J Heuristics (2007) 13: 265–314
DOI 10.1007/s10732-007-9018-2

Improving crossover operator for real-coded genetic algorithms using virtual parents

Domingo Ortiz-Boyer · César Hervás-Martínez · Nicolás García-Pedrajas

Received: 4 August 2005 / Revised: 20 July 2006 /
Accepted: 2 August 2006 / Published online: 25 April 2007

Abstract The crossover operator is the most innovative and relevant operator in real-coded genetic algorithms. In this work we propose a new strategy to improve the performance of this operator by the creation of virtual parents obtained from the population parameters of localisation and dispersion of the best individuals. The idea consists of mating these virtual parents with individuals of the population. In this way, the offspring are created in the most promising regions. This strategy has been incorporated into several crossover operators. After analysing the results we can conclude that this strategy significantly improves the performance of the algorithm in most problems analysed.

Keywords Real-coded genetic algorithms · Crossover operator · Optimisation methods

1 Introduction

Genetic algorithms are multiple, iterative, stochastic, general purpose searching algorithms based on natural evolution (Goldberg 1989; Holland 1975). They maintain a population of individuals (sometimes called chromosomes) made up of series of variables (or genes) that represent a possible solution to the given problem. A *fitness* function must be defined that measures the capability of each individual in solving the problem. The algorithm proceeds iteratively (each iteration is called a generation)

D. Ortiz-Boyer (✉) · C. Hervás-Martínez · N. García-Pedrajas
Department of Computing and Numerical Analysis, University of Córdoba, Córdoba, Spain
e-mail: dortiz@uco.es

C. Hervás-Martínez
e-mail: chervas@uco.es

N. García-Pedrajas
e-mail: npedrajas@uco.es

selecting the best individuals according to the fitness functions which are subject to the operations of crossover and mutation in order to obtain better solutions. If the GA is well designed it should converge to a reasonably good solution.

Although in the initial formulation of GAs the solutions were codified using a binary alphabet, their properties are not subject to the use of binary strings (Antonisse 1989; Radcliffe 1992). Binary representation, however, can be problematic in tasks that require a high numerical precision, as it can limit the window through which the algorithm sees the real problem (Koza 1992). This is the reason that problem specific GA operators for other representations have been developed. One of the most used representations is real coding, whose power is widely justified in several theoretical studies, e.g.: (Wright 1991; Goldberg 1991; Radcliffe 1991; Eshelman and Schaffer 1993). This is the most natural codification in continuous domain problems, where each gene represents a variable of the problem. A GA using this codification is usually termed *Real-Coded Genetic Algorithm* (RCGA).

RCGAs have shown their ability to solve a wide variety of real-world problems. Among others, they have been applied to parameter estimation (Ortiz-Boyer et al. 2003), neural networks (Bebis et al. 1997; García-Pedrajas et al. 2005a, 2005b), aerospace design (Périauz et al. 1995; Hajela 2002), biotechnology (Roubos et al. 1999), economic (McNeils 2001) and constrained parameter optimisation problems (Michalewicz 1992; Ortiz-Boyer et al. 2002).

In RCGAs the selection process and crossover and mutation operators establish a balance between the exploration and exploitation of the search space. The selection process drives the search towards the promising regions. The mutation operator increases the diversity of the population, lost during the selection phase, by means of the random mutation of one or more genes of the individual. It is an exploration operator which aims at avoiding premature convergence to suboptimal solutions. In fact, this operator implements a random search and inherits the features of such a search (Bäck 1996).

Crossover is the most innovative operator. It is a method for sharing information between individuals that combines the features of two or more individuals, the parents, to create potentially better offspring. The underlying idea is that the exchanging of genetic material among good individuals is bound to generate even better individuals. Crossover operator exploits the available information from the population. It is considered the primary search operator in a GA (Holland 1975; De Jong and Spears 1992). Most works focused on improving the performance of GAs are devoted to this operator (Liepins and Vose 1992; Kita 2001; Beyer and Deb 2001; Herrera et al. 2003; Hervás-Martínez and Ortiz-Boyer 2005).

Numerous crossover operators have been developed for RCGAs. The first attempts implemented an exploitative search, or depth search, as they generated offspring only in the region bounded by the parents. Among others we can mention simple crossover (Goldberg 1989), two-point crossover (Eshelman et al. 1989), uniform crossover (Syswerda 1989), flat crossover (Radcliffe 1991) and arithmetical crossover (Michalewicz 1992). This exploitative search can lead to a diminishing diversity of the population and, thus, a premature convergence. In order to avoid this circumstance, the operators BLX, Fuzzy crossover (Voigt et al. 1995) and Simulated Binary Crossover (SBX) (Deb and Agrawal 1995) have been proposed. They generate offspring in the exploration region near the parents, and not only within the region bounded by them.

These operators carry out a sampling around the region where the parents are placed, but do not take into account the fitness of the parents. As our objective is generating offspring better than its parents an alternative approach would be to generate offspring closer to the best parent as in the case of Linear BGA crossover (Schlierkamp-Voosen 1994). Nevertheless, these approaches may favour a quick convergence towards local optima. This problem might be overcome if we use, instead of just two parents, the information given by the population features of localisation and dispersion of a specially suitable subset of the population.

Another alternative is the use of an offspring selection mechanism. One of the most widely used selects the best offspring to form the next population (Wright 1991). Aiming at avoiding premature convergence, in (Affenzeller and Wagner 2003) an offspring selection mechanism is proposed where a part of the population is made by the offspring that have better fitness than their parents, and the other part by offspring that have a fitness worse than their parents. Based on this idea, SASEGASA (Self Adaptive SEgregative Genetic Algorithm with Simulated Annealing aspects) (Affenzeller and Wagner 2004) increases the broadness of the search process using different subpopulations and crossover operators and joins the population after local premature convergence in order to end up with a population including all genetic information sufficient for locating a global optimum.

Our approach is based on using the features of localisation and dispersion of a specially suitable subset of the population to construct three virtual parents that will be used in the crossover process. The localisation estimator of the values of the genes of the best n individuals of the population forms the first virtual individual. The dispersion estimator of these n individuals is used to obtain a confidence interval that with probability $(1-\alpha)$ contains the true value of the localisation estimator. The bounds of this confidence interval form the other two virtual parents.

In order to encourage the search within the most promising search regions, we propose the use of any crossover operator to mate these virtual individuals with the individuals of the population. This mating favours the generation of offspring that has higher probability of obtaining better descendants.

The same underlying theoretical principles have previously been used in the definition of two crossover operators based on confidence intervals: CIXL2 (Ortiz-Boyer et al. 2005) and CIXL1 (Hervás-Martínez et al. 2003). However, the use of virtual individuals that reflect the features of the best individuals of the population can be advantageously incorporated to any kind of crossover. So, in this work we present the application of this methodology to the most widely used crossover operators, and study its influence on the performance of the crossover.

The remainder of the paper is organised as follows: Sect. 2 states the theoretical basis of the construction of virtual individuals using confidence intervals; Sect. 3 explains the basis of the extension of crossover operators using virtual parents and proposes versions of the most common crossovers; Sect. 4 studies the effect on the behaviour of the operator of using virtual parents; Sect. 5 describes the experimental setup; Sect. 6 shows a statistical analysis of the results of the experiments; and Sect. 7 states the conclusion of our paper.

2 Construction of confidence intervals

For the definition of the virtual individuals that identify the most promising search regions we need to estimate the parameters of localisation and dispersion of the distributions associated with the best individuals. In this section we explain how to construct a confidence interval where, with a probability $(1-\alpha)$, the true value of the localisation parameter belongs.

Both the estimator of the localisation parameters and the confidence interval can be obtained using an L_2 norm, if we assume a normal distribution, or using an L_1 norm if we assume an unknown distribution.

