The Use of Fuzzy Connectives to Design Real-Coded Genetic Algorithms[*]

F. Herrera, M. Lozano and J.L. Verdegay
Dept. of Computer Science and Artificial Intelligence
University of Granada, 18071 - Granada, Spain
e-mail: herrera, lozano, verdegay@robinson.ugr.es

Abstract

Genetic algorithms are adaptive methods that use principles inspired by natural population genetics to evolve solutions to search and optimization problems. Genetic algorithms process a population of search space solutions with three operations: selection, crossover and mutation.

A great problem in the use of genetic algorithms is premature convergence; the search becomes trapped in a local optimum before the global optimum is found. Fuzzy logic techniques may be used for solving this problem. This paper presents one of them: the design of crossover operators for real-coded genetic algorithms using fuzzy connectives and its extension based on the use of parameterized fuzzy connectives as tools for tackling the premature convergence problem.

Keywords: Genetic Algorithms, Real Coding, Fuzzy Connectives.

1 Introduction

Genetic algorithms (GAs) are search algorithms that use operations found in natural genetics to guide the trek through a search space. GAs are theoretically and empirically proven to provide robust search in complex spaces, giving a valid approach to problems requiring efficient and effective search ([6]).

A GA starts with a population of randomly generated solutions, chromosomes, and advance towards better solutions by applying genetic operators modeled on the genetic processes occurring in nature. In these algorithms we maintain a population of solutions for a given problem; this population undergoes evolution in a form of natural selection. In each generation, relatively good solutions reproduce to give offspring that replace the relatively bad solutions which die. An evaluation of fitness function plays the role of the environment to distinguish between good and bad solutions.

Although there are many possible variants of the basic GA, the fundamental underlying mechanism operates on a population of chromosomes (representing possible solutions to the problem) and consists of the following operations which are applied during each generation t:

1. evaluation of individual fitness,

2. formation of a gene pool by choosing individuals in proportion to their relative fitness,

3. recombination by means of the genetic operator crossover and mutation.

The process is iterated until the system ceases to improve or a given generation T is reached.

Fixed-length and binary encoded strings for representation solution have dominated GA research, since there exist theoretical results that show them to be the most appropriate, and they are amenable to simple implementation. But the GA's good properties do not stem from the use of bit strings ([1]). For this reason, the path has been lain towards the use of alphabets with a higher cardinal, followed by the development of new genetic operators (crossover and mutation) on these alphabets. Non binary encodings include real number representations, which would seem particularly natural when we are tackling optimization problems of parameters with variables in continuous domains. Then a chromosome is a vector of floating point numbers whose size is kept the same as the length of the vector which is the solution to the problem. GAs with this type of encoding will be called real-coded GAs (RCGAs).

The use of real parameters makes it possible to use large domains (even unknown domains) for the variables, which is difficult to achieve in binary implementations, where increasing the domain would mean sacrificing precision, assuming a fixed length for the chromosomes. Another advantage when using real parameters is their capacity to exploit the graduality of the functions with continuous variables (where the concept of graduality refers to the fact that slight changes in the variables correspond to slight changes in the function). Lastly, they allow the tools that handle non-trivial restrictions to be designed more easily, and the representation of the solutions is very close to the natural formulation of many problems.

The mutation operator arbitrarily alters one or more components (called genes) of a selected structure so as to increase the structural variability of the population, e.g., it explores the search space. Under binary coding, given a gene with value "0", it is replaced by "1", and viceversa. Under real coding different versions of this operator were presented ([9]). Each gene of each chromosome in the population undergoes a random change according to a probability defined by the mutation rate, the mutation probability, p_m.

The crossover operator exploits the available information from the population about the search space. It combines the features of two parent structures to form two similar offspring. The classical crossover operator under binary coding builds

an offspring by linking together two gene segments, each one belonging to a different parent. This operator is applied with a probability of performance, the crossover probability, p_c, that determines the number of chromosomes in the population to be crossed. [7, 9] report crossover operator models under binary and real coding.

The crossover operator plays a central role in the GA's performance. It could be considered to be one of the algorithm's defining characteristics, and it is one of the components to bear in mind for improving the GA's behaviour ([12]). The mutation operator is a complement of the crossover operator. It is needed to avoid the loss of useful information produced by the crossover.

