4.3), when fuzzy rules involve universal inputs, and Coevolution with FBs (Subsection 4.4), when they involve inputs that are particular to the problem.

**Feedback between genetic operators.** It may be interesting to design FLCs taking into account the action of each genetic operator in relation to the behavior of each one of the remaining ones. The future action of an operator may be tuned depending on the repercussions of the actions of other operators (even itself). In this way, a feedback between operators must be established, allowing a suitable balance between their actions to be reached throughout the GA run.

ARGAF (Subsection 4.2.2) may be considered as a first approximation to do this, since it complements the role of the selection mechanism (either maintaining or eliminating diversity) with the role of the crossover operator (either creating or using diversity). Its significant performance shows promise in this future research front.

**FLC application frequency.** Usually, a fixed scheduling for firing the FLCs has been followed, i.e. after each fixed number of generations. However, the choice of the time-interval between controls is a parameter that has an influence on the final controller performance. If the controller is fired with a low time-interval the effects of previous controls may not be achieved, whereas if the controller is fired with a high time-interval, the search process may be misled by particular parameter values. A possible solution is to fire the controller when certain conditions relating to some performance measures are reached.

**Extensions for the FLCs-based AGA approach by Coevolution with FBs (Subsection 4.4).** This technique may be considered for adapting other genetic operators whose operation is determined by particular control parameters, such as the Dynamic FCB-crossover operators (Subsection 3.2), BLX-$\alpha$-$\beta$-$\gamma$ operator (Eshelman, Mathias & Schaffer, 1997) and *Multi-Parent* recombination operators (Eiben, 1997).

It may be used as well for problems where we intuit that particular features of the parents may be taken into account for allowing the crossover operator behavior to be effective, but we do not know the precise fuzzy rules determining the relation between these features and the appropriate control actions for the operator. In this fashion, this approach allows particular knowledge about the problem to be integrated in the GA in order to improve its behavior. Hence, it represents a possible way for *hybridization* (Davis, 1991).

Another possible extension involves its use for adaptations at component-level, which associate adaptive parameters with each component of an evolving individual that determine how each component is modified during reproduction (see Subsection 4.2.6).

**FLC-based adaptive representations and adaptive fitness functions.** FLCs may determine the way in which the chromosomes should be represented. For example, an FLC may controls the length of binary-coded chromosomes in order to obtain a suitable accuracy when chromosomes represent numerical variables. On the other hand, control parameters associated with some *fitness scaling* techniques (Goldberg, 1989; Michalewicz, 1992) may be controlled through FLCs as well.

**Application of FLCs-based AGAs for constrained parameter optimization problems.** Different ways may be considered for carrying out this task: 1) modify the FLCs-based AGA model presented in Voget & Kolonko (1998), which deals with multiobjective optimization problems, for solving constrained problems (the problem of satisfying a number of violated inequality constraints is clearly the multiobjective problem of optimizing the associated functions until given values are reached), 2) build methods of adaptive penalties based on FLCs with measures that describe the difficulty of the constrained problem (see Michalewicz & Schoenauer (1996)), 3) design adaptive genetic operators by coevolution with FBs for dealing with these problems, where the FBs take into account the degree of constraint satisfaction of the parents, etc.

We should point out that Fuzzy Logic-based techniques have been used for allowing Evolutionary Algorithms to solve these problems. In particular, in Van Le (1995) and Van Le (1996), a evolutionary approach is presented based on the *fuzzification* of the constrained optimization problems. In this method, the degrees of constraint satisfaction of the chromosomes are used as weight factors for calculating their fitness.

**FLCs-based Adaptive Distributed GAs.** The basic idea of the Distributed GAs lies in the partition of the population into several subpopulations, each one of them being processed by a GA,

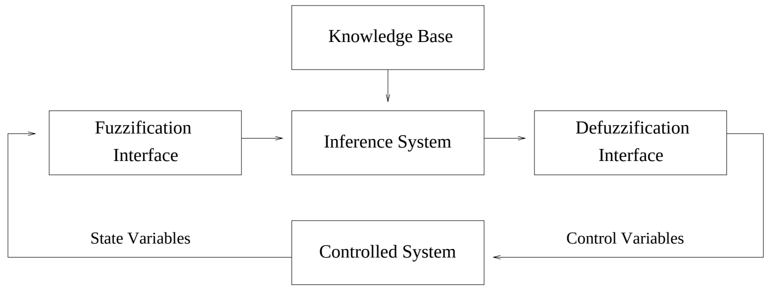


**Figure 10.** Generic structure of an FLC

independently from the others. Furthermore, a *migration* process produces a chromosome exchange between the subpopulations. Two important control parameters determine the operation of this process (Cantú-Paz, 1997): the *migration rate*, that controls how many chromosomes migrate and the *migration interval*, that specifies the number of generations between each migration. FLCs may be used for adapting these parameters, depending on the diversity and the convergence of the subpopulations. Furthermore, the application of Coevolution with Fuzzy Behaviors would be suitable for learning fuzzy rule bases that determine migration rates and migration intervals between pair of subpopulations.