2.1 Estimators based on L_2 norm

Let $\boldsymbol{\beta}$ be the set of N individuals with p genes that make up the population and $\boldsymbol{\beta^*} \subset \boldsymbol{\beta}$ the set of the best n individuals. If we assume that the genes β_i^* of the individuals belonging to $\boldsymbol{\beta^*}$ are independent random variables with a continuous distribution $H(\beta_i^*)$ with a localisation parameter $\mu_{\beta_i^*}$, we can define the model

$$\beta_i^* = \mu_{\beta_i^*} + e_i, \quad \text{for } i = 1, \ldots, p, \tag{1}$$

e_i being a random variable. If we suppose that, for each gene i, the best n individuals form a random sample $\{\beta_{i1}^*, \beta_{i2}^*, \ldots, \beta_{in}^*\}$ of the distribution of β_i^*, then the model takes the form

$$\beta_{ij}^* = \mu_{\beta_i^*} + e_{ij}, \quad \text{for } i = 1, \ldots, p \text{ and } j = 1, \ldots, n. \tag{2}$$

If we consider the L_2 norm defined by

$$\|\beta_i^*\|_2 = \sum_{j=1}^{n} (\beta_{ij}^*)^2, \tag{3}$$

and we use the estimator of $\mu_{\beta_i^*}$ associated with the steepest gradient descent method, that is

$$S_2(\mu_{\beta_i^*}) = -\frac{\partial D_2(\mu_{\beta_i^*})}{\partial \mu_{\beta_i^*}}, \tag{4}$$

where the dispersion function induced by the L_2 norm is

$$D_2(\mu_{\beta_i^*}) = \sum_{j=1}^{n} (\beta_{ij}^* - \mu_{\beta_i^*})^2, \tag{5}$$

from (4) we obtain

$$S_2(\mu_{\beta_i^*}) = 2\sum_{j=1}^{n} (\beta_{ij}^* - \mu_{\beta_i^*}), \tag{6}$$

and making (6) equal to 0 yields

$$\hat{\mu}_{\beta_i^*} = \frac{\sum_{j=1}^{n} \beta_{ij}^*}{n} = \bar{\beta}_i^*. \tag{7}$$

Hence, the estimator of the negative gradient of the localisation parameter by means of the L_2 norm is the mean of the distribution of β_i^* (Kendall and Stuart 1977), that is, $\hat{\mu}_{\beta_i^*} = \bar{\beta}_i^*$.

The sample mean estimator is a linear estimator,[1] so it has the properties of unbiasedness[2] and consistency,[3] and it follows a normal distribution $N(\mu_{\beta_i^*}, \sigma^2_{\beta_i^*}/n)$ when the distribution of the genes $H(\beta_i^*)$ is normal. Under this hypothesis, we construct a bilateral confidence interval for the localisation of the genes of the best n individuals, using the studentization method, the mean as the localisation parameter, and the standard deviation $S_{\beta_i^*}$ as the dispersion parameter:

$$I^{\text{CI}} = \left[\bar{\beta}_i^* - t_{n-1,\alpha/2} \frac{S_{\beta_i^*}}{\sqrt{n}}; \bar{\beta}_i^* + t_{n-1,\alpha/2} \frac{S_{\beta_i^*}}{\sqrt{n}} \right], \tag{8}$$

where $t_{n-1,\alpha/2}$ is the value of Student's t distribution with $n-1$ degrees of freedom, and $1-\alpha$ is the confidence coefficient, that is, the probability that the interval contains the true value of the population's mean.

From this definition of the confidence interval, we define three intervals to create three "virtual" parents, formed by the lower limits of the confidence interval of each gene, CILL,[4] the upper limits, CIUL,[5] and the means CIM.[6] These parents have the statistical information of the localisation features and dispersion of the best individuals of the population, that is, the genetic information the fittest individuals share. Their definition is:

$$\begin{aligned} \text{CILL} &= (\text{CILL}_1, \ldots, \text{CILL}_i, \ldots, \text{CILL}_p), \\ \text{CIUL} &= (\text{CIUL}_1, \ldots, \text{CIUL}_i, \ldots, \text{CIUL}_p), \\ \text{CIM} &= (\text{CIM}_1, \ldots, \text{CIM}_i, \ldots, \text{CIM}_p), \end{aligned} \tag{9}$$

where

$$\text{CILL}_i = \bar{\beta}_i^* - t_{n-1,\alpha/2} \frac{S_{\beta_i^*}}{\sqrt{n}},$$

[1] It is a linear combination of the sample values.

[2] An estimator $\hat{\theta}$ is an unbiased estimator of θ if the expected value of the estimator is the parameter to be estimate: $E[\hat{\theta}] = \theta$.

[3] A consistent estimator is an estimator that converges in probability to the quantity being estimated as the sample size grows.

[4] Confidence Interval Lower Limit.

[5] Confidence Interval Upper Limit.

[6] Confidence Interval Mean.

$$\mathrm{CIUL}_i = \bar{\beta}_i^* + t_{n-1,\alpha/2}\frac{S_{\beta_i^*}}{\sqrt{n}}, \tag{10}$$

$$\mathrm{CIM}_i = \bar{\beta}_i.$$

The CILL and CIUL individuals divide the domain of each gene into four subintervals: $D_i \equiv I_i^1 \cup I_i^2 \cup I_i^3 \cup I_i^4$, where $I_i^1 \equiv [a_i, \mathrm{CILL}_i)$; $I_i^2 \equiv [\mathrm{CILL}_i, \mathrm{CIM}_i]$; $I_i^3 \equiv (\mathrm{CIM}_i, \mathrm{CIUL}_i]$; $I_i^4 \equiv (\mathrm{CIUL}_i, b_i]$; a_i and b_i being the bounds of the domain.

2.2 Estimators based on L_1 norm

In the previous section we have assumed that the values of the genes of the best individuals follow a normal distribution and we have used the estimators $\bar{\beta}_i$ and S_{β_i}, which are the most efficient estimators for normal distributions. If the normality hypothesis is not fulfilled the estimators will be negatively affected. This situation may be common in multimodal problems, and in the first stages of the evolution when the best individuals are spread over the search space. On the other hand, the influence of the gross error[7] over the mean can produce large fluctuations of the confidence interval from one generation to the next. This will make it difficult to get a stable search direction. This circumstance will also produce an increment of S_{β_i} and the construction of less homogeneous confidence intervals.

In order to avoid these problems, we will also use L_1 norm to estimate robust intervals that do not depend on the distribution of the individuals. So, using model (2), we analyse an estimator of the localisation parameter for the ith gene based on the minimisation of the dispersion function induced by the L_1 norm. The L_1 norm is defined as

$$\|\beta_i^*\|_1 = \sum_{j=1}^{n} |\beta_{ij}^*|, \tag{11}$$

hence the associated dispersion and negative gradient functions are given respectively by

$$D_1(\mu_{\beta_i^*}) = \sum_{j=1}^{n} |\beta_{ij}^* - \mu_{\beta_i^*}| \tag{12}$$

and

$$S_1(\mu_{\beta_i^*}) = \sum_{j=1}^{n} \mathrm{sign}(\beta_{ij}^* - \mu_{\beta_i^*}). \tag{13}$$

Letting $H(\beta_i^*)$ denote the empirical distribution function, the estimating equation is defined by:

$$n^{-1}\sum_{j=1}^{n} \mathrm{sign}(\beta_{ij}^* - \mu_{\beta_i^*}) = \int \mathrm{sign}(\beta_i^* - \mu_{\beta_i^*})\,dH(\beta_i^*) = 0. \tag{14}$$

[7] An estimator is not sensitive to gross error if the presence of a small number of outliers cannot have a disproportionate effect on the estimate.

The solution of the equation is the median, $\hat{\mu}_{\beta_i^*} = M_{\beta_i^*}$, of the distribution of the β_i^* (Hettmansperger and McKean 1998), defined by

$$M_{\beta_i^*} = \begin{cases} \dfrac{\beta_{i(j)}^* + \beta_{i(j+1)}^*}{2} & \text{if } n = 2j, \\ \beta_{i(j)}^* & \text{if } n = 2j+1. \end{cases} \tag{15}$$

The median is a better localisation estimator than the sample mean when the form of $H(\beta_i^*)$ distribution is not known.

We have assumed that the distribution function of β_i^*, $H(\beta_i^*)$, is continuous, so the probability that k of the n observations of the β_i^* will not be above the median $M_{\beta_i^*}$ is

$$P\left(\sum_{j=1}^{n} I(\beta_{ij}^* < M_{\beta_i^*}) = k\right) = \binom{n}{k} (H(M_{\beta_i^*}))^k (1 - H(M_{\beta_i^*}))^{n-k}, \tag{16}$$

where $I(.)$ is the indicator function. If we take into account that $H(M_{\beta_i^*}) = \frac{1}{2}$, the probability of the event is

$$\binom{n}{k} \left(\frac{1}{2}\right)^k \left(1 - \frac{1}{2}\right)^{n-k}. \tag{17}$$

So, the distribution of the statistic $\sum_{j=1}^{n} I(\beta_{ij}^* < M_{\beta_i^*})$, that is, the number of individuals whose β_i^* gene is not above the median, will be a binomial distribution of parameter n and $\frac{1}{2}$, $B(n, \frac{1}{2})$, that is

$$P\left(\sum_{j=1}^{n} I(\beta_{ij}^* < M_{\beta_i^*}) = k\right) = \binom{n}{k} \left(\frac{1}{2}\right)^n \quad \text{for } k = 0, \dots, n. \tag{18}$$

We cannot use the studentization method (Kendall and Stuart 1977) to build the confidence interval of the estimator sample median $M_{\beta_i^*}$, because the binomial distribution does not fulfill the hypothesis to apply the method. In such a case, we apply the method of Neyman (1937), which does not depend on the sample distribution of the estimator of the localisation parameter.

