International Journal on Artificial Intelligence Tools
Vol. 17, No. 6 (2008) 1131–1160
© World Scientific Publishing Company

World Scientific
www.worldscientific.com

PROBABILISTIC ADAPTIVE CROSSOVER (PAX): A NOVEL GENETIC ALGORITHM CROSSOVER METHODOLOGY

SEBASTIÁN A. SALAH
Electrical Engineering Department, University of Chile,
Av. Tupper 2007, Casilla 412-3, Santiago, Chile
ssalah@ultra.biz

MANUEL A. DUARTE-MERMOUD*
Electrical Engineering Department, University of Chile,
Av. Tupper 2007, Casilla 412-3, Santiago, Chile
mduartem@ing.uchile.cl

NICOLÁS H. BELTRÁN
Electrical Engineering Department, University of Chile,
Av. Tupper 2007, Casilla 412-3, Santiago, Chile
Nicolas.Beltran@die.uchile.cl

Received 8 October 2007
Accepted 19 May 2008

A new crossover technique for genetic algorithms is proposed in this paper. The technique is called probabilistic adaptive crossover and denoted by PAX. The method includes the estimation of the probability distribution of the population, in order to store in a unique probability vector P information about the best and the worse solutions of the problem to be solved. The proposed methodology is compared with six crossover techniques namely: one-point crossover, two-point crossover, SANUX, discrete crossover, uniform crossover and selective crossover. These methodologies were simulated and compared over five test problems described by ONEMAX Function, Royal Road Function, Random L-MaxSAT, Bohachevsky Function, and the Himmelblau Function.

Keywords: Genetic algorithms; adaptive crossover; crossover methodologies.

1. Introduction

Genetic Algorithms (GAs)[1] is one of the techniques known under the name of Evolutionary Algorithms (EAs), where also Evolutionary Strategies,[2] Evolutionary Programming[3] and Genetic Programming[4] belong. All of them are inspired by the concept of biological evolution. The main idea is that each individual of the population is represented by a binary strip called chromosome, where each element in the strip is called

*Author to whom all correspondence should be addressed.

gene or locus. Each individual in the population corresponds to a possible solution of the problem to be solved. The evolution of the population will take place by generating randomly new possible solutions corresponding to the result of a crossover among the different individuals selected from the population according to the fitness of each individual (measured in terms of an objective function). New solutions can also be obtained by introducing mutation in the individuals.

In 1975 Holland[1] proposed the technique called one-point crossover. This crossover methodology basically consists of choosing a point (gene) within the individuals selected as parents and generate new individuals (offspring) by interchanging the contents of the parents from both sides of the selected point.

In his doctoral thesis published in 1975, De Jong[5] generalized the one-point crossover technique to an N-points crossover technique, interchanging the N-1 segments of the parents located between the crossing points to create the new individuals. The scheme with N = 2 was the most successful because the two-points crossover showed a better performance than the one-point crossover and also was less destructive than those of higher order (which is interesting from the Schemata Theorem viewpoint proposed previously by Holland).

In 1987, Ackley[6] proposed the uniform crossover technique, usually attributed to Syswerda,[7] that in 1989 presented a theoretical study of uniform crossover. This methodology can be seen as a generalization of the N-points crossover technique, where N is not fixed. Although this methodology is highly destructive presents the advantage of having no positional bias, as occurs with the one and two-point crossover techniques.

Also, some techniques that differ from the traditional genetic conception have been proposed. A group of these techniques are crossover methodologies where the new individuals are generated from more than two parents, like the one proposed by Eiben.[8] An example of this is given by Ackley,[9] that proposed the Stochastic Iterated Genetic Hill-climbing (SIGH) using a voting method (where the voters are all the individuals of the population) to determine the values of the new binary strip. This methodology demonstrated to be more efficient in four of the six problems where it was proven, when compared with the one and two-point crossover and uniform crossover techniques.

Mühlenbein[10] proposed a methodology denominated Gene-Pool Recombination (GPR) that consists in selecting several possible parents for each individual, choosing with replacement two parents for each locus to generate the crossover using any traditional operator. It is shown an improvement of 25% in the results obtained on the ONEMAX[6] problem as compared with the traditional algorithms.

A different approach from those mentioned previously, that moves away from the GAs since does not use the concept of crossover, corresponds to the Estimation of Distribution Algorithm (EDA).[11] This basically considers the probability distribution of the population and from this estimation new individuals are generated. One of the first approaches was proposed by Syswerda[12] and called Bit Simulated Crossover (BSC), where the objective is to try to maintain the expected number of ones and zeros for each locus when applying uniform crossover.

There exist also works where the estimation of the distribution of a variable is performed taking into account some dependencies. This theory can be found in works like Bi-variate Marginal Distribution Algorithms (BMDA).[13]

Another type of crossover techniques can be grouped under the term adaptive crossover, where some kind of adaptation is applied when the crossover is made. This kind of techniques will be presented later in this paper.

The main idea of this paper is to introduce a new adaptive crossover technique called Probabilistic Adaptive Crossover (PAX), where the estimation of the distribution of the population is made in a new fashion rewarding the best individuals of the population and penalizing the worse individuals. Once the estimation of the distribution is obtained, two parents are selected for generating new individuals transferring to the offspring the allele of each locus when the parents are equal and applying the estimated distribution to determine the alleles where the parents are different. From simulations it is empirically explored the kind of problems where PAX presents advantages over the existing methods.

The paper is organized as follows. In Chapter 2 some works related to the proposed methodology are discussed. Chapter 3 is devoted to the description of the new crossover technique. A brief explanation of the test problems used in this study is presented in Chapter 4, whereas in Chapter 5 an evaluation of the proposed method is performed and compared with six other crossover techniques, applied to the five classical test problems of Chapter 4. A discussion of the results obtained, some conclusions of the study and the future work still to be done are presented in Chapters 6, 7 and 8, respectively.

2. Related Work

There exists a set of crossover operators depending on a criterion which is established from the results of previous crossover and from the performance obtained with the generated individuals. This set of operators differs from the static operators, where always the same crossover is applied.

According to Yang[14] the way in which the crossover is adapted can be classified into three categories:

(a) **Adaptation of the crossover type:** This is an adaptation that occurs at high level, where the crossover type (or the operator) to be used is selected. Davis[15] proposes a methodology in which the operator to be employed is chosen from a set of operators (crossover or mutation), which has a fixed probability associated. Later, Spears[16] proposed a method where each individual has an extra bit that indicates if it must be crossed using two-point crossover (in the case that the additional bit is 1) or using uniform crossover (in the case that the additional bit is 0). If two individuals that have the same value in the additional bit are selected as parents, the crossover associated with that value will be used and in case that the value is different for both individuals, one of the two crossover methodology will be used with probability 0.5.

Eshelman and Schaffer[17] proposed a mechanism that selects the best crossover operator between *Half Uniform Crossover (HUX)*, which is a variant of the uniform crossover where half of the different bits between parents are modified, and *Shuffle Crossover (SHX)*, which is a variant of one-point crossover without positional bias. This mechanism takes place when a population has converged to some maximum, then the population is reinitiated excepting the best individual and the crossover method is alternated.

(b) Adaptation of the crossover rate: A second alternative of adaptation is proposed to operate at a medium level, where the operators to be used are fixed, but the probabilities of using one of each operator is adapted. Corne, Roos and Fang[18] proposed *Cost Operating Based Rate Adaptation* (*COBRA*), a methodology that updates a vector that represents the probability of applying each operator. The updating is made proportional to the contribution made to the fitness function of the individuals. Later, Tuson and Ross[19] incorporate the probability of the operators (crossover or mutation) to each individual, which is evolving with them.

(c) Adaptation of the crossover position or the probability of change of each bit: This kind of adaptation is considered a low level adaptation, since acts directly on the bits. Shaffer and Morishima[20] proposed a method called *Punctuated Crossover*, in which a bit map is attached to each individual and indicates the positions where it is possible to apply crossover. This bit map co-evolves with each individual. Louis and Rawlins[21] proposed a *Masked Crossover*, methodology that adds an extra binary strip to each chromosome, which is used like mask. This mask is compared in both parents and the points which are different are defined as crossover points. Then, using information about the fitness and information about the parents, the mask is updated.

Later, White and Oppacher[22] proposed *Adaptive Uniform Crossover* (*AUX*), where each bit of each individual is increased including an automata that contains the crossover probability of each bit. This operator also uses information of the evaluation function to identify the bits that remain together (dependencies) when crossover is applied.

Vekaria and Clack[23] proposed *Selective Crossover*, where each chromosome is increased by a dominance vector that represents the contribution of that bit to the fitness of that individual. When the crossover is applied, the bits of larger dominance are chosen for one son and the remaining bits for the other, updating later the dominance vector. This updating is made by increasing the values of the dominance vector in those positions where the son differs from the father (comparing the nearest individuals) proportional to the increment in the adaptation function. Initially the values for the dominance vectors are generated in a random fashion.

Later, Yang[14] proposed *Statistics-Based Adaptive Non-Uniform Crossover* (*SANUX*). This method assigns a smaller crossover probability to a bit of a chromosome, if the frequency of its value is high within the population. To do this an additional vector $f_1(i, t)$ is used in each generation, corresponding to the frequency of 1's at the position *i* in generation *t*. Then, the probability of interchanging the value of the *i-th* position is computed according to

$$p_s(i,t) = \begin{cases} f_1(i,t) & \text{if} \quad f_1(i,t) \le 0.5 \\ 1 - f_1(i,t) & \text{if} \quad f_1(i,t) > 0.5 \end{cases} \tag{2.1}$$

Once the probability of interchanging the value of the *i-th* position is calculated, a mask from the probabilities vector is created and those positions where the value of the mask is 1 are interchanged.

2.1. *Estimation of Distribution Algorithms* (*EDAs*)

The *Estimation of Distribution Algorithms* (*EDAs*)[11] is another family of algorithms that are used in optimization problems. These are more recent than the GAs and they do not have an inspiration in the Mendelian theory as the GAs do. In spite of this, some of the elements of EDAs, like the existence of a codified population accordingly to the problem, the existence of a stopping condition and also the performance obtained when they are applied to certain problems, are similar to GA.