**FLCs-based adaptation for Genetic Programming (GP) (Koza, 1992).** GP basic distinction from GAs is the evolution of dynamic tree structures, often interpreted as programs, rather than fixed-length vectors. It would be interesting to design FLCs for controlling diversity and convergence in a population of genetic programs, and apply the Coevolution with FBs (Subsection 4.4) for adapting the genetic operators that work with trees (this may be carried out by extending work appeared in Angeline (1996), Iba & de Garis (1996) and Teller (1996)).

# 6 Conclusions

In this paper, we have attempted to lay the foundations for FGAs. First, we proportionate different possible definitions for these algorithms, using the Zadeh's definition of Fuzzy Algorithms and under the two different meanings as FL may be viewed, a narrow interpretation and its wide sense. Then, we review different FGA approaches, ones involving GA components modeled using fuzzy tools and other being instances of FLCs-based AGAs. Finally, we have discussed future directions and some challenges for FGA research.

The good performance of the FGA approaches reviewed allows an important conclusion to be pointed out: *GAs may be improved through the use of FL*. Furthermore, the challenges outlined for future work show promise in the research on FGAs.

# Appendix A. Fuzzy Logic Controllers

An FLC is composed by a *knowledge base*, that includes the information given by the expert in the form of linguistic control fuzzy rules, a *fuzzification interface*, which has the effect of transforming crisp data into fuzzy sets, an *inference system*, that uses them together with the knowledge base to make inference by means of a reasoning method, and a *defuzzification interface*, that translates the fuzzy control action thus obtained to a real control action using a defuzzification method. The generic structure of an FLC is shown in Figure 10.

The knowledge base encodes the expert knowledge by means of a set of fuzzy control rules. A fuzzy control rule is a conditional statement with the form

"**If** *a set of conditions are satisfied* **Then** *a set of consequences can be inferred*"

in which the antecedent is a condition in its application domain, the consequent is a control action to be applied in the controlled system (notion of control fuzzy rule) and both antecedent and consequent are associated with fuzzy concepts, that is, linguistic terms (notion of fuzzy rule).

The knowledge base is composed of two components, a *data base*, containing the definitions of the fuzzy control rules linguistic labels, i.e., the membership functions of the fuzzy sets specifying the meaning of the linguistic terms, and a *fuzzy rule base*, constituted by the collection of fuzzy control rules representing the expert knowledge.

For more information about fuzzy systems and FLC, the following books may be studied: Driankow, Hellendoorn & Reinfrank (1993) and Yager & Zadeh (1992).

## References


Angeline, P.J. (1995). Adaptive and self-adaptive evolutionary computations. In M. Palaniswami, Y. Attikiouzel, R. Markc, D. Fogel & T. Fukuda (Eds.), *Computational Intelligence: A Dynamic Systems Perspective* (pp.152-163). Piscataway, NJ: IEEE Press.

Angeline, P.J. (1996). Two self-adaptive crossover operators for genetic programming. In P.J. Angeline & J.E. Kinnear, Jr. (Eds.), *Advances in Genetic Programming 2* (pp. 89-109). Cambridge, MA: MIT Press.

Arnone, S., Dell'Orto, M. & Tettamanzi, A. (1994). Toward a fuzzy government of genetic populations. In *Proc. of the 6th IEEE Conference on Tools with Artificial Intelligence* (pp. 585-591). Los Alamitos, CA: IEEE Computer Socity Press.

Bäck, T. (1992). The interaction of mutation rate, selection, and self-adaptation within genetic algorithm. In R. Männer & B. Manderick (Eds.), *Parallel Problem Solving from Nature 2* (pp. 85-94). Amsterdam: Elsevier Science Publishers.

Bäck, T. (1994). Selective pressure in evolutionary algorithms: a characterization of selection mechanisms. In Z. Michalewicz, J.D. Schaffer, H.-P. Schwefel, D.B. Fogel & H. Kitano (Eds.), *Proceeding of the First IEEE International Conference on Evolutionary Computation* (pp. 57-62). Piscataway, NJ: IEEE Press.