Therefore, in order to determine the $(1-\alpha)$ confidence interval, assuming the values of the genes are ordered, we must find $\hat{\mu}_{\beta_{i(L)}^*}$ and $\hat{\mu}_{\beta_{i(U)}^*}$ such as that

$$P(\hat{\mu}_{\beta_{i(L)}^*} \le M_{\beta_i^*} \le \hat{\mu}_{\beta_{i(U)}^*}) = 1 - \alpha. \tag{19}$$

Hence, we must find

$$\hat{\mu}_{\beta_{i(L)}^*} = \inf\left\{ t : \sum_{j=1}^{n} I(\beta_{i(j)}^* < t) < n - k \right\} = \#\{j : (\beta_{i(j)}^* < t) < n - k\}, \tag{20}$$

where inf is the infimum, # means the number of elements in the set, and k is given by $P(B(n, \frac{1}{2}) \le k) = \alpha/2$. Hence, $\hat{\mu}_{\beta_{i(L)}^*} = \beta_{i(k+1)}^*$. A similar argument shows that

$\hat{\mu}_{\beta^*_{i(U)}} = \beta^*_{i(n-k)}$. In summary, the $(1-\alpha)$ L_1 confidence interval based on the estimator $M_{\beta^*_i}$ is

$$[\beta^*_{i(k+1)}, \beta^*_{i(n-k)}] \quad \text{where } P\left(B\left(n, \frac{1}{2}\right) \leq k\right) = \alpha/2 \text{ determines } k. \tag{21}$$

This interval is a distribution-free or non-parametric confidence interval since the confidence coefficient is determined from the binomial distribution without making any assumption of the underlying gene distribution.

As the binomial distribution is discrete, it may happen that we cannot obtain discrete values of $\beta^*_{i(k+1)}$ and $\beta^*_{i(n-k)}$ that verify $P(\beta^*_{i(k+1)} \leq \mu_{\beta^*_i} \leq \beta^*_{i(n-k)}) = 1-\alpha$. This effect is specially important if the number of best individuals considered is small.

In such cases, a nonlinear interpolation method is used (Hettmansperger and McKean 1998) for obtaining the values of the lower bound, using $\beta^*_{i(k)}$ and $\beta^*_{i(k+1)}$, and the upper bound, using $\beta^*_{i(n-k)}$ and $\beta^*_{i(n-k+1)}$, of the confidence interval.

The interpolation method is the following. Let $\gamma = 1-\alpha$ be the desired confidence interval, and the two possible intervals taken from the binomial tables be $(\beta^*_{i(k)}, \beta^*_{i(n-k+1)})$, with a confidence coefficient of γ_k, and $(\beta^*_{i(k+1)}, \beta^*_{i(n-k)})$ with a confidence coefficient of γ_{k+1}, where $\gamma_{k+1} \leq \gamma \leq \gamma_k$. Then, the interpolated bounds of the interval are:

$$\hat{\mu}_{\beta^*_{i(L)}} = (1-\lambda)\beta^*_{i(k)} + \lambda\beta^*_{i(k+1)}, \quad \text{and} \tag{22}$$

$$\hat{\mu}_{\beta^*_{i(U)}} = (1-\lambda)\beta^*_{i(n-k+1)} + \lambda\beta^*_{i(n-k)}, \tag{23}$$

where

$$\lambda = \frac{(n-k)I}{k+(n-2k)I}, \quad \text{and} \tag{24}$$

$$I = \frac{\gamma_k - \gamma}{\gamma_k - \gamma_{k+1}}. \tag{25}$$

Thus, the distribution of the genes of the three virtual parents obtained using L_1 norm are independent of the distribution of the best individuals of the population. These individuals are given by:

$$\begin{aligned} \mathrm{CILL}_i &= \hat{\mu}_{\beta^*_{i(L)}}, \\ \mathrm{CIUL}_i &= \hat{\mu}_{\beta^*_{i(U)}}, \\ \mathrm{CIM}_i &= M_{\beta^*_i}. \end{aligned} \tag{26}$$

2.3 Analysis of applicability to discrete problems

The identification of the most promising search regions proposed in this paper requires a continuous search space. In discrete search spaces, where the values of the genes depend on the domain, the codification, and the constraints of the problem, this idea is difficult to apply. Nevertheless, we can work with the probability of each gene of having a certain value. To obtain such probability we must take into account the values that a gene can take, the constraints on those values, and other constraints given by the problem.

This information is necessary to identify the distribution function that the values of the genes of the bests individuals follow. Under the assumption of that distribution we should determine the best estimators for the localisation and dispersion values and construct the virtual individuals using these estimators. If the localisation estimator or the bounds of the confidence interval take values outside the domain, it would be necessary to define a mechanism to force the virtual individuals to the feasible region in order to be evaluated.

In summary, although the paper is devoted to improving crossover operator for real-coded genetic algorithms, whose field of application is optimisation problems in continuous domains, the study of a specific discrete problem with the ideas presented above might achieve a methodology for problems defined in discrete domains.

3 Crossovers based on virtual parents

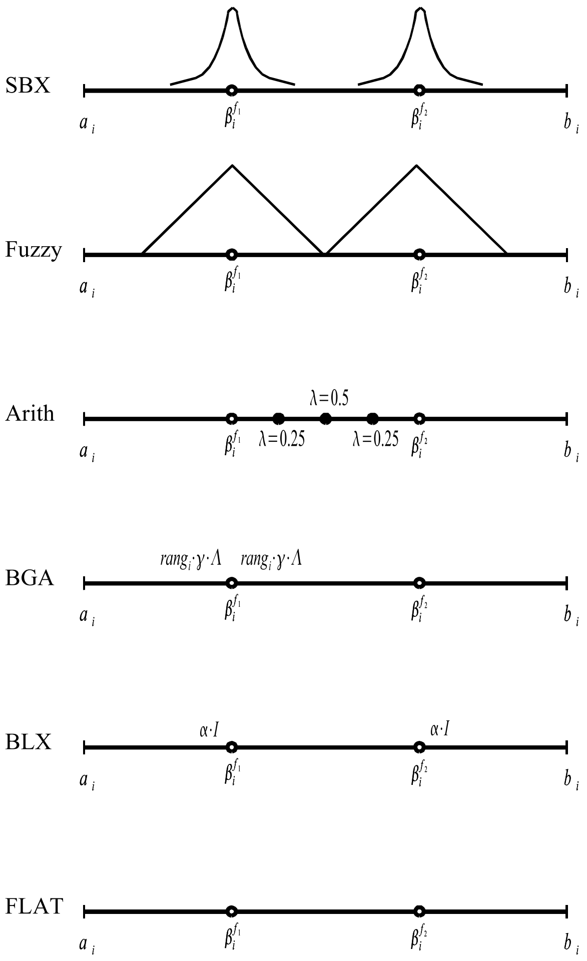
The proposed methodology uses the three virtual individuals obtained using norms L_1 and L_2 as virtual parents to be used in any crossover operator. The methodology creates one offspring $\beta^s = \{\beta_1^s, \beta_2^s, \ldots, \beta_i^s, \ldots, \beta_p^s\}$ from an individual of the population $\beta^f = \{\beta_1^f, \beta_2^f, \ldots, \beta_i^f, \ldots, \beta_p^f\}$ and a virtual individual VP $\in$ {CILL, CIM, CIUL}, following:

- $\beta_i^f \in I_i^1$: if the fitness of β^f is higher than CILL, then β_i^s = CrossoverVP(β_i^f, CILL$_i$), else β_i^s = CrossoverVP(CILL$_i$, β_i^f)
- $\beta_i^f \in I_i^2$ and the fitness of CILL is higher than CIM: if the fitness of β^f is higher than CILL, then β_i^s = CrossoverVP(β_i^f, CILL$_i$), else β_i^s = CrossoverVP(CILL$_i$, β_i^f)
- $\beta_i^f \in I_i^2$ and the fitness of CIM is higher than CILL: if the fitness of β^f is higher than CIM, then β_i^s = CrossoverVP(β_i^f, CIM$_i$), else β_i^s = CrossoverVP(CIM$_i$, β_i^f)
- $\beta_i^f \in I_i^3$ and the fitness of CIUL is higher than CIM: if the fitness of β^f is higher than CIUL, then β_i^s = CrossoverVP(β_i^f, CIUL$_i$), else β_i^s = CrossoverVP(CIUL$_i$, β_i^f)
- $\beta_i^f \in I_i^3$ and the fitness of CIM is higher than CIUL: if the fitness of β^f is higher than CIM, then β_i^s = CrossoverVP(β_i^f, CIM$_i$), else β_i^s = CrossoverVP(CIM$_i$, β_i^f)
- $\beta_i^f \in I_i^4$: if the fitness of β^f is higher than CIUL, then β_i^s = CrossoverVP(β_i^f, CIUL$_i$), else β_i^s = CrossoverVP(CIUL$_i$, β_i^f)

where CrossoverVP(*bestparent*, *worstparent*) represents the crossover operator adapted following the general rules:

- If the operator produces two descendants each one closer to each parent, as SBX and Fuzzy operators do, we will only generate the descendant closer to the parent with the best fitness.
- If the operator produces just one descendant near the best parents, as BGA does, the original formulation of the crossover is kept.