The essential difference between EDA and GA, is the way in which the individuals of the next generation are generated. The EDA estimates the distribution of the best individuals of a population and uses this estimation to generate new possible solutions. The simplest EDA considers only independent variables (or bits). There exists also another set of algorithms that consider relationships between variables, called *Bivariate Marginal Distribution Algorithm* (*BMDA*).[13]

Next, some of the algorithms that do not consider relationships between variables are presented.

2.2. *Bit-Based Simulated Crossover* (*BSC*)

Syswerda[12] proposed this new methodology, in which no explicit form of crossover between two parents appears. The population is treated like a conditional variable of a probability density function that predicts the probability of generating possible solutions in the space of the problem under study. The method considers all the population and after its application produces a new population of individuals that will have the same distribution of ones and zeros that one generated using uniform crossover.

This mechanism assumes that each locus in a population contains certain number of ones and zeros and that the individuals to whom these loci belong, have an assigned probability of being selected for crossover. Then, this probability can be used to compute a weighed average of ones and zeros that is expected to appear in each locus in the new population. If the probability obtained for each locus is used in the generation of loci for the new individuals of the next population, it will be obtained that in average the number of zeros and ones will be equal to that obtained using uniform crossover.

2.3. *Population-Based Incremental Learning* (*PBIL*)

Baluja and Caruana[24] proposed this methodology that is very similar to BSC with the following differences:

- The probability vector P does not consider the probability of the individuals to be used as parents of the next generation (as it is done in BSC). To compensate this fact, only the best M individuals are used to update the vector P.
- The probability vector P is not calculated at each generation in an independent way, but it is updated with a rate of learning $\delta\,(0 < \delta < 1)$. *PBIL* considers for each selected individual the following update rule

$$P = (1-\delta)\cdot P + \delta\cdot Selected\_Individual\,. \tag{2.2}$$

2.4. *Univariate Marginal Distribution Algorithms* (*UMDA*)

UMDA[11] also belongs to the EDA family and it can be described as a mixture of BSC and PBIL, since a probability distribution vector is estimated (like in BSC), using only some selected individuals according to some pre-specified criterion to build the probability vector (like in PBIL).

The probability distribution vector is defined as

$$p(X) = \prod_{i=0}^{n-1} p_i(x_i)\,. \tag{2.3}$$

where $p_i(x_i)$ corresponds to the probability of the i-locus of the chromosome, calculated as the frequency of ones in each locus of the selected individuals to estimate the probability distribution.

3. Probabilistic Adaptive Crossover (PAX)

3.1. *Motivation of the method*

There exist two main properties that are desirable in a search method, both of them are present in the GAs. These properties correspond to the exploration and the exploitation characteristics.

The exploration is the ability to explore new regions in the search space, whereas the exploitation corresponds to the fair use of the information about the search space to guide the search towards the regions that presents a larger potential. Roughly speaking we could say that in GAs the exploration is mainly done by mutation, since this mechanism introduces new solutions to the problem, whereas the exploitation is performed by crossover, since this mechanism preserves the common bits of the parents.[25] Nevertheless, both operators do exploration.

Using the information of the population distribution when applying crossover, has shown advantages in the exploitation of the best solutions. An example of this is SANUX where the information of the distribution is used to determine the points where crossover has to be done.

The main idea in the crossover methodology being proposed in this paper is to introduce a more effective combination between the exploration and exploitation characteristics, using the information of all the population to determine the value of the offspring. The information of the entire population will be used when applying crossover in those points in which the two parents are different. Thus, a faster propagation of the best solutions throughout the population will be obtained, allowing a larger exploration, since the number of possible different children would be greater than the case of using other crossover methodologies.

Using some of the original ideas of the genetics that establish the existence of natural selection to explain the way how individuals adapt to the environment, the aim is when two parents are chosen to perform crossover, those genes where the parents have the same value are directly transferred to the children. It is expected that by natural selection, the fact that both parents have the same genetic content in certain positions, implies that this value in those positions is inheritable to the children in a direct way, since it has produced good adaptation to their parents.

For those genes where the values of the parents are different, instead of inheriting the value of one's of the parents in each one of the offspring, what will be studied is which of the values of the parents produces a greater adaptation to the environment. This information is found from the experience of the entire population.

3.2. *Description and characteristics of the method*

The proposed crossover methodology uses an adaptive vector of probabilities P for the entire population, which is being constantly updated. Each element of this vector P represents the probability that a bit of an individual in the offspring takes the value 1 when both parents have different values at this point. This vector considers the information of the previous generations and is updated using the fitness of the individuals of the population. The vector updating is made considering as positive examples to follow, those individuals that present a fitness over the average of the population and as negative examples, those individuals that present fitness below the average.

When performing the updating process the information of the previous generations is introduced, weighed by a forgetting factor. This forgetting factor is introduced because the average of the fitness of the individuals theoretically increases, which is the reason why a solution that was on the average in generation t can be below the average in the generation $t + 1$. Thus, the probability vector is updated considering the history of the population, as well as the current population.

Another characteristic that is achieved using this updating method is that the contribution to the adaptive vector of probabilities of those individuals which are over the average is weighted in a different way. That is to say, each individual that is over the average does not contribute equally to update of the probability vector, but the contribution is made proportional to the fitness of each individual with respect to the others. The same concept is used for those individuals that are below the average, where those individuals of worse performance will be the most highly punished.

In case that no allele stands out over the rest in some loci, the vector of probabilities should converge to 0.5, which would transform the method into a discrete crossover.[26]

In a formal way the application of a forgetting factor α is written as

$$\tilde{P}(t+1) = (P(t) - 0.5 \cdot \vec{1}) \cdot (1-\alpha) + 0.5 \cdot \vec{1} \tag{3.1}$$

where $\vec{1}$ means a vector of length L, and all the components in it has the value 1 and $\tilde{P}$ is an intermediate probability vector.

After the application of the forgetting factor, the individuals of the population $Pobl_t$ are separated between the individuals that their fitness is over the mean and those that the fitness is below the mean. These two groups are denoted as two sub-sets $pobl_+$ and $pobl_-$, that are defined as

$$pobl_+ = \left\{ x_t \in Pobl_t \mid f(x_t) \geq \overline{f(t)} \right\}. \tag{3.2}$$

$$pobl_- = \left\{ x_t \in Pobl_t \mid f(x_t) < \overline{f(t)} \right\}. \tag{3.3}$$

Finally, the probability vector is updated as follows:

$$P(t+1) = \tilde{P}(t+1) + \frac{\gamma}{2} \left(\sum_{x_t \in pobl_+} \frac{\left(f(x_t) - \overline{f(t)}\right) \cdot \vec{x}_t + \left(f(x_t) - \overline{f(t)}\right) \cdot (\vec{x}_t - \vec{1})}{\sum_{x_t \in pobl_+} (f(x_t) - \overline{f(t)})} - \sum_{x_t \in pobl_-} \frac{\left(f(x_t) - \overline{f(t)}\right) \cdot \vec{x}_t + \left(f(x_t) - \overline{f(t)}\right) \cdot (\vec{x}_t - \vec{1})}{\sum_{x_t \in pobl_-} (f(x_t) - \overline{f(t)})} \right) \tag{3.4}$$

where γ corresponds to the learning rate, that determinates the maximum value that can be used to update one position of the vector P. This value is reached in the case where all the individuals in the set $pobl_+$ have the same value in one position and the individuals in the set $pobl_-$ have the opposite value in the same position (in the case of a binary alphabet). By construction each component of vector P lies in the interval [0 , 1].

Once the probabilities vector P is updated, the individuals that will be the parents of the next generation are selected according to the desired selection scheme (selection by tournament, ranking selection, SUS, etc.). Next, the chosen parents are grouped in pairs, to give rise to two new children, that will have the same value of the parents in those positions where the value of the parents coincide. The rest of the positions will be filled up with 1, with the probability indicated by the vector P. After this, it is possible to apply tournament between children and parents, elitism operators or mutation, according to the desired scheme.

3.3. *Application example*

Let us consider the population in generation t, composed by four different individuals of length $L = 4$, described by

$$\begin{aligned} x_{t\_1} &= [1 \quad 1 \quad 1 \quad 0] \\ x_{t\_2} &= [1 \quad 1 \quad 0 \quad 0] \\ x_{t\_3} &= [0 \quad 1 \quad 1 \quad 1] \\ x_{t\_4} &= [0 \quad 0 \quad 0 \quad 1] \end{aligned}$$

where the fitness of each individual is

$$\begin{aligned} f(x_{t\_1}) &= 0.8 \\ f(x_{t\_2}) &= 0.6 \\ f(x_{t\_3}) &= 0.4 \\ f(x_{t\_4}) &= 0.2 \end{aligned}$$

Let us consider a probability vector $P(t) = (0.5 \quad 0.7 \quad 0.4 \quad 0.6)$ a forgetting factor $\alpha = 0.3$ and a learning rate $\gamma = 0.2$. Replacing these values in Equation (3.1) we will get

$$\begin{aligned} \tilde{P}(t+1) = ((0.5 \quad 0.7 \quad 0.4 \quad 0.6) - (0.5 \quad 0.5 \quad 0.5 \quad 0.5)) \\ \cdot (1 - 0.3) + (0.5 \quad 0.5 \quad 0.5 \quad 0.5) \end{aligned}$$

$$\tilde{P}(t+1) = (0.5 \quad 0.64 \quad 0.43 \quad 0.57)$$

Evaluating Equation (3.4) we obtain

$$P(t+1)^T = \begin{pmatrix} 0.5 \\ 0.64 \\ 0.43 \\ 0.57 \end{pmatrix} + \frac{0.2}{2} \left(\begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \cdot \frac{0.3}{0.4} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \cdot \frac{0.1}{0.4} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \end{pmatrix} \cdot \frac{0.3}{0.4} + \begin{pmatrix} 0 \\ 0 \\ -1 \\ -1 \end{pmatrix} \cdot \frac{0.1}{0.4} \right.$$

$$\left. - \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \cdot \frac{0.1}{0.4} - \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \cdot \frac{0.3}{0.4} - \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot \frac{0.1}{0.4} - \begin{pmatrix} -1 \\ -1 \\ -1 \\ 0 \end{pmatrix} \cdot \frac{0.3}{0.4} \right)$$

$$P(t+1)^T = \begin{pmatrix} 0.5 \\ 0.64 \\ 0.43 \\ 0.57 \end{pmatrix} + \frac{0.2}{2} \begin{pmatrix} 2 \\ 1.5 \\ 1.5 \\ -2 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.64 \\ 0.43 \\ 0.57 \end{pmatrix} + \begin{pmatrix} 0.2 \\ 0.15 \\ 0.15 \\ -0.2 \end{pmatrix} = \begin{pmatrix} 0.7 \\ 0.79 \\ 0.58 \\ 0.37 \end{pmatrix}$$

This is the probability vector that will be used in the crossover. In Fig. 1 the crossover between the second and the third individual of the population is shown.