Bäck, T. & Kursawe, F. (1994). Evolutionary algorithms for fuzzy logic: a brief overview. In *Fifth International Conference IPMU: Information Processing and Management of Uncertainty in Knowledge-Based Systems* (pp. 659-664).

Baker, J.E. (1985). Adaptive selection methods for genetic algorithms. In *Proc. First Int. Conf. on Genetic Algorithms* (pp. 101-111). Hillsdale, MA: L. Erlbaum Associates.

Bergmann, A., Burgard, W. & Hemker, A. (1994). Adjusting parameters of genetic algorithms by fuzzy control rules. In K.-H. Becks & D. Perret-Gallix (Eds.), *New Computing Techniques in Physics Research III.* Singapore: World Scientific Press.

Buckley, J.J. & Hayashi, Y. (1994). Fuzzy genetic algorithm and applications. *Fuzzy Sets and Systems*, 61, 129–136.

Cantú-Paz, E. (1997). *A survey of parallel genetic algorithms.* IlliGAL Report 97003, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, IL.

Cordon, O., Herrera, F. & Lozano, M. (1997). On the combination of fuzzy logic and evolutionary computation: a short review and bibliography. In W. Pedrycz (Ed.), *Fuzzy Evolutionary Computation* (pp. 33-56). Kluwer Academic.

Davis, L. (1991). *Handbook of Genetic Algorithms.* New York: Van Nostrand Reinhold.

De Jong, K.A. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* Doctoral dissertation, University of Michigan, Ann Arbor. Dissertation Abstracts International, 36(10), 5140B (University Microfilms No 76-9381).

Driankow, D., Hellendoorn, H. & Reinfrank, M. (1993). *An Introduction to Fuzzy Control.* Springer-Verlag.

Eiben, A.E. (1997). Multi-parent recombination. In T. Bäck, D. Fogel & Z. Michalewicz (Eds.), *Handbook of Evolutionary Algorithms* (pp. 25-33). IOP Publishing Ltd. and Oxford University Press.

Eshelman, L.J., Mathias, K.E. & Schaffer, J.D. (1997). Crossover operator biases: exploiting the population distribution. In T. Bäck (Ed.), *Proc. of the Seventh Int. Conf. on Genetic Algorithms* (354-361). San Mateo: Morgan Kaufmann.

Fogel, D.B. (1994). An introduction to simulated evolutionary optimization. *IEEE Transactions on Neural Networks*, 5(1), 3-14.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.

Grefenstette, J.J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16, 122-128.

Iba, H. & de Garis, H. (1996). Extending genetic programming with recombinative guidance. In P.J. Angeline & J.E. Kinnear, Jr. (Eds.), *Advances in Genetic Programming 2* (pp. 69-88). Cambridge, MA: MIT Press.

Herrera, F., Herrera-Viedma, E., Lozano, M. & Verdegay, J.L. (1994). Fuzzy tools to improve genetic algorithms. In *Proc. of the European Congress on Intelligent Techniques and Soft Computing* (pp. 1532-1539).

Herrera, F. & Lozano, M. (1996a). Adaptation of genetic algorithm parameters based on fuzzy logic controllers. In F. Herrera & J.L. Verdegay (Eds.), *Genetic Algorithms and Soft Computing* (pp. 95-125). Physica-Verlag.

Herrera, F. & Lozano, M. (1996b). Heuristic crossover for real-coded genetic algorithms based on fuzzy connectives. In H-M. Voigt, W. Ebeling, I. Rechenberg & H-P. Schwefel (Eds.), *4th International Conference on Parallel Problem Solving from Nature* (pp. 336-345). Lecture Notes in Computer Science 1141, Berlin: Springer-Verlag.

Herrera, F. & Lozano, M. (1998). *Adaptive genetic algorithms based on coevolution with fuzzy behaviors.* Technical Report #DECSAI-98-01-05, Dept. of Computer Science and A. I., University of Granada, Spain. (Available at the URL address: http://decsai.ugr.es/~lozano/public.html)

Herrera, F., Lozano, M. & Verdegay, J.L. (1994). Applying genetic algorithms in fuzzy optimization problems. *Fuzzy Sets & Artificial Intelligence*, 3, 39-52.

Herrera, F., Lozano, M. & Verdegay, J.L. (1996). Dynamic and heuristic fuzzy connectives-based crossover operators for controlling the diversity and convengence of real-coded genetic algorithms. *Int. Journal of Intelligent Systems*, 11, 1013-1041.