An important problem in the use of GAs is *premature convergence*; the search becomes trapped in a local optimum before the global optimum is found. This is produced by the lack of *diversity* in the population and a disproportionate *exploitation/exploration* relationship; an adequate balance between a broad search and a sufficient refinement is not established.

Fuzzy logic techniques may be used to solve these problems. An attempt consists of the use of fuzzy logic based systems for the dynamic control parameters of RCGA (p_m, p_c, population size, etc) in such a way that the correct exploitation/exploration relationship and suitable diversity levels became established ([8, 11]). Another one uses fuzzy connectives to design crossover operators ([8]).

In this paper we present crossover operators for RCGAs based on the use of fuzzy connectives, and their extension based on the use of parameterized fuzzy connectives for designing dynamic crossover operators with the main objective of introducing population diversity in the GA search.

2 Design of Crossover Operators for RCGA using Fuzzy Connectives

As has already been pointed out, a chromosome is a vector of real numbers, and its precision will be marked by that of the computer under which the algorithm is carried out. The size of the chromosome is kept the same as the length of the vector that is the solution to the problem; in this way, each gene represents a variable of the problem. The values of the chromosome genes are forced to remain in the interval established by the variable that the chromosome represents, so the genetic operators must preserve this requirement.

In [3] it was pointed out that the crossover operator is a key point for solving the premature convergence problem. Thus, solutions to this problem may be found by designing new alternatives to this operator. Here, the development of such crossover operators is attempted. We present crossover operators for RCGA based on the use of fuzzy connectives: t-norms, t-conorms, average functions and generalized compensation operators ([14, 15]) which induce different diversity levels in the population, and therefore the premature convergence problem may be controlled.

Let us assume that the chromosomes $C_1 = (c_1^1, c_2^1, ..., c_n^1)$ and $C_2 = (c_1^2, c_2^2, ..., c_n^2)$

are selected to apply the crossover operator to them, and two genes c_i^1 and c_i^2 to be crossed over, $c_i^1, c_i^2 \in [m_i, M_i]$, being $x_i = \min(c_i^1, c_i^2)$ and $y_i = \max(c_i^1, c_i^2)$. It seems reasonable to imagine the possibility of obtaining good descendents outside this interval. In short, the interval of action of the gene i, $[m_i, M_i]$, may be divided into three regions $[m_i, x_i]$, $[x_i, y_i]$, and $[y_i, M_i]$, where good descendents may be obtained; even considering a region $[x_i', y_i']$ with $x_i' \leq x_i$ and $y_i' \geq y_i$ would seem reasonable. Graphically

Figure 1: Action interval for a gene

We shall now go on to put forward a set of crossover operators that allow descendents to be obtained in the previous intervals. In order to do that, we use four functions F, S, M and L defined from $[a, b] \times [a, b]$ in $[a, b]$, $a, b \in \Re$, which fulfill:

(P1) $\forall x, y \in [a, b]$ $F(x, y) \leq \min(x, y)$,

(P2) $\forall x, y \in [a, b]$ $S(x, y) \geq \max(x, y)$,

(P3) $\forall x, y \in [a, b]$ $\min(x, y) \leq M(x, y) \leq \max(x, y)$,

(P4) $\forall x, y \in [a, b]$ $F(x, y) \leq L(x, y) \leq S(x, y)$,

(P5) F, S, M, and L are monotone non-decreasing.

Let us assume that $Q \in \{F, S, M, L\}$, and $C_1 = (c_1^1 \ldots c_n^1)$ and $C_2 = (c_1^2 \ldots c_n^2)$ are two chromosomes that have been selected to apply the crossover operator to them. We may generate the chromosome $H = (h_1 \ldots h_n)$ as

$$H = Q(C_1, C_2), \ h_i = Q(c_i^1, c_i^2), \quad i = 1, \ldots, n.$$

With the t-norm operators, t-conorms, averaging functions and generalized compensation operators used as fuzzy connectives, we shall associate F with a t-norm, S with a t-conorm, M with an averaging operator and L with a generalized compensation operator. First, we need a set of linear transformations to be able to apply these operators under the gene definition intervals.