In Fig. 1 it can be observed that in the first loci both children acquire the value 1 from Father 1, because from the probability vector, the probability in that position indicates that one is better than zero. In the third and fourth loci it is seen that values one and zero are generated, one for each son. It is in this fashion that a greater exploitation and exploration is obtained.

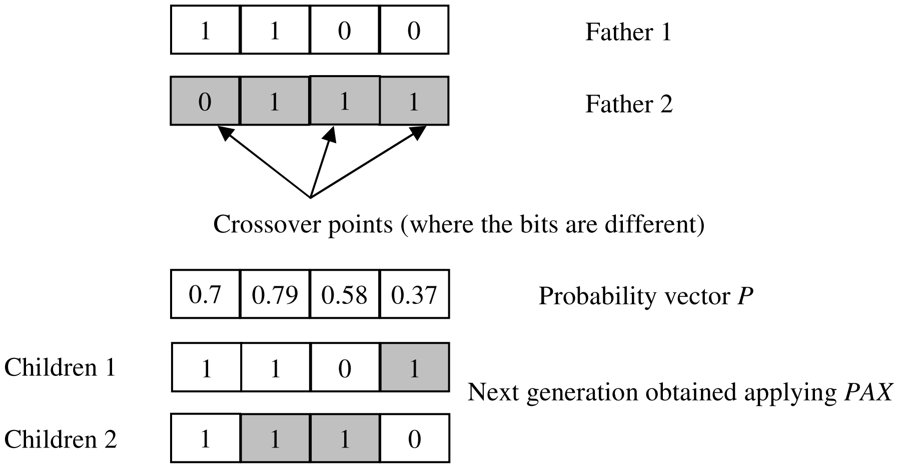

Fig. 1. Crossover example using PAX.

3.4. *Evaluation and comparison of the methodology*

In order to analyze empirically the performance of the proposed method, it is compared with different crossover methodologies. For comparison purposes, some classic methodologies are considered (one and two-point crossover, uniform crossover and discrete crossover) and also some adaptive methodologies are chosen (SANUX and Selective Crossover). All of them are studied using a GA with the following characteristics:

- **Selection methodology:** Fitness proportional selection with stochastic universal sampling (SUS).
- **Crossover probability:** $P_c = 0.7$.
- **Bit flip mutation probability:** $P_m = 0.01$

These parameters were chosen since they are used in several works done in this area like those of Vekaria[27] and Yang.[14] Several test problems were selected having different characteristics so that they allow us to draw conclusions on the *PAX* behavior. The number of individuals in each population and the length of these will depend on each problem. In the next section the different test problems used in this study are described.

4. Test Problems Used in the Study

4.1. *ONEMAX function*

The ONEMAX function[6] corresponds to a linear evaluation function (in terms of the Hamming distance), where each bit in each individual contributes to its fitness proportional to the contribution of other bits with value 1. Mathematically this evaluation function is defined as

$$fitness = \sum_{i=0}^{l} x_i \,. \tag{4.1}$$

This evaluation function is characterized as a simple linear optimization problem, with only one maximum, that does not present epitasis or deception. The optimal solution of the problem is well known and is defined by the length of the individuals that are used.

In order to run the different tests a population of one hundred individuals and a strip length of fifty were used. The stopping criterion corresponds at the moment in which the global maximum is reached, that is to say, the moment that the fitness of one of the individuals takes the value fifty. One hundred different initial populations were used, performing one hundred tests with each one of them, modifying for each test the random number generator seed.

4.2. *Royal Road function*

The Royal Road problem was proposed initially by Mitchell, Forrest and Holland[28]. To perform tests in GAs, particularly in one point crossover, the structure consists of small building blocks (defining-length 7) having a small contribution to the fitness function (Level 0). When joining two of these adjacent blocks (from left to right), they produce larger blocks (Level 1) and thus schemes of higher levels are obtained, until the solution of the problem is built, where the 64 bits composing each individual take value 1 and the fitness is 192 (which corresponds to the sum of the contributions of all the schemes of Levels 0, 1 and 2).

Later, using this same idea Vekaria and Clack[29] proposed modifications to this kind of problems. Those basically consist of changing the structure of the low level building blocks, increasing their defining-length.

This type of functions present a high epitasis degree, because the contribution of each bit to the fitness is conditioned to the values of the other bits defining the building block, since independently these bits do not contribute to the fitness.

The problem does not present deception, since when putting together two building blocks the result never gets worse.

In order to study the performance of the different crossover methodologies, the same parameters used by Mitchell, Forrest and Holland[28] were considered here. The number of individuals is N = 128 and their length is L = 64. Like in the case of ONEMAX, where the optimal global is known, the stopping criterion corresponds to the moment in which the fitness of some individual is 192. One hundred different initials populations were used, performing one hundred tests, modifying each time the random number generator seed.

4.3. *Random L-MaxSAT problems*

The problems of Boolean satisfaction correspond to problems satisfying constraints, consisting in finding values of the variables to satisfy a collection of clauses that are generally in a Conjunctive Normal Form (CNF). This problem corresponds to an NP-

Complete problem,[30] even in the cases when the number of variables by clause is 3 (the 3-SAT),[31] which means that it is not known if there exists a deterministic algorithm able to solve the problem in polynomial time.

The authors Mitchell, Selman and Levesque[32] proposed a generator of L-Sat expressions, that later was used by De Jong, Potter and Spears[33] to conduct studies on the behavior of the algorithms considering different epitasis levels (interaction between genes of a chromosome). This generator of expressions consists basically in forming C clauses of length L on a universe of V variables. Each clause is generated selecting L of the V variables and denying each one of them with probability 0.5, producing that each variable is present on average in $C\cdot L/V$ clauses. Because the Conjunctive Normal Form is used, it is enough that one of the L variables of each clause can be fulfilled in order to consider that the clause is satisfied.

While the number of clauses is increased, it is expected that each variable appears in a larger number of clauses, producing an increase in the epitasis of the problem, being more difficult to find combinations of 0's and 1's to satisfy the larger number of clauses.

The simplest codification for the representation of this problem is the use of binary strips of length V where each bit represents a variable. The fitness corresponding to each individual is

$$f(individual) = \frac{1}{C}\sum_{i=0}^{C} f(clause\_i)\,. \tag{4.2}$$

where $f(clause\_i)$ corresponds to the individual contribution of the clause i, having value 1 if the clause is satisfied or 0 if is not.

The parameters used to perform the tests are the same as those used by De Jong, Potter and Spears,[33] where the number of variables is V = 100, the length of the clauses is L = 3 and the number of clauses is varied from 300 clauses (for an experiment presenting low epitasis) to 1200 clauses (for a case with average level of epitasis) and up to 2400 clauses (for a problem with high level of epitasis). Because not always all the clauses can be satisfied, the existence of a global maximum equal to 1 can not be assured and therefore a stopping criterion is established when the number of generations is 600. A population of size N = 100 is considered. The results are obtained after the generation of 50 sets of different clauses, considering 10 different initial populations for each one of them.

4.4. *Bohachevsky function*

The inverted Bohachevsky function is shown in Fig. 2 which corresponds to the formula

$$f(x,y) = 3.6042 - (x^2 + 2\cdot y^2 - 0.3\cdot\cos(3\cdot\pi\cdot x) - 0.4\cdot\cos(4\cdot\pi\cdot y) + 0.7)\,. \tag{4.3}$$

This function has multiple local maxima and a unique global maximum located at $(0, 0)$, where $f(0, 0) = 3.6042$. This function will be used to verify premature convergence under different crossover schemes, using the niche methodology called deterministic crowding.[35]

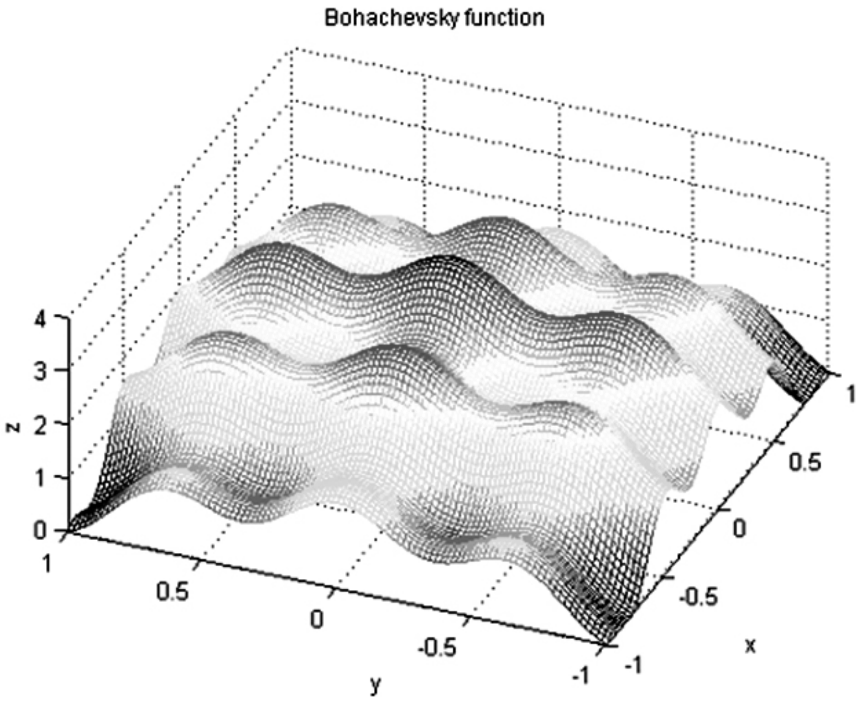

Fig. 2. Bohachevsky function.

17 bits were used to code each variable in the interval [–1, 1] and a population of 500 individuals was chosen. The results were registered after 200 generations and every 20 generations the number of individuals around the global maximum was verified. The results were obtained performing simulations over 200 different initial populations and for each initial population 5 simulations were performed. To appreciate the effect of the learning rate γ in the PAX scheme, two values of γ were considered; γ = 0, 2 and γ = 0.002.