Fig. 1 Graphic representation of the original crossovers (**a**) and their corresponding adaptation using virtual parents for the case when $\beta_i^f \in I^1{}_i$ (**b**). The *dotted line* shows the region where the offspring is generated for the case when the fitness of β^f is better than the fitness of VP = {CILL, CIM, CIUL}, and the *continuous line* the region otherwise



(a)

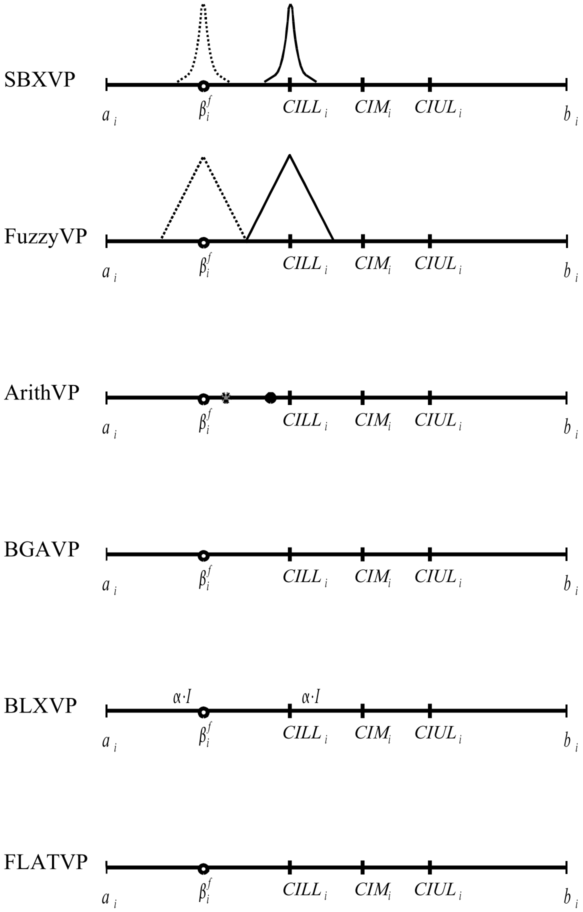
- If the operator uses an interval bounded by the parent, as Flat crossover does, or somewhat wider, BLX, the bound of the interval closest to the worst parent will be moved to the medium point of the interval made up by the two parents.

In the following sections we show the adaptation of several of the most widely used crossover operators using these basic principles. The operators chosen are SBX, Fuzzy, Arithmetical, BGA, BLX and Flat. Figure 1 shows a graphic representation of the original crossovers (on the left) and their corresponding adaptation using virtual parents for the case when $\beta_i^f \in I^1{}_i$ (on the right). The dotted line shows the region where the offspring is generated for the case when the fitness of β^f is better than the fitness of $VP =$ {CILL, CIM, CIUL}, and the continuous line the region otherwise.

3.1 SBX crossover

SBX (Simulated Binary Crossover) was proposed in (Deb and Agrawal 1995). The operator simulates the effect of one point binary crossover. Given two parents

Fig. 1 (Continued)



(b)

β^{f_1}, β^{f_2} the crossover generates two descendants whose genes $\beta_i^{s_1} = \frac{1}{2}[(1 + B_k)\beta_i^{f_1} + (1 - B_k)\beta_i^{f_2}]$ and $\beta_i^{s_2} = \frac{1}{2}[(1 - B_k)\beta_i^{f_1} + (1 + B_k)\beta_i^{f_2}]$, where $B_k \geq 0$ is a sample from a random number generator having the density

$$p(B) = \begin{cases} \frac{1}{2}(\eta + 1)B^{\eta}, & \text{if } 0 \leq B \leq 1, \\ \frac{1}{2}(\eta + 1)\frac{1}{B^{\eta+2}}, & \text{if } B > 1. \end{cases} \tag{27}$$

This distribution can easily be obtained from a uniform $u(0,1)$ random number source by the transformation

$$B(u) = \begin{cases} (2u)^{\frac{1}{\eta+1}}, & \text{if } u \leq \frac{1}{2}, \\ (\frac{1}{2(1-u)})^{\frac{1}{\eta+1}}, & \text{if } u > \frac{1}{2}. \end{cases} \tag{28}$$

Figure 1 shows how the offspring is generated near the two parents depending on the distribution function given above whose amplitude is governed by the para-

Table 1 Triangular probability distributions for fuzzy crossover

Distribution	Minimum	Mode	Maximum
$\phi_{\beta_i^{f_1}}$	$\beta_i^{f_1} - d\vert\beta_i^{f_2} - \beta_i^{f_1}\vert$	$\beta_i^{f_1}$	$\beta_i^{f_1} + d\vert\beta_i^{f_2} - \beta_i^{f_1}\vert$
$\phi_{\beta_i^{f_2}}$	$\beta_i^{f_2} - d\vert\beta_i^{f_2} - \beta_i^{f_1}\vert$	$\beta_i^{f_2}$	$\beta_i^{f_2} + d\vert\beta_i^{f_2} - \beta_i^{f_1}\vert$

meter η. A value of $\eta = 5$ has shown a good compromise between exploration and exploitation for a wide range of problems (Deb and Beyer 2001; Herrera et al. 2003).

SBXVP crossover Given a parent $\beta^f \in \boldsymbol{\beta}$ and a virtual individual VP = {CILL, CIM, CIUL} the genes of the offspring $\beta_i^s = \frac{1}{2}[(1 + B_k)\beta_i^f + (1 - B_k)\mathrm{VP}_i]$ if the fitness of β^f is better than the fitness of VP. Otherwise, $\beta_i^s = \frac{1}{2}[(1 - B_k)\beta_i^f + (1 + B_k)VP_i]$. Figure 1 shows the case when $\beta_i^f \in I^1{}_i$. The dotted line shows the distribution used for the generation of the offspring in the first case, and the continuous line in the second case.

3.2 Fuzzy crossover

In this operator (Voigt et al. 1995) given two parents β^{f_1}, β^{f_2} two descendants are generated in such a way that the probability that β_i^s takes a value z_i is given by the distribution $p(z_i) \in \{\phi_{\beta_i^{f_1}}, \phi_{\beta_i^{f_2}}\}$, where $\phi_{\beta_i^{f_1}}$ and $\phi_{\beta_i^{f_2}}$ are triangular probability distributions whose features for $\beta_i^{f_1} \leq \beta_i^{f_2}$ and $d \in [0, 1]$ are shown in Table 1.

The parameter d defines the amplitude of the triangular distribution, $d = 0.5$ being a suitable value (Voigt et al. 1995; Voigt 1995; Herrera and Lozano 2000) for a wide range of problems (Fig. 1).

FuzzyVP crossover This crossover is adapted in the same way as SBX crossover. If the fitness of β^f is better than the fitness of the VP the distribution used is $\phi_{\beta_i^f}$, otherwise ϕ_{VP_i}. These two distributions are shown in Fig. 1, the former with a dotted line and the latter with a continuous line.

3.3 Arithmetical crossover

Two descendants are created, $\beta_i^{s_1} = \lambda\beta_i^{f_1} + (1 - \lambda)\beta_i^{f_2}$ and $\beta_i^{s_2} = \lambda\beta_i^{f_2} + (1 - \lambda)\beta_i^{f_1}$, where $\lambda \in [0, 1]$ is a constant (Michalewicz 1992). This crossover tends to generate solutions near the centre of the search space. A value of $\lambda = 0.25$ is common in the works where this operator is used (Michalewicz 1992; Herrera et al. 2003) (Fig. 1).