Herrera, F., Lozano, M. & Verdegay, J.L. (1997). Fuzzy connectives based crossover operators to model genetic algorithms population diversity. *Fuzzy Sets and Systems*, 92(1), 21-30.

Herrera, F., Lozano, M. & Verdegay, J.L. (1998). Tackling real-coded genetic algorithms: operators and tools for the behavioral analysis. To appear in *Artificial Intelligence Reviews*.

Herrera, F. & Verdegay, J.L. (Eds.) (1996). *Genetic Algorithms and Soft Computing*. Physica-Verlag (Studies in Fuzziness and Soft Computing, Vol. 8).

Hinterding, R., Michalewicz, Z. & Eiben, A.E. (1997). Adaptation in evolutionary computation: a survey. In *Proc of the 4th IEEE Conf. on Evolutionary Computation* (pp. 65-69). Piscataway, NJ: IEEE Press.

Koza, J.R. (1992). *Genetic Programing: On the Programming of Computers by Means of Natural Selection*. Cambridge: The MIT press.

Lee, M.A. & Takagi, H. (1993). Dynamic control of genetic algorithms using fuzzy logic techniques. In S. Forrest (Ed.), *Proc. of the Fifth Int. Conf. on Genetic Algorithms* (pp. 76-83). San Mateo: Morgan Kaufmann.

Lee, M.A. & Takagi, H. (1994). A framework for studying the effects of dynamic crossover, mutation, and population sizing in genetic algorithms. In T. Furuhashi (Ed.), *Advances in Fuzzy Logic, Neural Networks and Genetic Algorithms* (pp. 111-126). Lecture Notes in Computer Science 1011, Berlin: Springer-Verlag.

Liepins, G.E. & Vose, M.D. (1992). Characterizing crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 5, 27-34.

Mahfoud, S.W. (1992). Crowding and preselection revised. In R. Männer & B. Manderick (Ed.), *Parallel Problem Solving from Nature 2* (pp. 27-36). Amsterdam: Elsevier Science Publichers.

Meyer, L. & Feng, X. (1994). A fuzzy stop criterion for genetic algorithms using performance estimation. In *Proc. Third IEEE Int. Conf. on Fuzzy Systems* (pp. 1990-1995).

Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. New York: Springer-Verlag.

Michalewicz, Z. & Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4(1), 1-32.

Mizumoto, M. (1989a). Pictorial representations of fuzzy connectives, part I: cases of t-norms, t-conorms and averaging operators. *Fuzzy Sets and Systems*, 31, 217-242.

Mizumoto, M. (1989b). Pictorial representations of fuzzy connectives, part II: cases of compensatory operators and seft-dual operators. *Fuzzy Sets and Systems*, 32, 45-79.

Mühlenbein, H. & Schlierkamp-Voosen, D. (1993). Predictive models for the breeder genetic algorithm I. continuous parameter optimization. *Evolutionary Computation*, 1, 25-49.

Ono, I. & Kobayashi, S. (1997). A real-coded genetic agorithm for function optimization using unimodal normal distribution crossover. In T. Bäck (Ed.), *Proc. of the Seventh Int. Conf. on Genetic Algorithms* (pp. 246-253). San Francisco: Morgan Kaufmann Publishers.

Reynolds, R.G. & Chung, C-J. (1997). Regulating the amount of information used for self-adaptation in cultural algorithms. In T. Bäck (Ed.), *Proc. of the Seventh Int. Conf. on Genetic Algorithms* (pp. 401-408). San Francisco: Morgan Kaufmann Publishers.

Ronald, E. (1993). When selection meets seduction. In S. Forrest (Ed.), *Proc. of the Fifth Int. Conf. on Genetic Algorithms* (pp. 167-173). San Mateo: Morgan Kaufmann.

Sanchez, E. (1993). Fuzzy genetic algorithms in soft computing environment. Invited Plenary Lecture in *Fifth IFSA World Congress*, Seoul.

Sanchez, E., Shibata, T. & Zadeh L.A. (Eds) (1996). *Genetic Algorithms and Fuzzy Logic Systems. Soft Computing Perspectives*. World Scientific.

Schwefel, H-P. (1995). *Evolution and Optimum Seeking. Sixth-Generation Computer Technology Series*. New York: Wiley.

Smith, J.E. & Fogarty, T.C. (1997). Operator and parameter adaptation in genetic algorithms. *Soft Computing*, 1(2), 81-87.

Teller, A. (1996). Evolving programmers: the co-evolution of intelligent recombination operators. In P.J. Angeline & J.E. Kinnear, Jr. (Eds.), *Advances in Genetic Programming 2* (pp. 45-68). Cambridge, MA: MIT Press.