4.5. *Himmelblau function*

The normalized inverted Himmelblau function is shown in Fig. 3. This function corresponds to one with multiple maxima of similar size (in some cases they are considered equals but they are not) located at the points indicated in Table 1.

This function is used to verify that the multiple maxima are reached and preserved. To do this deterministic crowding[35] is used, combining several crossover techniques. To appreciate the effect of the learning rate γ in PAX methodology, two values will be used; γ = 0, 2 and γ = 0.002.

A 15 bits coding is used for each variable restricted to the interval [–6, 6], and a population of 100 individuals was chosen. The results obtained after 200 generations are

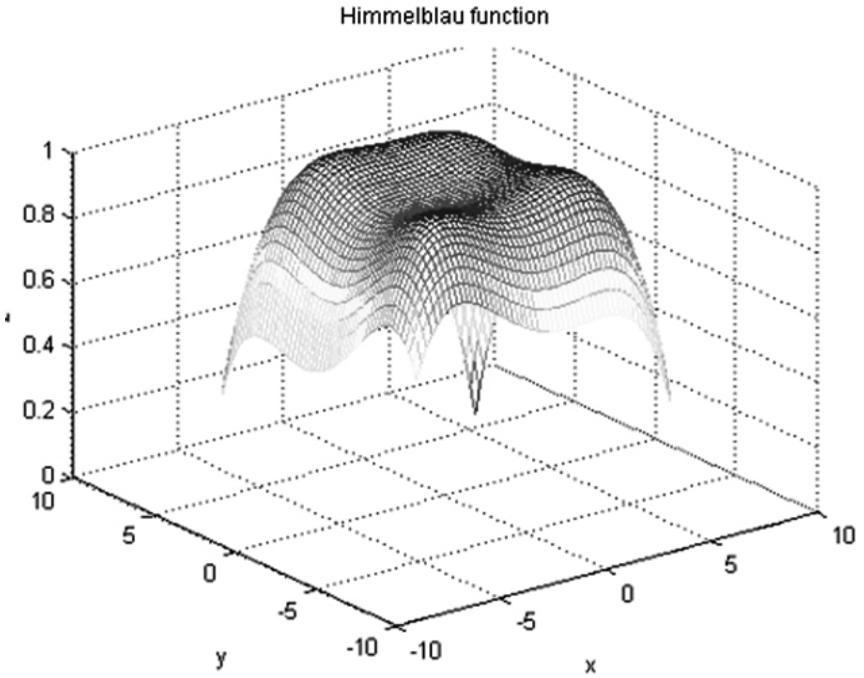

Fig. 3. Himmelblau function.

Table 1. Maxima of the Himmelblau function.

x	y	$f(x, y)$
3.0000	2.0000	1.00000000000000
3.5840	–1.8480	0.99999999566791
–3.7790	–3.2830	0.99999999748801
–2.8050	3.1310	0.99999999801805

registered and every 100 generations the number of individuals around each maximum is verified. The results were obtained performing simulations over 200 different initial populations and for each initial population 5 simulations were done.

5. Simulation Results

In this section the results obtained with PAX and six other crossover techniques are presented and compared. The methodologies used in the simulations are one-point crossover, two-point crossover, SANUX, discrete crossover, uniform crossover, selective crossover and they are compared with the proposed methodology probabilistic adaptive crossover (*PAX*). These methodologies were used over each of the five test problems described in Section 4.

The choice of the parameters used in simulations for each test problem described in Sections 4.1 to 4.5 were chosen either from other works (for comparison purposes) or after a series of previous experiments (to choose the more adequate).

5.1. *ONEMAX*

In Table 2 the average number of computations and the standard deviation needed to reach the maximum of ONEMAX is shown. The number of computations corresponds to the number of times that the function is called for evaluation. This average is considered over each different initial population and the different simulations performed on each initial population.

Figure 4 shows the curves corresponding to the average of the best individual as a function of the number of computations, for each one of the crossover techniques studied on the ONEMAX function.

Table 2. Average number of computations needed to reach the maximum of ONEMAX. In parenthesis the standard deviation is shown.

Crossover Methodology	PAX	Discrete	One Point	Two Points	Uniform	Selective Crossover	SANUX
Average number of computations	**907.0** **(134.8)**	6469.3 (1984.0)	6935.5 (2073.8)	6193.4 (1730.2)	7195.8 (2441.3)	12791.5 (51042.9)	5735.1 (1378.5)

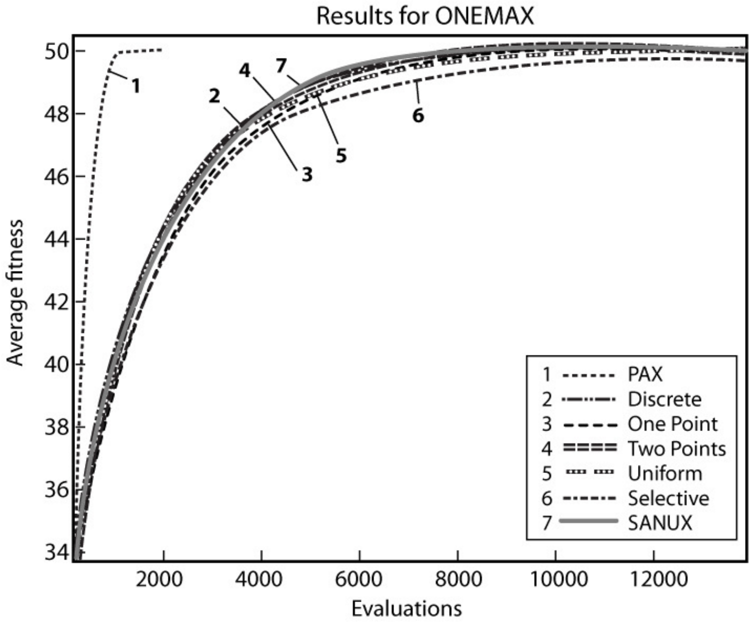

Fig. 4. Average fitness function of the best individual over ONEMAX.

From Fig. 4 is observed that all the crossover techniques compared reach the maximum of the ONEMAX function, having differences in the behavior of the methodologies only in the number of computations, i.e. in the speed of convergence to the optimum.

The average number of computations employed by the GA with PAX is much lesser than those used by the other crossover methodologies, corresponding to a 15.81% of those employed by SANUX, the second methodology as far as behavior is concerned.

5.2. *Royal Road function*

Table 3 shows the average number of computations used by each methodology over the different variants on the Royal Road function. In parenthesis the standard deviation is shown.

Table 3. Average number of computations needed to reach the maximum on the different codification of Royal Road function. In parenthesis the standard deviation is shown.

Crossover/Basis	RoyalRoad – A	RoyalRoad – B	RoyalRoad – C	RoyalRoad – D
PAX	26885.4 (13823.1)	26807.9 (13508.5)	26789.3 (13914.2)	26877.1 (13822.7)
Discrete	23564.2 (9694.8)	23462.0 (9863.7)	23240.3 (9826.0)	23302.0 (9832.2)
One-Point	27071.8 (14622.9)	28209.1 (15746.8)	29004.2 (14730.9)	31058.6 (13113.5)
Two-Points	23826.6 (12888.4)	24667.1 (13407.9)	26781.4 (12661.4)	28970.0 (12108.1)
Uniform	26991.6 (12155.8)	26633.2 (12173.3)	26469.1 (12196.7)	26452.1 (12341.2)
Selective	40381.8 (23520.9)	39407.7 (23291.3)	39104.7 (23033.6)	39318.3 (23269.5)
SANUX	**23227.2** **(9536.4)**	**23258.6** **(9802.9)**	**22973.0** **(9876.6)**	**23113.7** **(9617.9)**

From Table 3 is appreciated that the one-point and two-point crossover deteriorates their performance when the defining length is larger. The other crossover techniques behave similarly for all codifications. Also, it is observed that SANUX is the best methodology for the different codification.

The average curve of the best individual as a function of the number of computations for the database Royal Road A is presented in Fig. 5. In general, the curves obtained for the other codifications (Royal Road B, C and D) are quite similar for all the crossover techniques studied except for the one and two-point crossover techniques. For these techniques there is positional bias and their behavior is poor with respect to those

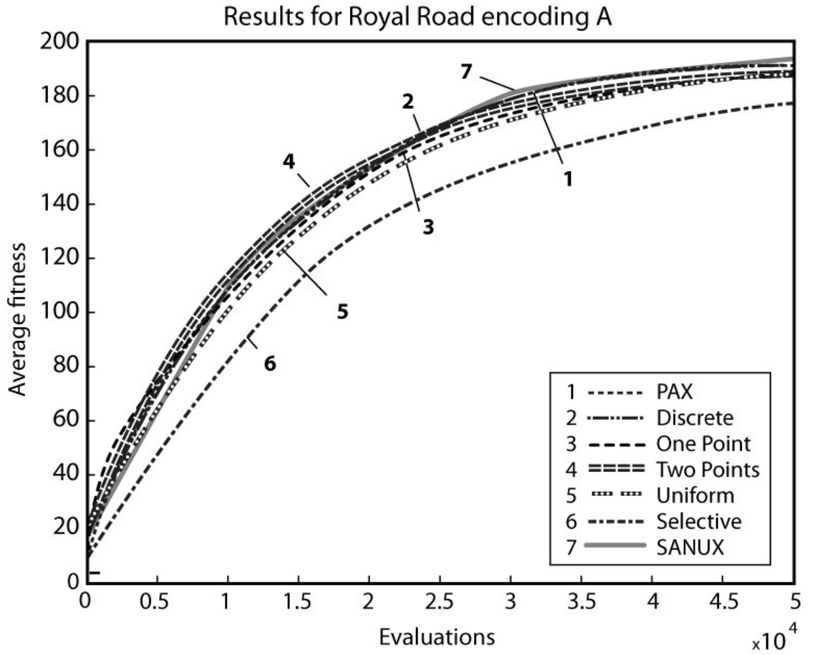

Fig. 5. Average fitness on Royal Road A.