ArithmeticalVP crossover If the fitness of β^f is better than the fitness of VP, $\beta_i^s = \lambda\mathrm{VP}_i + (1 - \lambda)\beta_i^f$, or $\beta_i^s = \lambda\beta_i^f + (1 - \lambda)\mathrm{VP}_i$ otherwise. With a value of $\lambda = 0.25$ offspring are generated near the best individual (Fig. 1). A value of $\lambda > 0.5$ implies inverting the two cases to keep the behaviour of the operator.

3.4 Linear BGA crossover

Given two parents β^{f_1}, β^{f_2} and considering that β^{f_1} is the individual with the best fitness, a descendant is created β^s whose genes $\beta_i^s = \beta_i^{f_1} \pm \text{rang}_i \gamma \Lambda$, where $\Lambda = \frac{\beta_i^{f_2} - \beta_i^{f_1}}{\|\beta^{f_1} - \beta^{f_2}\|}$. The sign "−" is chosen with probability 0.9. Usually $\text{rang}_i = 0.5(b_i - a_i)$ and $\gamma = \sum_{k=0}^{15} \alpha_k 2^{-k}$, where $\alpha_i \in \{0, 1\}$ is randomly generated with $p(\alpha_i = 1) = \frac{1}{16}$ (Schlierkamp-Voosen 1994). This operator is based on Mühlenbein's mutation (Mühlebein and Schlierkamp-Voosen 1993) and performs a search near the best individual, whose amplitude depends on both the distance between the values of the genes and the distances between the chromosomes (Fig. 1). The high probability of a negative sign makes the crossover very exploratory and most of the descendants are generated outside the interval defined by the parents.

Linear BGAVP crossover The only difference with the original crossover is that in this version of the operator only a parent from the population β^f is involved together with a virtual individual VP = {CILL, CIM, CIUL}. If β^f has a better fitness than VP, $\beta_i^s = \beta_i^f \pm \text{rang}_i \gamma \Lambda$, where $\Lambda = \frac{\text{VP}_i - \beta_i^f}{\|\beta^f - VP\|}$. Otherwise, $\beta_i^s = \text{VP}_i \pm \text{rang}_i \gamma \Lambda$, where $\Lambda = \frac{\beta_i^f - \text{VP}_i}{\|\text{VP} - \beta^f\|}$. Figure 1 shows both cases.

3.5 BLX crossover

Given two parents β^{f_1}, β^{f_2} a descendant is created β^s whose genes β_i^s are chosen randomly in the interval $[\min - I \cdot \alpha, \max + I \cdot \alpha]$, being $\max = \max(\beta_i^{f_1}, \beta_i^{f_2})$, $\min = \min(\beta_i^{f_1}, \beta_i^{f_2})$ and $I = \max - \min$ (Eshelman and Schaffer 1993). If $\alpha = 0$ this crossover is the same as Flat crossover. For $\alpha = 0.5$, the probability that the genes of the offspring take values within and without the interval of the values of their parents is the same. In (Herrera et al. 1998) different values of α are tested obtaining a best value of $\alpha = 0.5$.

BLXVP crossover Given a parent $\beta^f \in \boldsymbol{\beta}$ and a virtual individual VP = {CILL, CIM, CIUL} the genes β_i^s of the offspring are chosen randomly in the interval $[\min - I \cdot \alpha_L, \max + I \cdot \alpha_U]$, being $\max = \max(\beta_i^f, \text{VP}_i)$, $\min = \min(\beta_i^f, \text{VP}_i)$ and $I = \max - \min$. If the individual whose gene is min has a better fitness than the individual whose gene is max belongs to then $\alpha_L = -0.5$ and $\alpha_U = \alpha$, otherwise $\alpha_L = \alpha$ and $\alpha_L = -0.5$. In this way the bound of the interval closest to the worst parent is moved to the middle point of the interval defined by the two parents.

Figure 1 shows the case for $\beta^f \in I^1$. If the fitness of β^f is better than the fitness of CILL the offspring are generated in the dotted segment, otherwise they are generated in the continuous segment.

3.6 Flat crossover

It creates a descendant whose genes β_i^s are randomly generated in the interval $[\beta_i^{f_1}, \beta_i^{f_2}]$ (Radcliffe 1991). It is an exploitative crossover (Fig. 1).

FlatVP crossover As in the case of BLX crossover, if the fitness of β^f is better than the fitness of VP, β_i^s takes values in the interval $[\beta_i^f, \beta_i^f + (\mathrm{VP}_i - \beta_i^f)/2]$, otherwise it takes values in $[\mathrm{VP}_i, \mathrm{VP}_i + (\beta_i^f - \mathrm{VP}_i)/2]$ (Fig. 1).

4 Virtual parent effect

This section aims at analysing the effect of using virtual parents as they have been defined in the previous sections. We study this effect on BLX crossover, as it is one of the most widely used, and its two modifications BLXL2VP, when we use L_2 norm, and BLXL1VP, when we use L_1 norm.

The study is made using a unimodal function, Sphere (De Jong 1975), and a multimodal function, Rastrigin (Rastrigin 1974) (Table 2). In order to be able to study the effect of virtual parents we use 5 dimensions, a positive domain $x_i \in [0, 12]$ and an optimum value shifted to $\mathbf{x}^* = \{10, 10, 10, 10, 10\}$ with $f(\mathbf{x}^*) = 0$. The RCGA used is the same as the one in the experiments (see Sect. 5) with a limit of 100 generations. For constructing the confidence intervals we use $n = 5$, and a confidence coefficient of $1 - \alpha = 0.99$. The algorithm is run once for each operator using the same initial population and the same series of random values.

For the Sphere function, simple and strongly convex, Fig. 2 shows that BLX does not always generate offspring nearer the optimum values $x_i = 10$. Besides, for BLXL2VP (Fig. 3) and BLXL1VP (Fig. 4) the offspring is always nearer the optimum than the parents. The genes of the virtual parents have a marked tendency towards the optimum of the function. The offspring tends to approach the virtual parents and therefore the optimum of the function. Figures 3f and 4f show that the offspring generated using the virtual parents always has better fitness than its parents. On the other hand, the average fitness of the offspring generated by BLX (Fig. 2f) is almost always worse than the fitness of their parents. This has an effect on the convergence speed, and the RCGA using virtual parents converges faster.

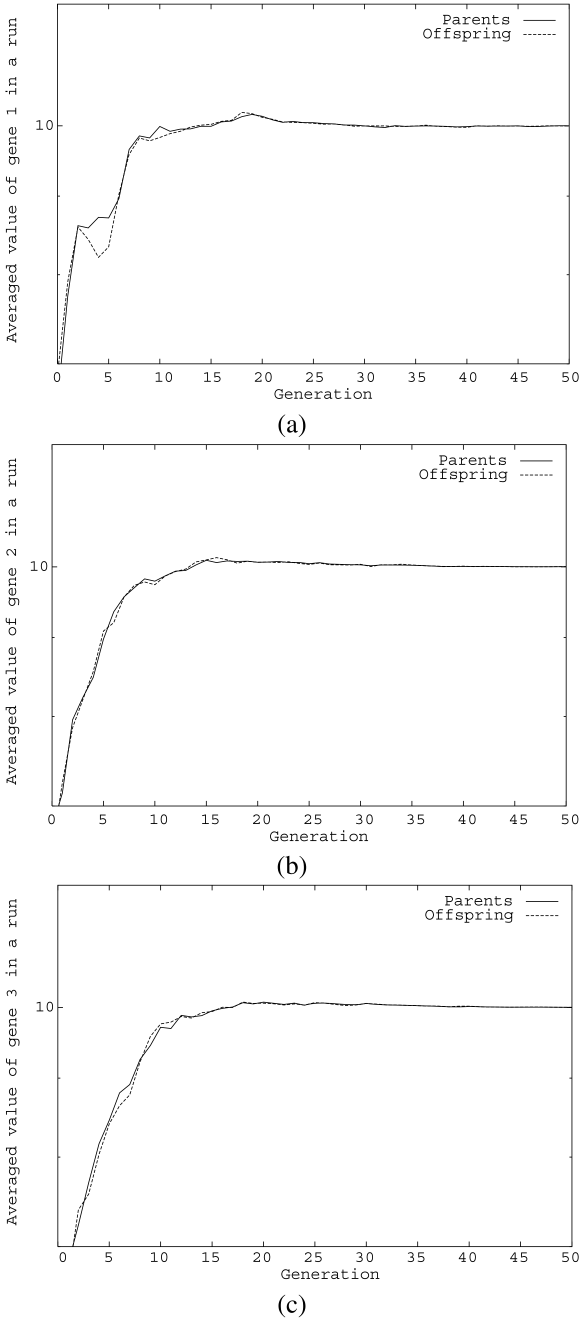
The Rastrigin function has a contour made up of a large number of local minima whose value increases with the distance to the global minimum. As in the case of Sphere, BLX does not always generates offspring closer to the optimum than their parents (Fig. 5). The multimodality of this function makes the hypothesis of normality of the best n individuals less probable. This might explain the poor performance of BLXL2VP (Fig. 6). On the other hand, the confidence intervals constructed for BLXL1VP (Fig. 7) are independent of the distribution of the individuals, and their behaviour is better. The multimodality of the function prevents the offspring always being nearer the optimum than their parents, however in most cases the offspring is closer to the optimum. Figure 7f shows that the average fitness of the descendants using BLXL1VP is better, as a general rule, than the fitness of their parents. Moreover, the offspring generated by BLXL2VP (Fig. 6f) does not consistently generate better offspring. The behaviour of BLX (Fig. 5f) is similar to the behaviour for Sphere, the offspring has a fitness similar to their parents.