Let O be an operator belonging to the set formed by the t-norms, t-conorms, averaging functions and generalized compensation operators. For each position $i \in \{1,...n\}$, the following operations will be carried out:

1. Transform c_i^1 and c_i^2 into the values $s_i^1, s_i^2 \in [0,1]$ such that

$$s_i^k = \frac{c_i^k - m_i}{M_i - m_i}, \quad k = 1, 2.$$

   Using this step, we transform the values of the genes so that the operator may be applied to them.

2. Apply the operator $O(s_i^1, s_i^2)$ and calculate the value h_i:

$$h_i = m_i + (M_i - m_i) \cdot O(s_i^1, s_i^2),$$

   so that the gene h_i is in relation to its original limits, $h_i \in [m_i, M_i]$.

3. h_i will be the value for the gene at position i of the chromosome resulting from the crossover of the chromosomes C_1 and C_2.

Complying with a set of fuzzy connectives, $(T_j, G_j, P_j, \hat{C}_j)$, $j = 1, ..., k$, a set of functions F_j, S_j, M_j and L_j $j = 1, ..., k$ is built as we describe below:

$$F_j(c_i^1, c_i^2) = m_i + (M_i - m_i) \cdot T_j(s_i^1, s_i^2)$$
$$S_j(c_i^1, c_i^2) = m_i + (M_i - m_i) \cdot G_j(s_i^1, s_i^2)$$
$$M_j(c_i^1, c_i^2) = m_i + (M_i - m_i) \cdot P_j(s_i^1, s_i^2)$$
$$L_j(c_i^1, c_i^2) = m_i + (M_i - m_i) \cdot \hat{C}_j(s_i^1, s_i^2)$$

These crossover operators have different features: the F- and S-crossovers show exploration, the M-crossover operators show exploitation and the L-crossover show relaxed exploitation.

3 Example

We have carried out different experiments that help to compare the behaviour of a binary coded GA, and some RCGAs with crossover operators that have been proposed in other publications, with a set of algorithms based on the crossover operators proposed, which use the fuzzy connectives $(T_j, G_j, P_j, \hat{C}_j)$, $j = 1, ..., 5$ showed in Table 1.

t-norm	t-conorm	Averaging Operator
Logical Product $T_1(x,y) = \min(x,y)$	Logical Sum $G_1(x,y) = \max(x,y)$	$P_1(x,y) = (1-p)x + py$
Hamacher Product $T_2(x,y) = \frac{xy}{x+y-xy}$	Hamacher Sum $G_2(x,y) = \frac{x+y-2xy}{1-xy}$	$f(x) = \frac{1-x}{x}$ $P_2(x,y) = \frac{1}{\frac{y-yp-xy+xp}{xy}+1}$
Algebraic Product $T_3(x,y) = xy$	Algebraic Sum $G_3(x,y) = x+y-xy$	$f(x) = -\log x$ $P_3(x,y) = x^{1-p}y^p$
Einstein Product $T_4(x,y) = \frac{xy}{1+(1-x)(1-y)}$	Einstein Sum $G_4(x,y) = \frac{x+y}{1+xy}$	$f(x) = \log\frac{2-x}{x}$ $P_4(x,y) = \frac{2}{1+(\frac{2-x}{x})^p(\frac{2-y}{y})^{1-p}}$
Bounded Product $T_5(x,y) = 0 \vee (x+y-1)$	Bounded Sum $G_5(x,y) = 1 \wedge (x+y)$	$f(x) = 1-x$ $P_5(x,y) = (1-p)x + py$

Table 1: *Set of Operators*

Each t-conorm is dual to the t-norm shown to its left. The averaging function is calculated from the formula of the quasi-arithmetic averages, using as the f function the additive generator function of the t-norm placed in the same line, except for the first one which, not being Archimedean, does not have a generator function, and we shall use $f(x) = x$. This function is shown in the upper part of the cells where these averaging operators are specified. For each family of operators in Table 1, a generalized compensation operator $\hat{C}_j$ will be considered, defined as follows:

$$\hat{C}_j = P_j(T_j, G_j)$$

For the family of Logical operators, we shall consider $\hat{C}_1 = T_1^{1-p}.S_1^p$.

The GA families are differentiated according to how they carry out the following two steps:

1. Generation of offspring using the different crossover operators.

2. Selection of offspring resulting from the crossover which will form part of the population.

A proposal is the following: For each pair of chromosomes from a total of $\frac{1}{2} \cdot p_c \cdot N$ (p_c crossover probability, N population size), four offspring are generated, the result of applying specific functions F, S, M, and L to them. The two most promising offspring of the four replace their parents in the population.