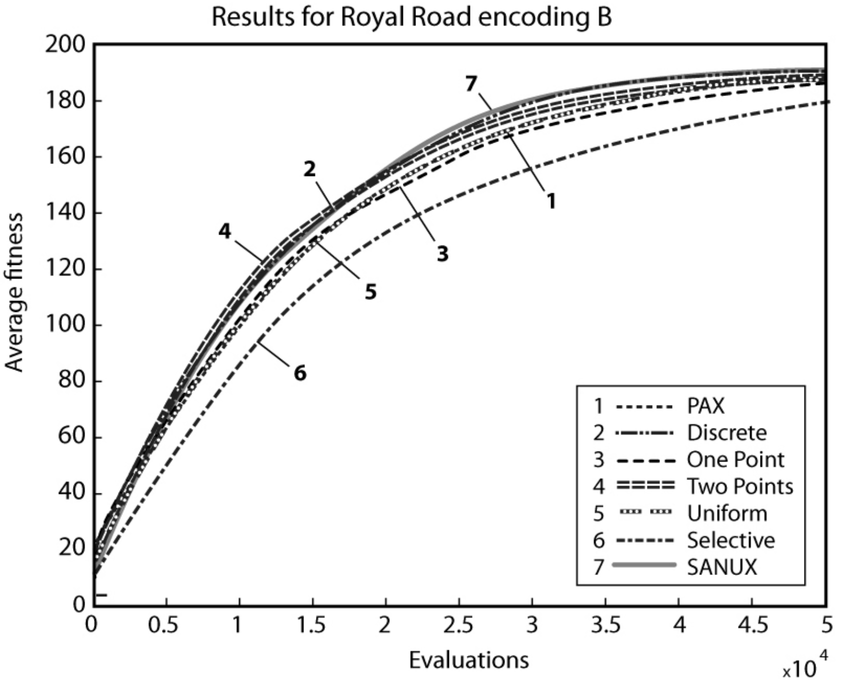

Fig. 6. Average fitness on Royal Road B.

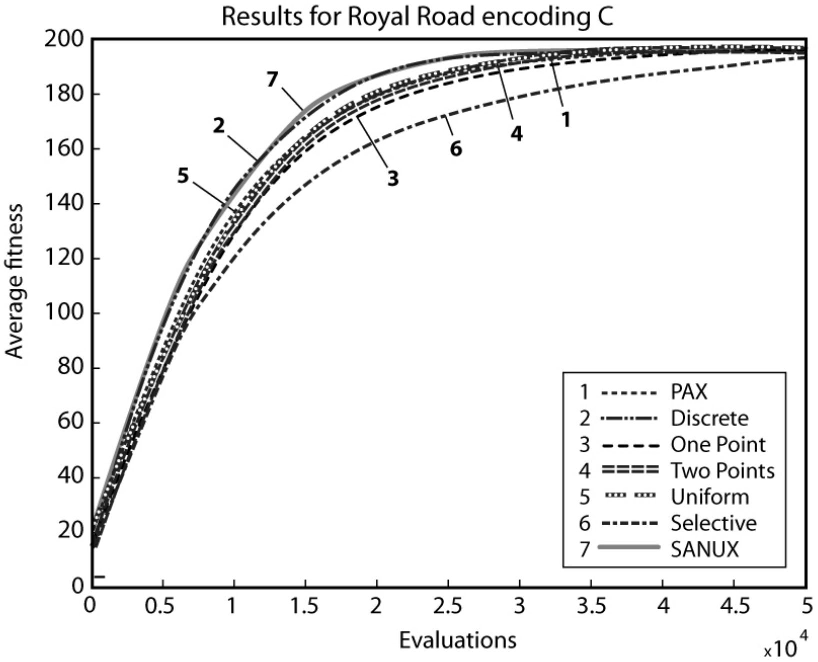

Fig. 7. Average fitness on Royal Road C.

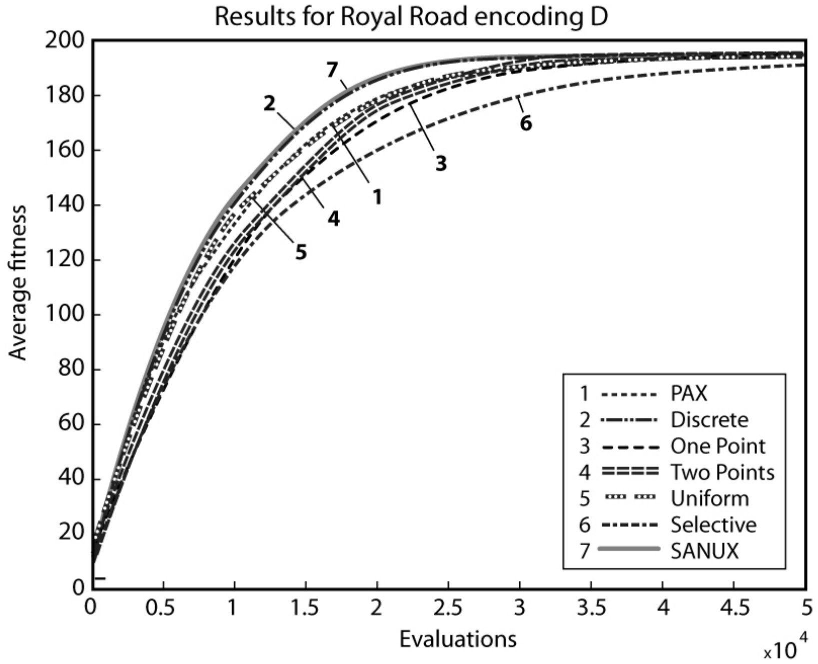

Fig. 8. Average fitness on Royal Road D.

codifications where the definition length of the building blocks is larger as compared with those in which the length is smaller. The evolution of the average of the best individuals for the codification Royals Road B, C and D are shown in Figs. 6, 7 and 8 respectively.

For this type of test database there is no individual contributions of each locus, but the contribution to the objective function is only considered when the complete scheme appears. Due to this, the probability vector used in PAX only receive useful information on specific loci when this is part of a scheme completely constituted on one individual. This is the reason why PAX uses approximately a 15% more computations than SANUX and discrete crossover in reaching the maximum. Compared with the uniform crossover, the number of computations is similar, fact that can be observed from Figs. 4 to 8.

It is important to point out that in the case of the selective crossover we were not able to reproduce the results reported in the thesis by Vekaria[27] and the performance was the worse of all the schemes studied.

5.3. *Random L-MaxSAT*

The average fitness reached by the best individual after 600 generations is shown in Table 4. This average is obtained over all the initial populations with different epitasis levels, using the testing function L-SAT. In parenthesis the standard deviation is shown.

Table 4. Average maximum reached by the best individual for database LSAT with different epitasis degrees. In parenthesis the standard deviation is shown.

Crossover/Basis	LSAT-200	LSAT-1200	LSAT-2400
PAX	1.0000 (0)	0.9558 (0.0028)	0.9347 (0.0022)
Discrete	0.9941 (0.0043)	0.9336 (0.0036)	0.9176 (0.0028)
One-point	0.9980 (0.0031)	0.9408 (0.0042)	0.9231 (0.0032)
Two-points	0.9981 (0.0031)	0.9411 (0.0039)	0.9235 (0.0031)
Uniform	0.9934 (0.0041)	0.9330 (0.0037)	0.9171 (0.0027)
Selective	0.9936 (0.0057)	0.9343 (0.0051)	0.9184 (0.0038)
SANUX	0.9944 (0.0043)	0.9337 (0.0036)	0.9178 (0.0028)

In Figs. 9 to 11 the average curve obtained for each epitasis degree are depicted.

For the three epitasis levels it is observed that the GA combined with PAX finds a solution that satisfies a larger number of clauses, which is reflected in the value reached by the adaptation function. This is particularly appreciated in cases of larger epitasis, where some clauses may not be feasible since they are contradictory to each other.

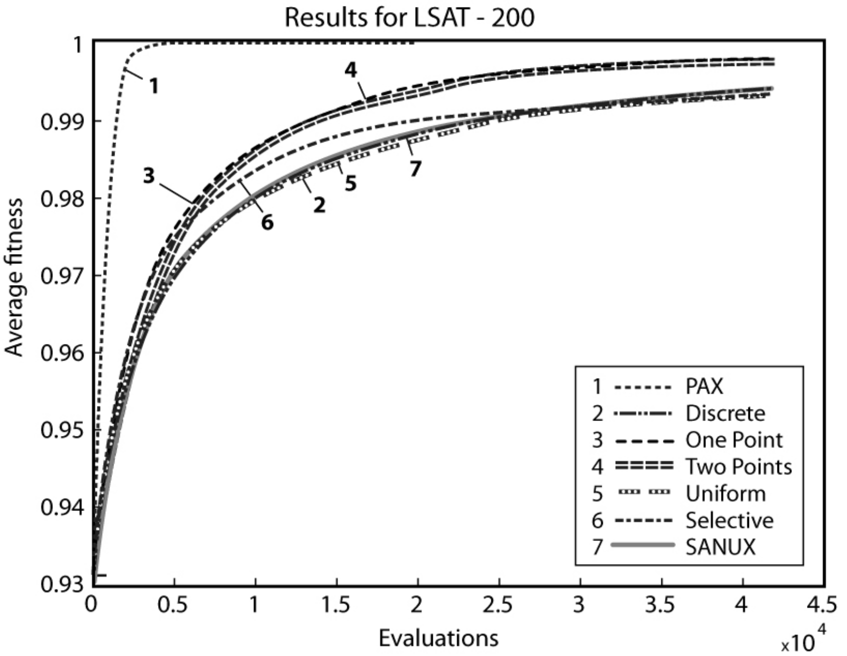

Fig. 9. Average fitness over the basis LSAT-200.