Tettamanzi, A.G. (1995). Evolutionary algorithms and fuzzy logic: a two-way integration. In *2nd Joint Conference on Information Sciences* (pp. 464-467), Wrightsville Beach, NC.

Tuson, A.L. & Ross, P. (1996). Co-evolution of operator settings in genetic algorithms. In T.C. Fogarty (Ed.), *Evolutionary Computing, AISB Workshop* (pp. 286-296). Lecture Notes in Computer Science 1143, Berlin: Springer-Verlag.

Van Le, T. (1995). A fuzzy evolutionary approach to solving constraint problems. In *Proc. 1995 IEEE Int. Conf. on Evolutionary Computation* (pp. 317-319). Piscataway, NJ: IEEE Press.

Van Le, T. (1996). A fuzzy evolutionary approach to constrained optimisation problems. *Proc. 1996 IEEE Int. Conf. on Evolutionary Computation* (pp. 274-278). Piscataway, NJ: IEEE Press.

Voigt, H.M. (1992). *Fuzzy evolutionary algorithms*. Technical Report tr-92-038, International Computer Science Institute (ICSI), Berkeley.

Voigt, H.-M. (1995). Soft genetic operators in evolutionary algorithms. In W. Banzhaf & F.H. Eeckman (Eds.), *Evolution and Biocomputation* (pp. 123-141). Lecture Notes in Computer Science 899, Berlin: Springer-Verlag.

Voigt, H. M. & Anheyer, T. (1994). Modal mutations in evolutionary algorithms. In Z. Michalewicz, J.D. Schaffer, H.-P. Schwefel, D.B. Fogel & H. Kitano (Eds.), *Proceeding of the First IEEE International Conference on Evolutionary Computation* (pp. 88-92). Piscataway, NJ: IEEE Press.

Voigt, H.M., Born, J. & Santibáñez-Koref, I. (1993). *A multivalued evolutionary algorithms.* Technical Report tr-93-022, International Computer Science Institute (ICSI), Berkeley.

Voigt, H.M., Mühlenbein, H. & Cvetković, D. (1995). Fuzzy recombination for the breeder genetic algorithm. In L. Eshelman (Ed.), *Proc. of the Sixth Int. Conf. on Genetic Algorithms* (pp. 104-111). San Francisco: Morgan Kaufmann Publishers.

Voget, S. (1996). Multiobjective optimization with genetic algorithms and fuzzy-control. In *Proc of the Fourth European Congress on Intelligent Techniques and Soft Computing* (pp. 391-394).

Voget, S. & Kolonko, M. (1998). Multi-criteria optimization with a fuzzy-genetic-algorithm. *To appear in Journal of Heuristic.*

Wang, P.Y., Wang, G.S. & Hu, Z.G. (1997). Speeding up the search process of genetic algorithm by fuzzy logic. In *Proc. of the European Congress on Intelligent Techniques and Soft Computing* (pp. 665-671).

Xu, H.Y. & Vukovich, G. (1993). A fuzzy genetic algorithm with effective search and optimization. In *Proc. of 1993 International Joint Conference on Neural Networks* (pp. 2967-2970).

Xu, H.Y., Vukovich, G., Ichikawa, Y. & Ishii, Y. (1994). Fuzzy evolutionary algorithms and automatic robot trajectory generation. In Z. Michalewicz, J.D. Schaffer, H.-P. Schwefel, D.B. Fogel & H. Kitano (Eds.), *Proceeding of the First IEEE International Conference on Evolutionary Computation* (pp. 595-600). Piscataway, NJ: IEEE Press.

Yager, R.R. & Zadeh, L.A. (Eds.) (1992). *An Introduction to Fuzzy Logic Applications in Intelligent Systems.* Kluwer Academic Press.

Zadeh, L.A. (1968). Fuzzy algorithms. *Information and Control*, 12, 94-102.

Zadeh, L.A. (1973). Outline of a new approach to the analysis of the complex systems and decision processes. *IEEE Trans. Systems, Man and Cybernetics*, SMC-3, 28-44.

Zadeh, L.A. (1993). Foreword. In P.Z. Wang & K.V. Loe (Eds.), *Between Mind and Computer: Fuzzy Science and Engineering.* World Scientific.

Zeng, X. & Rabenasolo, B. (1997). A fuzzy logic based design for adaptive genetic algorithms. In *Proc. of the European Congress on Intelligent Techniques and Soft Computing* (pp. 660-664).