This selection strategy introduces a high exploitation level with an underlying exploration caused by the use of the different Q-crossover operators.

Next, the results on Rosenbrock's Generalized function ([4]) are shown. The analytical and graphical formulation together with the element that represents the global optimum (minimum) are:

$$f(\vec{x}) = \sum_{i=1}^{n-1} (100 \cdot (x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$$

$$-5.12 \leq x_i \leq 5.12$$

$$\min(f) = f(1, \ldots, 1) = 0$$

Figure 2:

where $n = 5$.

A set of nine RCGA based on crossover operators presented in the literature were considered for the experiments (RGA1-RGA9). Below there is a table indicating the type of crossover and mutation used by each of them, together with their names.

Algorithms	*Mutation*	*Crossover*	
RGA1	Random	Simple	[13, 17]
RGA2	Non-Uniform	Simple	[13, 17]
RGA3	Random	Uniform Arithmetical	[13]
		$a = 0.35$	
RGA4	Non-Uniform	Uniform Arithmetical	[13]
		$a = 0.35$	
RGA5-α	Non-Uniform	BLX-α	[5]
		$(\alpha = 0, .15, .3, .5)$	
RGA6	Non-Uniform	Discrete	[16]
RGA7	Non-Uniform	Linear	[17]
RGA8	Non-Uniform	Extended Intermediate	[16]
RGA9	Non-Uniform	Extended Line	[16]

Table 2: *Real Coded Genetic Algorithms*

By BGA we denote a binary coded GA with 30 genes per variable, multiple crossover with two points and proportional selection probability.

By $NRGA1, ..., NRGA5$ we denote a family of GA based on the fuzzy connectives crossover and the generation and selection of offspring proposals, using the families of fuzzy connectives: Logical, Hamamcher, Algebraic, Einstein and Bound, respectively. In all cases we use the stochastic universal sampling ([2]) selection procedure and the elitist model. We carried out our experiments using the following parameters: the population size is 61 individuals, the crossover probability

$p_c = 0.6$, and the probability of chromosome update $p_u = p_m \cdot 5 = 0.6$, and the parameter b used by the non- uniform mutation is 5. We executed all the algorithms 3 times, each one with 100, 500 and 5000 generations, and present the average value of them in Table 3.

Algorithms	100	500	5000
BGA	1.1262e+01	3.6069e+00	1.9045e+00
RGA1	3.0446e+01	2.0070e+01	6.0669e+00
RGA2	6.9230e+00	2.1448e+00	4.7343e-01
RGA3	4.1941e+00	8.6031e+00	6.3745e+00
RGA4	3.7915e+00	2.9624e+00	8.9244e-01
RGA5-0.0	4.5504e+00	2.3454e+00	9.1602e-01
RGA5-0.15	3.2157e+00	2.9556e+00	7.0929e-01
RGA5-0.3	3.7477e+00	1.5844e+00	4.8854e-01
RGA5-0.5	8.2653e+00	2.1099e+00	1.7329e+00
RGA6	5.5393e+00	2.8379e+00	3.5106e-01
RGA7	4.7115e+00	1.8487e+00	5.1499e-01
RGA8	4.6861e+00	3.5337e+00	5.3325e-01
RGA9	4.2196e+00	2.9374e+00	3.8014e-02
NRGA1	4.1431e+00	1.9687e+00	4.9364e-03
NRGA2	1.2378e+01	7.3868e-01	4.0848e-02
NRGA3	1.0031e+01	3.5037e+00	1.0099e+00
NRGA4	1.2425e+01	4.3985e+00	1.8347e+00
NRGA5	6.0121e+00	2.6930e+00	1.2150e+00

Table 3: *Results*

The best behaviour corresponds to the Logical crossover operator. This operator together with the offspring selection mechanism offer a suitable exploitation/exploration balance, although we must point out that this mechanism is more time expensive because need more evaluations. Other offspring selection mechanisms are proposed in [10].

4 Design of Dynamic Crossover Operators Using Parameterized Fuzzy Connectives

An idea for avoiding the premature convergence consists in allowing the exploration in the beginning of the search process and the exploitation at the end of it. With the exploration the diversity becames greater, increasing the probability of finding zones which are closed to optimal solutions. Then, supposing that the population have information about these zones, the convergence towards the optimum is produced through exploitation.