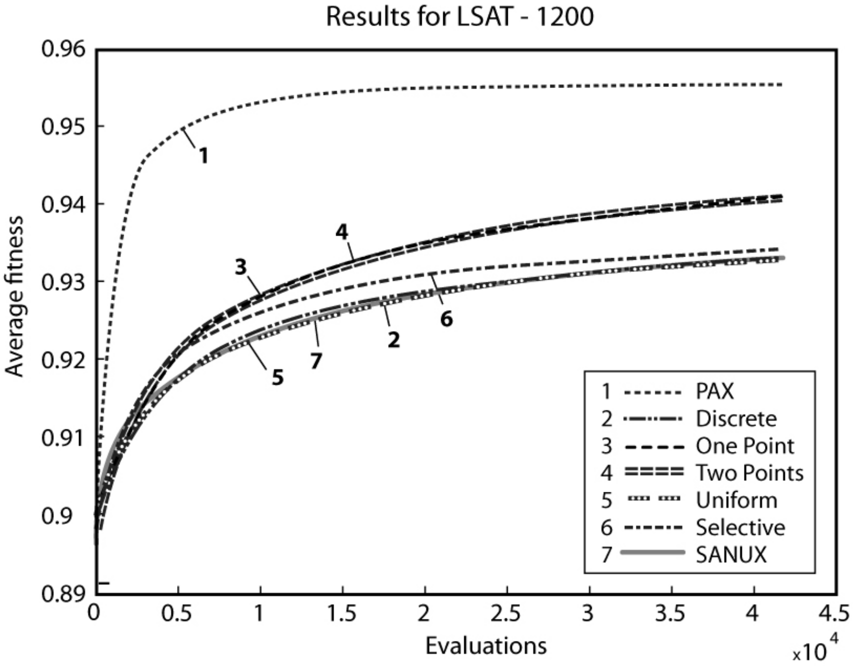

Fig. 10. Average fitness over the basis LSAT-1200.

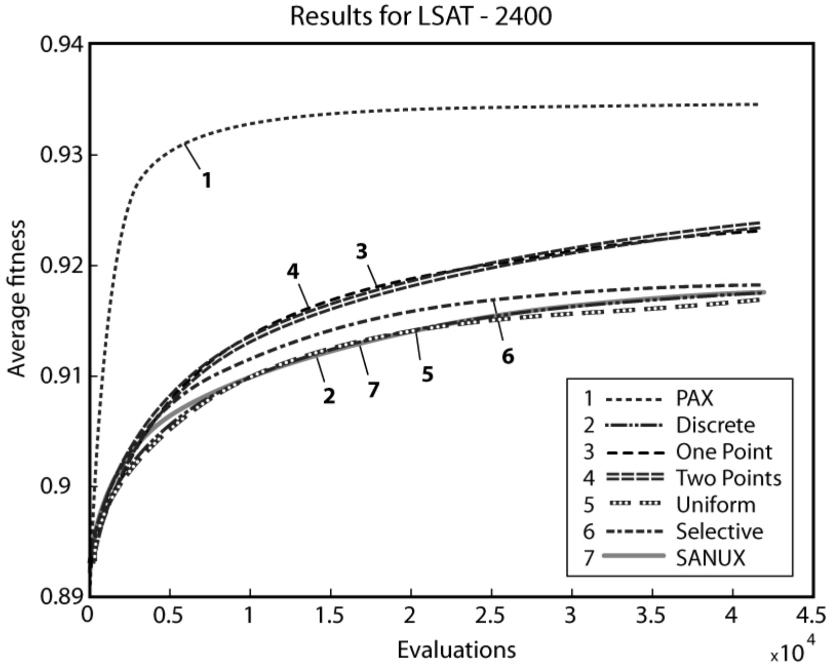

Fig. 11. Average fitness over the basis LSAT-2400.

Another interesting factor is the larger evolution speed reached by the GA using PAX as compared with the other crossover methodologies. For example, for the case of 3-SAT with an epitasis level of 300 clauses, the GA with PAX takes on average about 15% of the time taken by one and two-point crossover.

The good performance exhibited by the GA using PAX for high epitasis levels is explained as follows. Although PAX is performing an independent estimation for each locus through the probability vector, if appears an individual satisfying a larger number of clauses (which is reflected in its adaptation function) the values contained in each locus are distributed in a faster way to the rest of the population. This is due to the way in which the probability vector is updated, giving higher weights to those individuals presenting a better behavior.

5.4. *Himmelblau function*

In Table 5 the average number of individuals and its standard deviation around the four maxima, considering two different radii is shown. This information was extracted from generation 100 and 200.

Notice that the variance around each maximum using PAX with $\gamma = 0.2$ is larger than the rest of crossover methodologies. Since this is a problem with multiple maxima, the maximum found first by the algorithm will be surrounded by a larger number of

Table 5. Average number of individuals around the maxima of the Himmelblau function. In parenthesis the standard deviation is shown.

Maximum	Radius	Generation	PAX ($\gamma = 0.2$)	PAX ($\gamma = 0.002$)	Discrete	One-Point	Two Point	Uniform	SANUX
Point (3,2)	0.1414	100	65.50 (42.75)	39.41 (21.44)	29.26 (13.70)	16.03 (10.57)	28.92 (13.05)	35.52 (17.27)	1.67 (1.65)
		200	69.03 (43.67)	56.92 (26.33)	44.29 (14.38)	27.93 (11.44)	36.00 (11.10)	48.96 (18.18)	2.08 (2.13)
	0.0141	100	8.48 (23.25)	1.51 (5.48)	0.28 (0.92)	0.02 (0.21)	0.41 (1.04)	0.91 (3.92)	0.01 (0.08)
		200	25.89 (41.68)	23.81 (27.99)	6.85 (9.48)	1.04 (3.45)	10.41 (10.87)	17.58 (19.99)	0.01 (0.10)
Point (3.6,-1.8)	0.1414	100	11.02 (27.71)	16.62 (15.89)	12.94 (9.75)	17.38 (8.99)	15.38 (9.99)	15.33 (12.50)	1.33 (1.37)
		200	11.20 (29.14)	21.21 (20.37)	18.96 (10.95)	25.25 (10.39)	19.03 (9.76)	20.26 (13.65)	1.70 (1.78)
	0.0141	100	0.19 (3.20)	0.10 (0.70)	0.03 (0.18)	0.12 (0.55)	0.21 (0.94)	0.03 (0.22)	0.00 (0.03)
		200	1.18 (9.71)	2.52 (10.55)	0.36 (1.71)	2.42 (5.45)	3.63 (6.58)	1.18 (4.63)	0.00 (0.05)
Point (–3.7, –3.3)	0.1414	100	5.00 (18.59)	5.09 (6.18)	5.55 (4.84)	9.25 (6.43)	8.46 (5.66)	5.87 (6.05)	1.21 (1.22)
		200	5.30 (20.55)	5.00 (7.52)	8.28 (6.14)	13.25 (7.26)	12.23 (6.34)	7.63 (7.26)	1.50 (1.50)
	0.0141	100	0.27 (4.10)	0.01 (0.14)	0.00 (0.05)	0.02 (0.16)	0.03 (0.17)	0.01 (0.08)	0.00 (0.04)
		200	0.97 (9.25)	0.20 (2.12)	0.06 (0.32)	0.29 (1.18)	0.34 (1.10)	0.10 (0.65)	0.00 (0.04)
Point (–2.8, 3.1)	0.1414	100	12.25 (29.40)	12.42 (12.24)	10.38 (7.43)	13.33 (7.92)	21.08 (10.59)	11.96 (10.45)	1.71 (1.58)
		200	13.33 (32.02)	14.02 (16.61)	13.65 (8.51)	17.20 (8.24)	26.67 (10.90)	15.20 (12.42)	2.07 (1.91)
	0.0141	100	0.72 (7.41)	0.04 (0.35)	0.02 (0.17)	0.15 (0.73)	1.06 (2.86)	0.03 (0.36)	0.00 (0.03)
		200	2.60 (14.77)	1.14 (5.62)	0.25 (1.14)	1.95 (4.42)	9.16 (10.25)	1.02 (4.72)	0.00 (0.04)

individuals of the population, since the value of the variables will be diffused through the probability vector and therefore the variance of the number of individuals around the maximum is larger for PAX.

This larger diffusion of the maximum brings as a consequence that also PAX with $\gamma = 0.2$ has a little better speed of convergence. In Fig. 12 the average evolution of the individuals when applying the different crossover schemes over the Himmelblau function is shown.

It is surprising that SANUX, which showed a good behavior in the other test functions, exhibits a rather modest behavior this time.

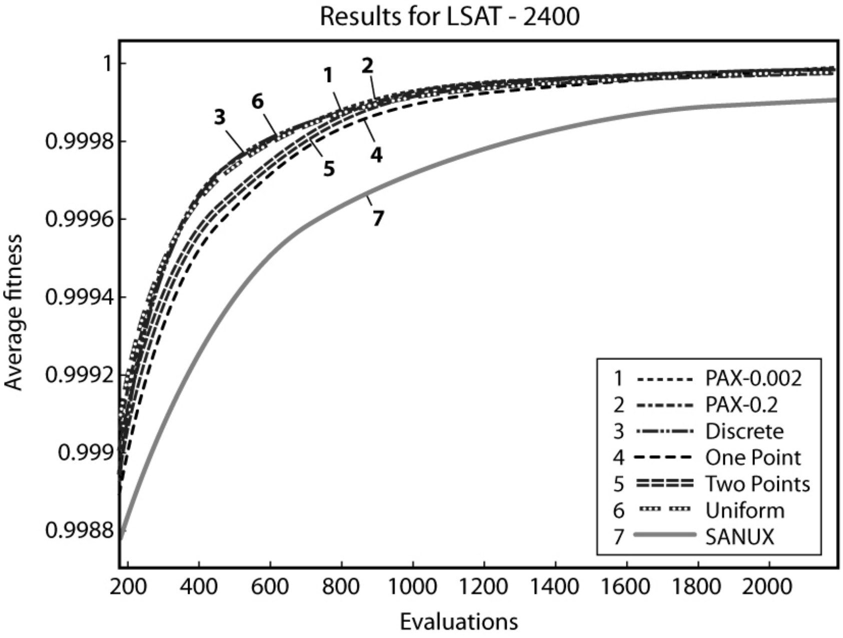

Fig. 12. Average adaptation function for Himmelblau problem.

5.5. *Bohachevsky function*

Next, in Fig. 13 it is shown the average curves obtained by using the best individual of each population for the crossover techniques under study, for the case of the Bohachevsky function.

In general the results are quite similar to those corresponding to the Himmelblau function, where there is a higher diffusion of the optimum in the population when PAX is used with $\gamma = 0.2$. All the schemes simulated reach the global maximum. The number of individual around the global optimum can be seen in Figs. 14 and 15 for two different values of the radius.