The parameter n is responsible for determining the region where the search will be directed. If n is small, the population will move to the most promising individuals quickly. This may be convenient for increasing the convergence speed. Nevertheless,

Table 2 Definition of each function together with its features

Function	Definition	Multimodal?	Separable?	Regular?
Sphere	$f_{\mathrm{Sph}}(\mathbf{x}) = \sum_{i=1}^{p} x_i^2$ $x_i \in [-5.12, 5.12]$ $\mathbf{x}^* = (0, 0, \ldots, 0)$; $f_{\mathrm{Sph}}(\mathbf{x}^*) = 0$	no	yes	n/a
Schwefel's Double sum	$f_{\mathrm{SchDS}}(\mathbf{x}) = \sum_{i=1}^{p}(\sum_{j=1}^{i} x_j)^2$ $x_i \in [-65.536, 65.536]$ $\mathbf{x}^* = (0, 0, \ldots, 0)$; $f_{\mathrm{SchDS}}(\mathbf{x}^*) = 0$	no	no	n/a
Rosenbrock	$f_{\mathrm{Ros}}(\mathbf{x}) = \sum_{i=1}^{p-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ $x_i \in [-2.048, 2.048]$ $\mathbf{x}^* = (1, 1, \ldots, 1)$; $f_{\mathrm{Ros}}(\mathbf{x}^*) = 0$	no	no	n/a
Rastrigin	$f_{\mathrm{Ras}}(\mathbf{x}) = 10p + \sum_{i=1}^{p}(x_i^2 - 10\cos(2\pi x_i))$ $x_i \in [-5.12, 5.12]$ $\mathbf{x}^* = (0, 0, \ldots, 0)$; $f_{\mathrm{Ras}}(\mathbf{x}^*) = 0$	yes	yes	n/a
Schwefel	$f_{\mathrm{Sch}}(\mathbf{x}) = 418.9829 \cdot p + \sum_{i=1}^{p} x_i \sin(\sqrt{\lvert x_i \rvert})$ $x_i \in [-512.03, 511.97]$ $\mathbf{x}^* = (-420.9687, \ldots, -420.9687)$; $f_{\mathrm{Sch}}(\mathbf{x}^*) = 0$	yes	yes	n/a
Weierstrass	$f_{\mathrm{Wei}}(\mathbf{x}) = \sum_{i=1}^{p}(\sum_{k=0}^{k_{\max}}[a^k \cos(2\pi b^k x_i]) + 2p$ $x_i \in [-0.5, 0.5]$; $a = 0.5$; $b = 3$; $k_{\max} = 20$ $\mathbf{x}^*(0, 0, \ldots, 0)$; $f_{\mathrm{Wei}}(\mathbf{x}^*) = 0$	yes	no	n/a
Schaffer	$f_{\mathrm{Schaf}}(x_1, x_2, x_3 \cdots x_p)$ $= F(x_1, x_2) + F(x_2, x_3) + \cdots + F(x_p, x_1)$ $F(x, y) = 0.5 + \frac{\sin^2(\sqrt{x^2+y^2}) - 0.5}{(1+0.001(x^2+y^2))^2}$ $x_i \in [-100, 100]$ $\mathbf{x}^*(0, 0, \ldots, 0)$; $f_{\mathrm{Schaf}}(\mathbf{x}^*) = 0$	yes	no	n/a
Ackley	$f_{\mathrm{Ack}}(\mathbf{x}) = 20 + e - 20\exp(-0.2\sqrt{\frac{1}{p}\sum_{i=1}^{p} x_i^2})$ $-\exp(\frac{1}{p}\sum_{i=1}^{p}\cos(2\pi x_i))$ $x_i \in [-30, 30]$ $\mathbf{x}^* = (0, 0, \ldots, 0)$; $f_{\mathrm{Ack}}(\mathbf{x}^*) = 0$	yes	no	yes
Griewangk	$f_{\mathrm{Gri}}(\mathbf{x}) = 1 + \sum_{i=1}^{p} \frac{x_i^2}{400p} - \prod_{i=1}^{p}\cos(\frac{x_i}{\sqrt{i}})$ $x_i \in [-600, 600]$ $\mathbf{x}^*(0, 0, \ldots, 0)$; $f_{\mathrm{Gri}}(\mathbf{x}^*) = 0$	yes	no	yes
Fletcher Powell	$f_{\mathrm{Fle}}(\mathbf{x}) = \sum_{i=1}^{p}(A_i - B_i)^2$ $A_i = \sum_{j=1}^{p}(a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j)$ $B_i = \sum_{j=1}^{p}(a_{ij}\sin x_j + b_{ij}\cos x_j)$ $x_i, \alpha_i \in [-\pi, \pi]$; $a_{ij}, b_{ij} \in [-100, 100]$ $\mathbf{x}^* = \alpha$; $f_{\mathrm{Fle}}(\mathbf{x}^*) = 0$	yes	no	no
Langerman	$f_{\mathrm{Lan}}(\mathbf{x}) = -\sum_{i=1}^{m} c_i \cdot \exp(-\frac{1}{\pi}\sum_{j=1}^{p}(x_j - a_{ij})^2)$ $\cdot\cos(\pi\sum_{j=1}^{p}(x_j - a_{ij})^2)$ $x_i \in [0, 10]$; $m = p$ $\mathbf{x}^* =$ random; $f_{\mathrm{Lan}}(\mathbf{x}^*) =$ random	yes	no	no

Fig. 2 Average values of the 5 genes of the parents and offspring (**a**–**e**) and fitness of the fittest individual and average fitness of the parents and offspring (**f**) for Sphere using BLX



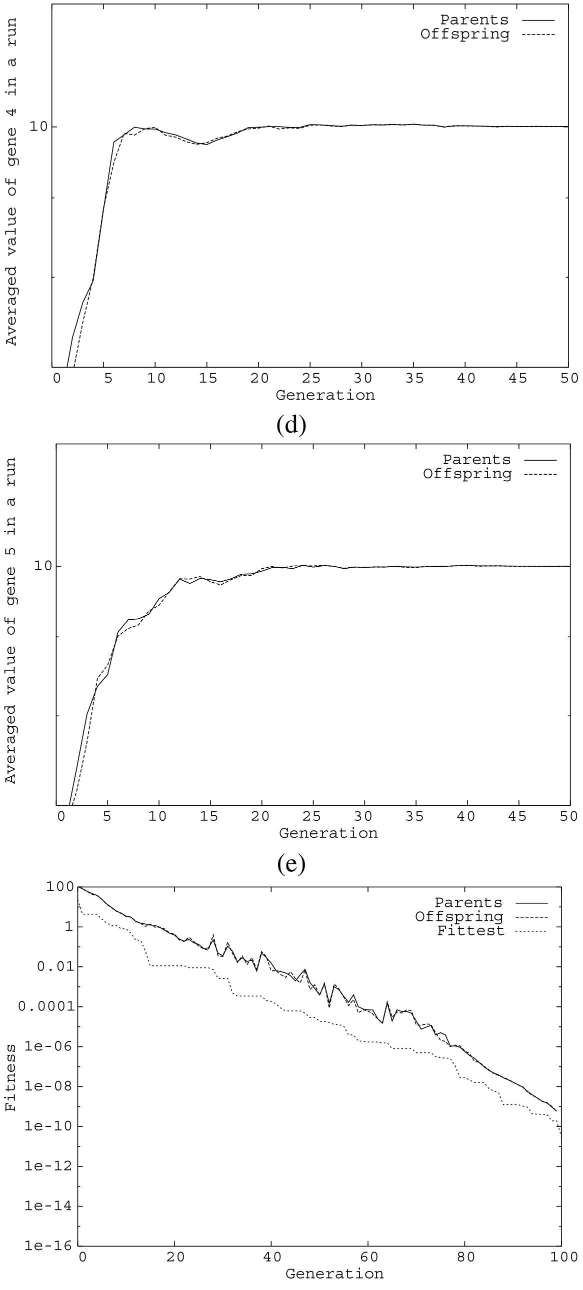
(a)

(b)

(c)

it can produce a premature convergence to suboptimal values. If n is large, both the shifting and the speed of convergence will be smaller. However, the evolutionary process will be more robust, this feature being perfectly adequate for the optimisation of multimodal, non-separable, highly epistatic functions.