A mutation operator for RCGA called non-uniform mutation ([13]) is based on the aforementioned principle. The proportion in which a real gene is mutated decreases as the GA's execution advance. Thus, the changes produced on the genes are smaller in the last generations producing a local tuning.

We may extend the use of the crossover operators presented in order to follow the aforementioned ideas. Different fuzzy connectives could be used during the GA's run. Firstly, we shall use fuzzy connectives that produce high diversity levels, and later other ones producing a sufficient diversity level to allow the convergence to be reached.

We present a set of dynamic crossover operators based on the use of parameterized t-norms, t-conorms and averaging functions ([14, 15]). In the first stages, we shall use t-norms and t-conorms distant from the minimum t-norm and maximum t-conorm respectively, so high diversity is induced. Later, t-norms close to the minimum and t-conorms close to the maximum are considered. The convergence is caused and the good behaviour of the logical fuzzy connectives will be kept. As was shown in Table 3, these operators are in general the most profitable.

To do this, we propose a set of crossover operators based on the functions families: $\{F^p\}_{p=1,...,G}$, $\{S^p\}_{p=1,...,G}$, and $\{M^p\}_{p=1,...,G}$, $G \in N$ defined from $[a,b] \times [a,b]$ in $[a,b]$, $a,b \in \Re$, which fulfill the corresponding P1-P5 properties and:

(P6) $\forall x,y \in [a,b]$, $\displaystyle\lim_{p \to G} F^p(x,y) \cong \min(x,y)$

(P7) $\forall x,y \in [a,b]$, $\displaystyle\lim_{p \to G} S^p(x,y) \cong \max(x,y)$

(P8) $\forall x,y \in [a,b]$, $\min(x,y) \leq M^p(x,y) \leq \frac{x+y}{2}$ or $\frac{x+y}{2} \leq M^p(x,y) \leq \max(x,y)$
and $\forall x,y \in [a,b]$, $\displaystyle\lim_{p \to G} M^p(x,y) \cong \frac{x+y}{2}$

Let us consider a GA with a maximum number of generations τ and $C_1^t = (c_1^{1t},...c_n^{1t})$ and $C_2^t = (c_1^{2t},...c_n^{2t})$ two chromosomes that were selected in the generation t to apply the crossover operator to them. If $Q^p \in \{F^p, S^p, M^p\}$ $p = 1,...,\tau$ we may generate the chromosome $H^t = (h_1^t,...,h_n^t)$ as

$$H^t = Q^t(C_1^t, C_2^t), \quad h_i^t = Q^t(c_i^{1t}, c_i^{2t}), \quad i = 1,...,n.$$

We shall build functions families with the (P6) and (P7) properties using the parameterized t-norms and t-conorms described in Table 4. Table 5 shows the properties of the parameterized t-norms in this table. The properties of the parameterized t-conorms are analogous. We must point out that T_6 is the drastic t-norm (the smallest t-norm).

Type	t-norm	t-conorm
Yager $(q > 0)$	$T_1^q(x,y) =$ $= 1 - (1 \wedge \sqrt[q]{(1-x)^q + (1-y)^q})$	$G_1^q(x,y) = 1 \wedge \sqrt[q]{x^q + y^q}$
Frank $(q > 0)$ $(q \neq 1)$	$T_2^q(x,y) = log_q[1 + \frac{(q^x - 1)(q^y - 1)}{q-1}]$	$G_2^q(x,y) =$ $= 1 - log_q[1 + \frac{(q^{1-x} - 1)(q^{1-y} - 1)}{q-1}]$
Dombi $(q > 0)$	$T_3^q(x,y) = \frac{1}{1 + \sqrt[q]{(\frac{1-x}{x})^q + (\frac{1-y}{y})^q}}$	$G_3^q(x,y) = 1 - \frac{1}{1 + \sqrt[q]{(\frac{x}{1-x})^q + (\frac{y}{1-y})^q}}$
Dubois $(0 \leq q \leq 1)$	$T_4^q(x,y) = \frac{xy}{x \vee y \vee q}$	$G_4^q(x,y) = 1 - \frac{(1-x)(1-y)}{(1-x) \vee (1-y) \vee q}$

Table 4: *Dynamic Operators*

Tipo	T_6	T_5	T_4	T_3	T_2	T_1
Yager	$0 \leftarrow$	1				∞
Frank		∞		$\rightarrow 1$		$\rightarrow 0$
Dombi	$0 \leftarrow$				1	∞
Dubois				1		0

Table 5: *Properties of the Dynamic Operators*

We may observe that there is an order relation in the convergence rate towards the minimum among these operators, which is: Yager, Dombi, Frank and Dubois.