When analyzing the number of individuals around the optimum in PAX and considering a radius of 0.01414, the number of individuals approaching the optimum experiment a stagnation, which is attributed to a possible convergence (for the simulations using some initial population) to the optimum value around a radius of 0.01414. This behavior is not observed when a smaller learning rate $\gamma = 0.002$ is used. This can be seen as one of the possible disadvantages of PAX with respect to other methodologies. Premature convergence appears also when using a selection of individuals highly biased to the best individuals. The role of the parameters of PAX in the trade-off of exploration and exploitation will be determinant to avoid this kind of problems. The effect of the parameter modification can be seen through the previous example where different learning rates are used obtaining also different results.

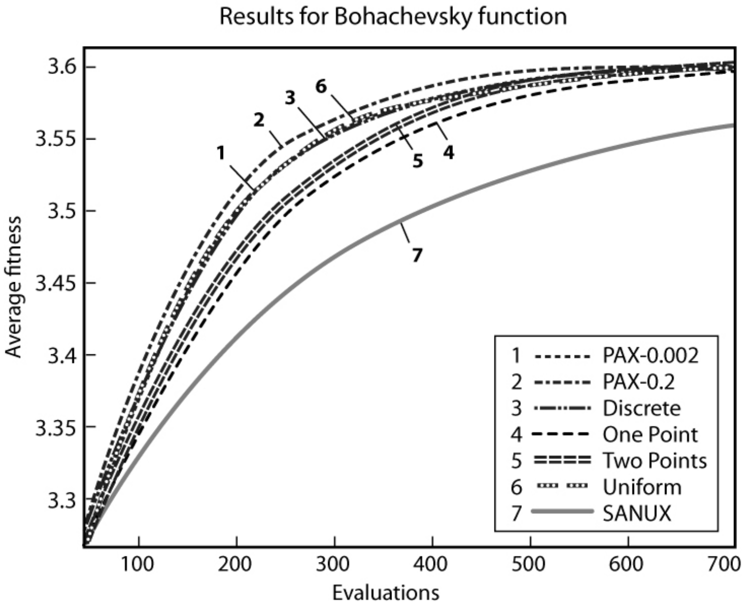

Fig. 13. Average fitness function using the Bohachevsky function.

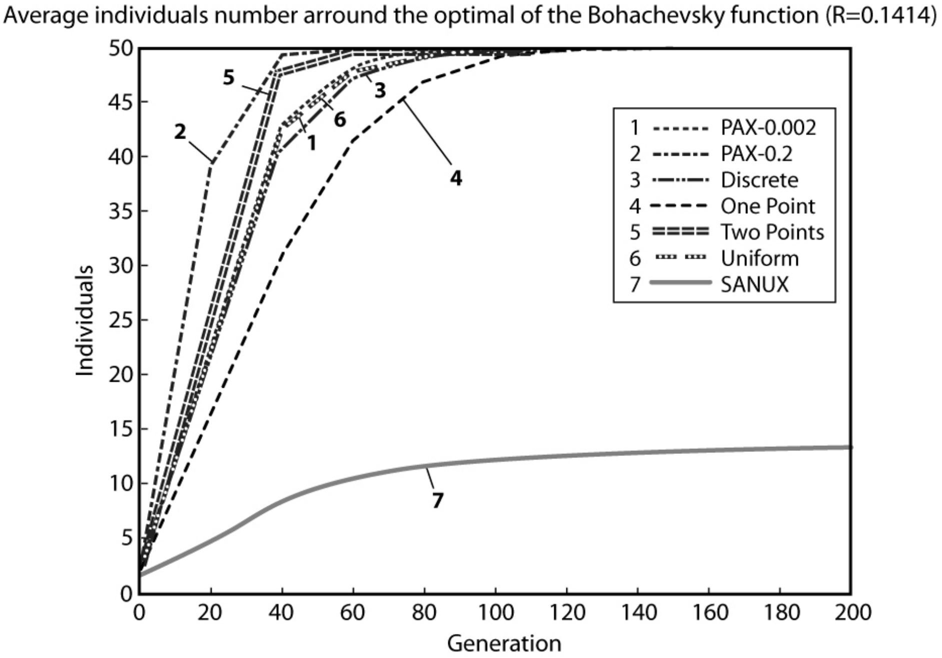

Fig. 14. Evolution of the average number of individuals around the optimum of the Bohachevsky function (Radius = 0.1414).

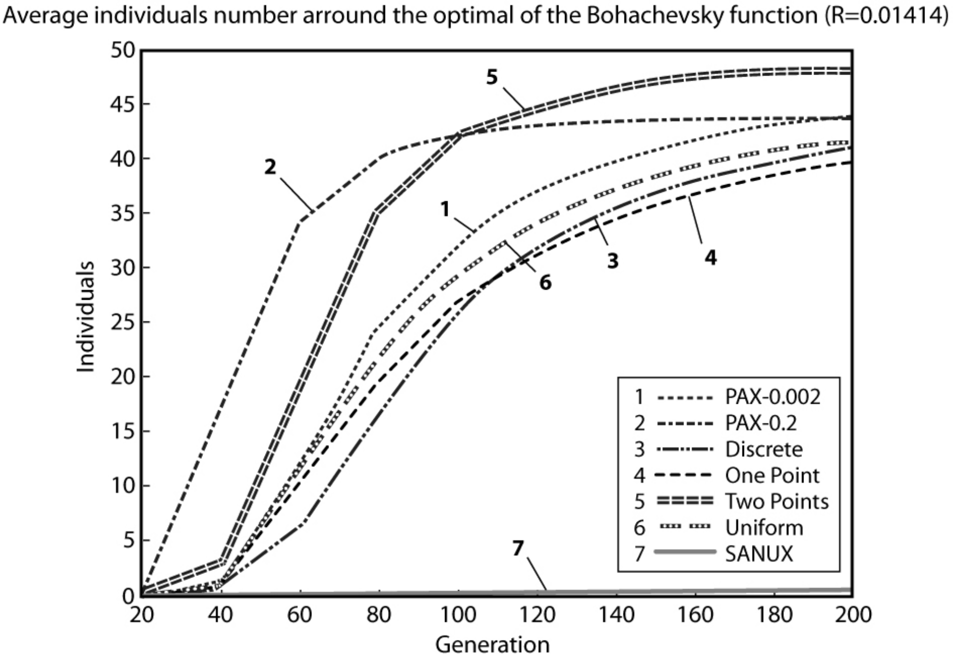

Fig. 15. Evolution of the average number of individuals around the optimum of the Bohachevsky function (Radius = 0.01414).

6. Analysis and Discussion of the Results

The simulations were carried out in 12 Pentium IV computers of 1.5 GHz and 256 MB of RAM memory running independently. The software used for the simulation was MATLAB 6.5 with the Toolbox Genetic Algorithm.[34] The different crossover methodologies studied do not present important differences as far as the processing time per generation is concerned. However the processing time is different depending on the function to be considered.

In the case of the ONEMAX problem, the approximate time for the evaluation of each individual is 75 [ms] giving a processing time per simulation (a complete run for an initial population) from 0.56 [s] for the case of PAX until 17.2 [s] for the selective crossover case.

For the case of Royal Road functions the evaluation is approximately 1.04 [s] giving processing times for one simulation ranging from 11.25 [s] to 47.03 [s].

In the case of L-SAT functions, the processing time depends on the epitasis level. With 200 clauses an average time of 6.82 [ms] is obtained for the evaluation of each individual, giving processing times of 5 minutes per simulation. It is important to notice that for each crossover method about 3,500 simulations were performed, taking about 752 hours to obtain a result (on one computer). In the case of 600 clauses the evaluation of each individual is slower taking about 19.45 [ms], which corresponds to 13 minutes per simulation, taking approximately 752 hours to obtain a result (on one computer). If the number of clauses is augmented to 1200, the time also increases requiring 38.77 [ms] per evaluation and 26 minutes per simulation (4 months of simulations on one computer). In order to reduce the computation time 12 computers were used in the study.

The time used to evaluate the Himmelblau function is about 3.03 [ms] and 6.6 seconds per simulation. In the case of the Bohachevsky function the average time is 15.12 [ms] per evaluation and 10.52 [s] per simulation.

In the optimization problems with a unique maximum and with individual contributions of each bit to the adaptation function, like the ONEMAX and L-SAT, it is observed that the performance of PAX is much better in the sense of the solution reached as well as in the computations needed to reach it. In the case of ONEMAX it is required only a 15.81% of the computations needed by SANUX, the methodology which is in second place.

For L-SAT problems with a low epitasis level, the utilization of PAX reveals that a lower number of computations to reach the maximum is necessary as compared to the other schemes. When using PAX and the epitasis level is augmented, besides reaching the results obtained by the other methodologies with a lower number of computations, the solution found by PAX is superior in the sense that a larger number of clauses is satisfied.

When there is no individual contribution to the adaptation function, the performance of the proposed methodology is slightly inferior with respect to the other methodologies used in the study and similar to that using uniform crossover. An example of this can be found with the different codification of the Royal Road problem, where it can be appreciated that PAX has no positional bias and the evolution of the GA with PAX is slightly inferior than the others using up to 15% more computations. This result is attributed to the non existence of individual contributions of each locus to the fitness function.

When using PAX in multimodal problems it is appreciated that there is not an evident overexploitation that avoids reaching results close to the maximum. It is important not to use too high learning rates since this can lead to premature convergence.

For the case of the Himmelblau function, the four maxima are reached. The variance of the number of individuals around each is higher for the case of PAX with a high learning rate. This is attributed to the fact that the optimum solution which is reached first is diffused in the population, although the other solutions are also reached and maintained.

For the Bohachevsky function, the results indicate that using PAX the global maximum is reached without premature convergence to some local maximum. Similar to the case of the Himmelblau function, it is observed that the learning rate determines the speed of convergence as well as the number of individuals around a maximum. With small learning rates, the speed of convergence is slower and the increment in the number of individuals around the maximum is also lower.

In conclusion, PAX presents significant advantages in the problems with a unique maximum and when each locus contributes individually to the objective function. In the multiple maxima problems or when there are no explicit contributions of each locus to the objective function, PAX has no significant advantages over the other methodologies and its performance is similar.

7. Conclusions

A new crossover methodology for GAs, denoted as PAX, has been proposed in this paper. The method include the estimation of the probability distribution of the population, in order to store in a unique probability vector P information about the best and the worse solutions of the problem to be optimized.