Fig. 2 (Continued)



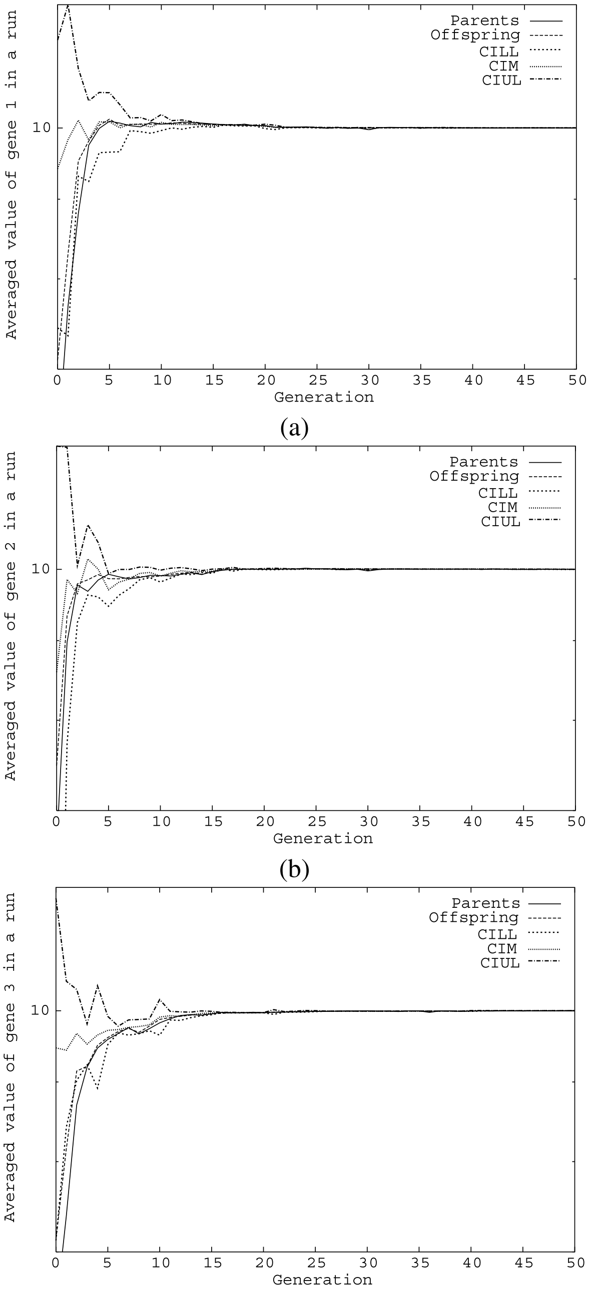
(d)

(e)

(f)

Previous works (Hervás-Martínez et al. 2003; Ortiz-Boyer et al. 2005; Hervás-Martínez and Ortiz-Boyer 2005) show that the selection of the best $n = 5$ individuals of the population would suffice for obtaining a localisation estimator good enough to guide the search process even for multimodal functions where a small

Fig. 3 Average values of the 5 genes of the parents and offspring together with values of the three virtual parents (**a**–**e**) and fitness of the fittest individual and average fitness of the parents and offspring (**f**) for Sphere using BLXL2VP



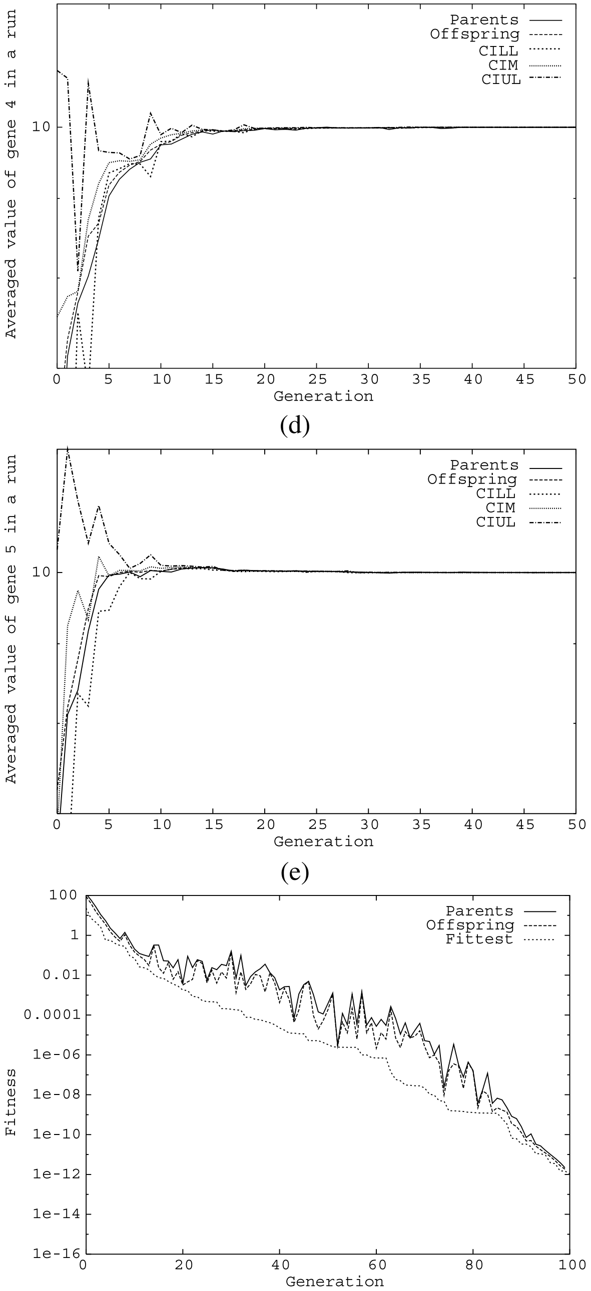
(a)

(b)

(c)

value of n could favour the convergence to local optima. However, if the virtual parents have a worse fitness than the parent from the population, the offspring is generated near the latter, and the domain can be explored in multiple di-

Fig. 3 (Continued)



(d)

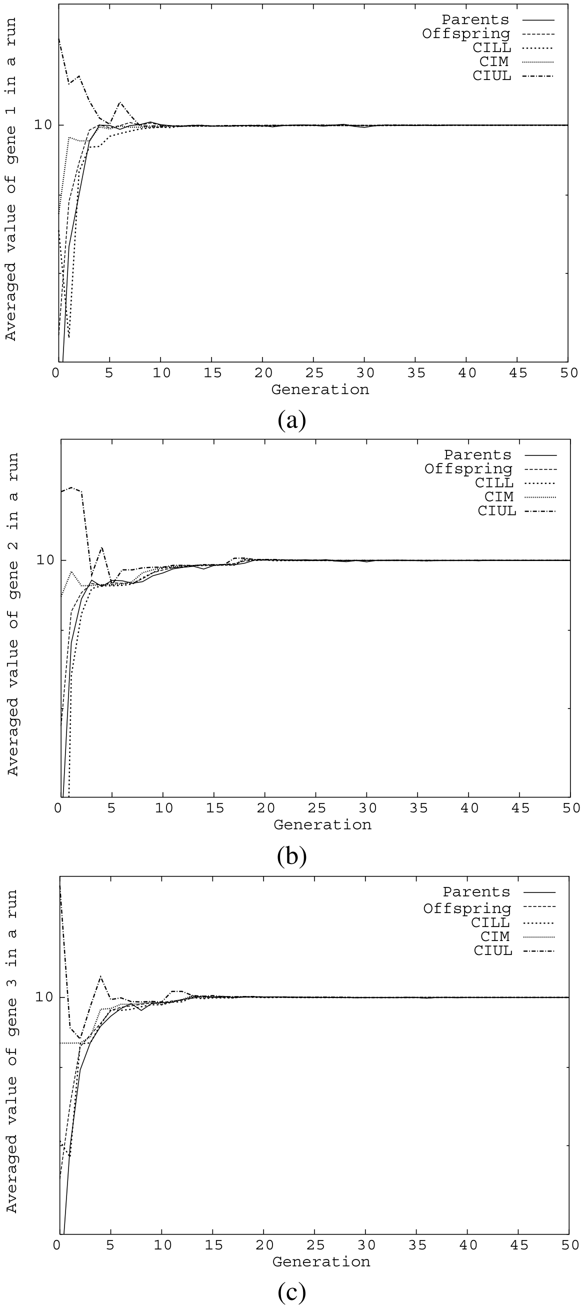
(e)

(f)

rections. In this way, the premature convergence to suboptimal virtual parents is avoided.