$\{F^p\}_{p=1,...,\tau}$ and $\{S^p\}_{p=1,...,\tau}$ functions families may be built using a parameterized t-norm T^q converging on the minimum and a parameterized t-conorm converging on the maximum, respectively ([14, 15]). To do so, we need a function $\delta(\cdot)$ for transforming the values of p, $\{1,...,\tau\}$, into the range of the parameter q such as $q = \delta(p)$ with $\lim_{p \to \tau} \delta(p) = \delta^*$ and $T^{\delta^*} \cong \min(x,y)$, $G^{\delta^*} \cong \max(x,y)$.

We may obtain a family of functions $\{M^p\}_{p=1,...,\tau}$ using parameterized averaging functions. An example of these functions is:

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

which is computed through the generalized mean defined by:

$$Mg(x,y) = f^{-1}((1-\lambda)f(x) + \lambda f(y)) \ (0 \leq \lambda \leq 1)$$

with $f(x) = x^q$ and $\lambda = \frac{1}{2}$.

The properties of these operators are ([14]):

q	$P^q(x, y)$
$-\infty$	$x \wedge y$
-1	$\frac{2xy}{x+y}$
0	$\sqrt{xy}$
1	$\frac{x+y}{2}$
$+\infty$	$x \vee y$

Table 6: *Properties of $P^q(\cdot, \cdot)$*

We build $\{M^p\}$ using P^q with $q = \psi(p)$ where

i) $\psi : [1, \tau] \rightarrow [-\infty, 1]$ is an increasing monotonous function, or

ii) $\psi : [1, \tau] \rightarrow [1, +\infty]$ is a decreasing monotonous function.

Four GAs based on the use of the dynamic crossover operators presented were built, which were denoted as DRGA1-DRGA4.

For each pair of chromosomes from the total population that undergoes crossover, four offspring are generated, the result of applying functions F^p, S^p, and two M^p, each one fulfilling a different part of (P8). The two most promising offspring of the four replace their parents in the population.

Table 7 shows the $\delta(\cdot)$ functions related with each one of the dynamic operators used.

Operators	$\delta(p)$
Yager	$\frac{1}{\ln(\frac{\tau+1}{p})}$
Frank	$\ln(\frac{\tau+1}{p})$
Dombi	$\frac{1}{\ln(\frac{\tau+1}{p})}$
Dubois	$\frac{1}{\sqrt{p}}$

Table 7: *Functions $\delta(\cdot)$*

The function $\psi(p)$ used is $1 + \ln(\frac{p}{\tau})$ as increasing monotonous function and $1 + \ln(\frac{\tau}{p})$ as decreasing monotonous function.

The average results returned by these algorithms under the above conditions are the following:

Algorithms	*100*	*500*	*5000*
DRGA1	5.149822e+00	3.004678e+00	8.060713e-01
DRGA2	8.111676e+00	2.855141e+00	5.509746e-01
DRGA3	8.175044e+00	2.343028e+00	8.944534e-01
DRGA4	2.073980e+00	8.210735e-01	1.406304e-03

Table 8: *Results of DRGA1-DRGA4*

The results show that the DRGA1-DRGA3 algorithms do not improve the results of Table 3. The reason is that the dynamic operators of Yager, Frank and

Dombi have a slow convergence towards the logical operators which are the ones with best behaviour. For that, these operators induced higher diversity levels during the GA's run than the Dubois operators, which cause smaller local tuning at the end of the GA. The case of DRGA4 is different: it shows the best results for 100 and 5000 generations, and the second place for 500 generations. The rapid convergence towards the logical operators shown by the Dubois operators (see Table 5) allows diversity at the beginning to be introduced and a similar behaviour as the algorithm NRGA1 (which is defined using Logical operators) to be reached quickly. The DRGA4 algorithm induced a strong exploitation/exploration relationship, offering good diversity levels.