From the tests performed over artificial problems like L-SAT, ONEMAX, Royal Road, Himmelblau and Bohachevsky functions, as well as over a Chilean wine database composed of liquid chromatograms (not shown here)[36,37,38] it can be seen that using PAX in an optimization problem with only one global optimum where there exist individual contributions of each bit of the chromosome to the objective function, finds better solutions using a much lesser number of computations as compared with the other crossover schemes studied.

In problems where there are no explicit contributions of each bit of the chromosome to the objective function, the proposed methodology evolves in a similar fashion to the other schemes studied.

Although the proposed methodology does not correspond to those where the distribution is estimated using a jointly probability approach, PAX has good performance in problems with different epitasis degrees, since the dependence or epitasis is reflected in the probability vector.

When facing multimodal problems, PAX finds all the different optima even though in some cases a premature convergence scenario can be found. This premature convergence can be controlled through the different parameters of the method such as the learning rate, the bounding parameters and the forgetting factor.

Comparing PAX with other adaptive schemes such as SANUX and Selective Crossover, it presents a competitive behavior in one maximum problems. When applied to multimodal problems PAX retains in a better fashion the characteristics observed in one maximum problems whereas the other adaptive schemes strongly deteriorate their performances.

The introduction of new parameters in the proposed crossover methodology gives advantages and disadvantages. The advantages are that the evolution of the GA can be handled with these parameters. The disadvantages are that more parameters have to be adjusted which can generate very different results depending on the values chosen for the parameters.

Finally, it is suggested to use PAX in order to face one maximum optimization problems with GA, because the advantages can be considerable whereas the disadvantages are rather small. Even considering a situation of premature convergence, due to the high speed attained by the optimization process with PAX, this could be used as a first step within a scheme of approaching the optimum. Next we can proceed to a second phase that considers as initial population the solutions attained with PAX. This kind of idea has been used by Beligiannis *et al.*[39]

8. Future Research

From the results obtained in this work it is possible to visualize several future research lines. One interesting subject for further study is the inclusion of mutation using a probability vector updated in a similar fashion like the one proposed in PAX. This will allow having more control over the loci to mutate improving a possible scenario of premature convergence.

Also it is important to modify in the proposed methodology the way in which the probability vector is updated, passing from an independent probability approach to a one considering jointly probabilities, such as the one presented in the methodology BMDA.

It is also attractive to study the use of the probability vector considered in PAX in EDA schemes, where it is possible to consider a probability estimate which is close to the best individuals and it is far from the worst individuals.

Besides, we can think of developing an updating procedure for the probability vector for mutimodal problems, where more than one probability vector could be considered. These vectors could be updated with adaptation functions based on the closest individuals to be considered only when these individuals are used in the crossover.

Another modification that can be studied for the proposed scheme PAX, could consider the use of adjustable time-varying parameters, allowing for example a wider diffusion of the best solutions at the beginning of the execution and a lower diffusion of these from a specific generation and on (according to a decreasing function of the number of generations).

Acknowledgments

The results presented in this work were supported by CONYCIT-Chile, under the grant FONDEF D01-1016, “Chilean Red Wine Classification by means of Intelligent Instrumentation”.

References

1. Holland J.H., *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, Second Edition, 1992.
2. Schwefel. H.P., *Numerical Optimization of Computer Models*. Wiley, Chichester, UK, 1981.
3. Fogel L., Owens A.J. and Walsh, M.J,. *Artificial Intelligence through Simulated Evolution*. JohnWiley, New York, 1966.
4. Koza J., *Genetic Programming: On the Programming of Computers by means of Natural Selection*, MIT Press, Cambridge, Massachusetts, 1992.
5. De Jong K., *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, 1975.
6. Ackley D., *A Connectionist Machine for Genetic Hillclimbing*. The Kluwer international series in engineering and computer science. Kluwer Academic Publishers, Boston, Massachusetts, Vol. 28, 1987.

7. Syswerda G., "Uniform Crossover in Genetic Algorithms". *Proceedings of the Third International Conference on Genetic Algorithm*, Morgan Kaufmann Publishers, San Mateo, California, 1989, pp. 2–9.
8. Eiben A., "Multi-parent recombination", in *Handbook of Evolutionary Algorithms*. IOP Publishing Ltd. and Oxford University Press, 1997, pp. 25-33.
9. Ackley D., "An empirical study of bit vector function optimization". *Genetic Algorithms and Simulates Annealing*, Morgan Kaufmann, 1987, pp. 170-215.
10. Mühlenbein H. and Voigt H., "Gene pool recombination in genetic algorithms". *Proceedings of the Metaheuristics Conference*. Kluwer Academic Publishers, Norwell, 1995.
11. Mühlenbein H. and Paaß G., "From recombination of genes to the estimation of distributions", *Lecture Notes in Computer Science* 1141: Parallel Problem Solving from Nature, Springer: Berlin, 1996, pp.188-197.
12. Syswerda G., "Simulated Crossover in Genetic Algorithms", *Foundations of Genetic Algorithms 2*. Morgan Kaufmann., 1993, pp. 239-255.
13. Pelikan M. and Mühlenbein H., "The Bivariate Marginal Distribution Algorithm", *Advances in Soft Computing - Engineering Design and Manufacturing*, Springer-Verlag, London, 1999, pp. 521-535.
14. Yang S., "Adaptive non-uniform crossover based on statistics for genetic algorithms", *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO 2002*, New York, 9-13 July 2002, Morgan Kaufmann Publishers, San Francisco, CA, 2002, pp. 650-657.
15. Fukunaga K., *Introduction to Statistical Pattern Recognition*. Academic Press Inc, 1990.
16. Spears W., "Adapting Crossover in a Genetic Algorithm", *Technical Report AIC-92-025*, Washington, DC: Naval Research Laboratory, Navy Center for Applied Research on Artificial Intelligence, 1992.
17. Narendra P. and Fukunaga K., "A branch and bound algorithm for feature selection". *IEEE Transactions on Computers*, vol. 26, September 1977, pp. 917-922.
18. Corne D., Ross P. and Fang H., "Fast Practical Evolutionary Timetabling", *Evolutionary Computing: AISB Workshop 1994*, Springer Verlag, 1994.
19. Koller D. and Sahami M., "Toward optimal feature selection". *Proceedings of 13 International Conference on Machine Learning*, 1996, pp.284-292.
20. Schaffer J. and Morishima A., "An Adaptive Crossover Distribution Mechanism for Genetic Algorithms". *Proceedings of the Second International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, 1987, pp. 36-40.
21. Louis S. and Rawlins G., "Designer Genetic Algorithms: Genetic Algorithms in Structure Design". *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann, 1991, pp. 53-60.
22. White T. and Oppacher F., "Adaptive Crossover Using Automata". *Proceedings of the Third Conference Parallel Problem Solving from Nature Conference*, Springer-Verlag, 1994, pp. 229-238.
23. Vekaria K. and Clack C. "Selective Crossover in Genetic Algorithms". Genetic Programming 1998: *Proceedings of the Third Annual Conference*, Morgan Kaufmann, 1998, p. 609.
24. Baluja S. and Caruana R., "Removing the genetics from the standard genetic algorithm," Proceedings of Machine Learning 1995, *Twelfth International Conference on Machine Learning*, Morgan Kaufmann, 1995, pp. 38-46.
25. Spears W. M., *The Role of Mutation and Recombination in Evolutionary Algorithms*. Ph.D. Dissertation, George Mason University, Fairfax, Virginia. 1998.
26. Mühlenbein H. and Schlierkamp-Voosen D., "Predictive Models for the Breeder Genetic Algorithm: I. Continuous Parameter Optimization", *Evolutionary Computation*, Vol. 1, No. 1, Spring 1993, pp.25-49.
27. Vekaria K., *Selective Crossover as an Adaptive Strategy for Genetic Algorithms*, Ph.D. Thesis, Department of Computer Science, University of London, March 2000.
28. Mitchell M., Forrest S. and Holland J., "The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance". In *Toward a Practice of Autonomous Systems:*

Proceedings of the First European Conference on Artificial Life. Cambridge, MA: MIT Press, 1991.

29. Vekaria K. and Clack C., “Royal Road Encodings and Schema Propagation in Selective Crossover”. *Soft Computing in Industrial Applications*, Springer-Verlag. 2000, pp. 281-292.
30. Cook S., “The complexity of theorem proving procedures”. *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing*, 1971, pp. 151-158.
31. Garey M. and Johnson D., *Computers and Intractability A Guide to the Theory of NP-Completeness*. Freeman and Company, 1979.
32. Mitchell D., Selman B. and Levesque H., “Hard and Easy Distributions of SAT Problems”. In *Proceedings of the Tenth National Conference on Artificial Intelligence*. MIT Press. 1992, pp. 459-465.
33. De Jong K., Potter M. and Spears W., “Using Problem Generators to Explore the Effects of Epitasis”. *Proceedings of the Seventh International Conference on Genetic Algorithms*. Morgan Kaufmann. 1997, pp. 338-345.
34. Chipperfield A. and Fleming P., “The MATLAB Genetic Algorithm Toolbox”. *IEEE Colloquium on Applied Control Techniques Using MATLAB*, Digest No. 1995/014, 26 Jan, 1995.
35. Mahfoud S.W., *Niching Methods for Genetic Algorithms*. Ph.D. Thesis, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory (IlliGAL) Report No. 95001, May 1995.
36. Beltrán N.H., Duarte-Mermoud M.A., Bustos M.A., Salah S.A., Loyola E.A., Peña-Neira A.I., and Jalocha J.W., “Feature extraction and classification of Chilean wines”. *Journal of Food Engineering*, vol. 75, No. 1, 2006, pp. 1-10.
37. Beltrán N.H., Duarte-Mermoud M.A., Salah S.A., Bustos M.A., Peña-Neira A.I., Loyola E.A., and Jalocha J.W., "Feature selection algorithms using Chilean wine chromatograms as examples". *Journal of Food Engineering*, vol. 67, No. 4, 2005, pp. 483-490.
38. Duarte-Mermoud M.A., Beltrán N.H. and Salah, S.A., “Probabilistic adaptive crossover (PAX) applied to Chilean wine classification”, May 2008 (Submitted to *Engineering Applications of Artificial Intelligence*).
39. Beligiannis G.N., Tsirogiannis G.A., and Pintelas P.E., “Restartings: A technique to improve classic genetic algorithms’ performance”, *International Journal of Computational Intelligence*, vol. 1, No. 2, 2005, pp. 102-105.