5   Conclusions

In this paper we have presented the use of fuzzy connectives to design new crossover operators for RCGA. The results obtained show that by using this type of operators the performance of the RCGA is improved. These operators were defined for inducing different diversity levels in the population. This is a very important feature against premature convergence since a correct exploitation/exploration balance may be established and so this problem will be controlled. With the use of the dynamic operators the diversity levels may be changed during the GA's run, in such a way that this will be high at the beginning and slow at the end, showing a good GA performance.

References

[1]   Antonisse J., A new interpretation of schema notation that overturns the binary encoding constraint. Proc. of the Third Int. Conf. on Genetic Algorithms, J. D. Schaffer (Ed.), (Morgan Kaufmann Publishers, San Mateo, 1989), 86-91.

[2]   Baker. J. E., Reducing bias and inefficiency in the selection algorithm. Proc. of the Second Int. Conf. on Genetic Algorithms, (L. Erlbaum Associates, Hillsdale, MA, 1987), 14-21.

[3]   Booker, L., *Improving Search in Genetic Algorithms. Genetic Algorithms and Simulated Annealing*, L. Davis (Ed.), (Morgan Kaufmann Publishers, Los Altos, 1987), 61-73.

[4]   De Jong, K.A., *An analysis of the behavior of a class of genetic adaptive systems.* Doctoral dissertation, University of Michigan, (1975).

[5]   Eshelman L.J., Schaffer J.D., *Real-Coded Genetic Algorithms and Interval-Schemata.* Foundations of Genetic Algorithms-2, L. D. Whitley (Ed.), (Morgan Kaufmann Publishers, San Mateo, 1993), 187-202.

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

[7]   Herrera, F., Lozano, M., Verdegay, J.L., Algoritmos Genéticos: Fundamentos, Extensiones y Aplicaciones. Technical Report #DECSAI-94105, Dpto. of Computer Science and Artificial Intelligence, University of Granada, Spain, (1994). To appear in ARBOR. (In Spanish)

[8]   Herrera, F., Lozano, M., Verdegay, J.L., Fuzzy Tools to Improve Genetic Algorithms. Second European Congress on Intelligent Techniques and Soft Computing, September 20-23, (1994), Aachen, 1532-1539.

[9]   Herrera, F., Lozano, M., Verdegay, J.L., Tackling Real-Coded Genetic Algorithms: Operators and Tools for the Behaviour Analysis. Technical Report #DECSAI-95107, Dept. of Computer Science and Artificial Intelligence, University of Granada, Spain, (1995).

[10]   Herrera, F., Lozano, M., Verdegay, J.L., Fuzzy Connectives Based Crossover Operators to Model Genetic Algorithms Population Diversity. Technical Report #DECSAI-95110, Dept. of Computer Science and Artificial Intelligence, University of Granada, Spain, (1995).

[11]   Lee, M.A., Takagi, H., Dynamic Control of Genetic Algorithms using Fuzzy Logic Techniques. Proc. of the Fifth Int. Conf. on Genetic Algorithms, S. Forrest (Ed.), (Morgan Kaufmann Publishers, San Mateo, 1993), 76-83.

[12]   Liepins, G.E., Vose, M.D., Characterizing Crossover in Genetic Algorithms. *Annals of Mathematics and Artificial Intelligence* 5, 1, (1992), 27-34.

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

[14]   Mizumoto M., Pictorial Representations of fuzzy connectives, Part I: Cases of t-norms, t-conorms and averaging operators. *Fuzzy Sets and Systems* **31**, (1989), 217-242.

[15]   Mizumoto M., Pictorial Representations of fuzzy connectives, Part II: Cases of Compensatory Operators and Seft-dual Operators. *Fuzzy Sets and Systems* **32**, (1989), 45-79.

[16]   Mühlenbein H., Schlierkamp-Voosen D., Predictive Models for the Breeder Genetic Algorithm I. Continuous Parameter Optimization. *Evolutionary Computation* 1, 1, (1993), 25-49.

[17]   Wright A., *Genetic Algorithms for Real Parameter Optimization.* Foundations of Genetic Algorithms 1, G.J.E Rawlin (Ed.), (Morgan Kaufmann, San Mateo, 1991), 205-218.