Figures 6 and 7 show BLXL2VP and BLXL1VP are able to avoid these local minima for Rastrigin function. If the virtual parents have worse fitness than the individual

Fig. 4 Average values of the 5 genes of the parents and offspring together with values of the three virtual parents (**a**–**e**) and fitness of the fittest individual and average fitness of the parents and offspring (**f**) for Sphere using BLXL1VP



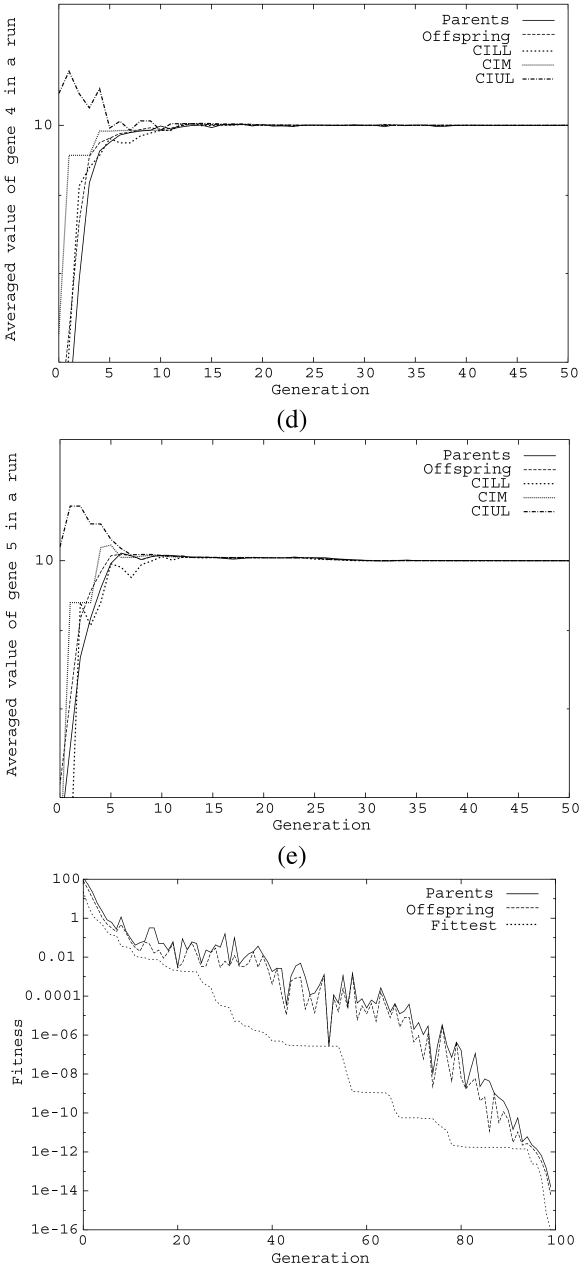
(a)

(b)

(c)

with whom it mates, the offspring is not generated near these virtual parents, but in the proximity of the other parent. In this way, the displacement of the population towards regions where the improvement of the fitness is not significant is prevented. So, in several cases, after a phase where the genes of the virtual parents converge to a

Fig. 4 (Continued)



(d)

(e)

(f)

local optimum, this optimum is left as fitter individuals are generated in its proximity which allows the localisation of a more promising region.

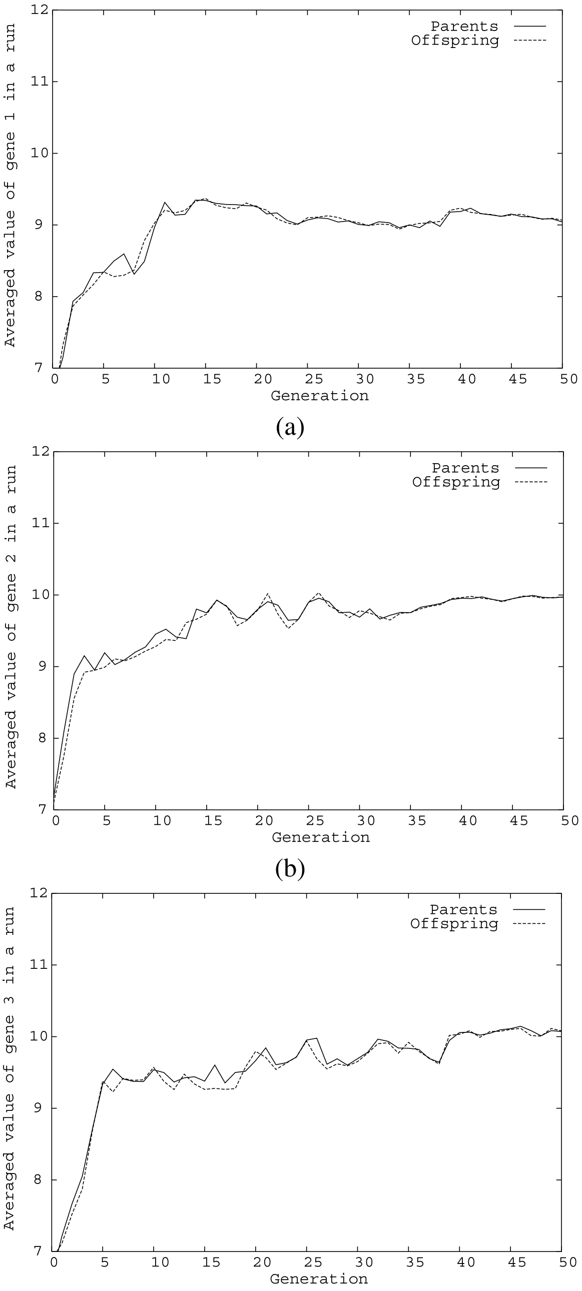


Fig. 5 Average values of the 5 genes of the parents and offspring (**a**–**e**) and fitness of the fittest individual and average fitness of the parents and offspring (**f**) for Rastrigin using BLX

(a)

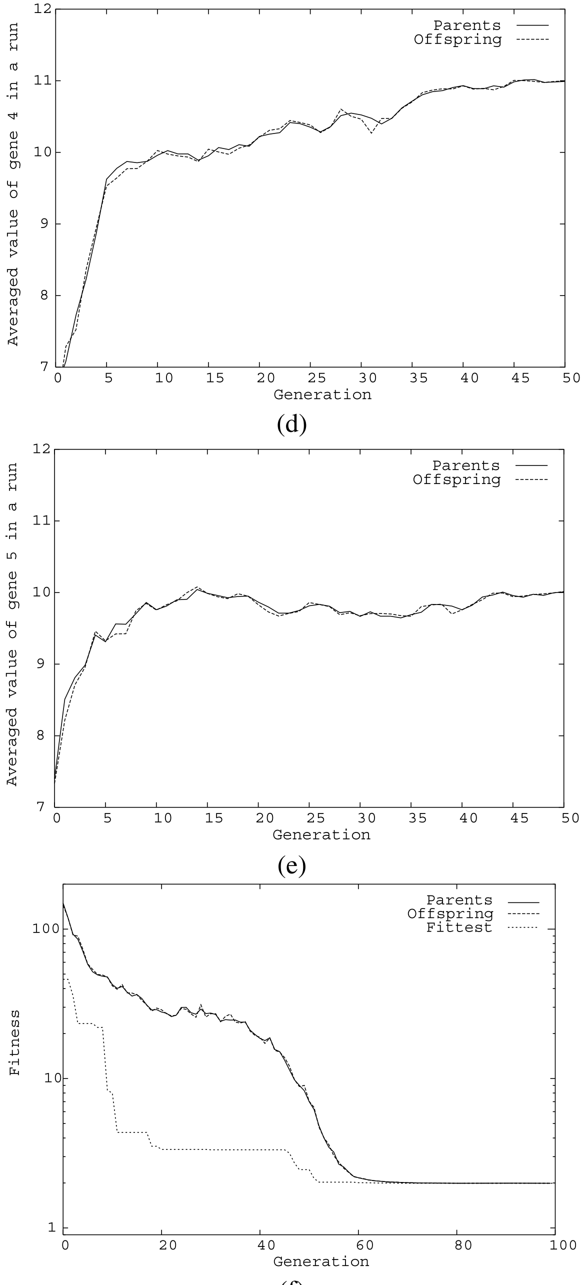
(b)

(c)

5 Experimental setup

In order to study the performance of the proposed methodology we use a generational RCGA (Bäck et al. 1997) with a population of 101 individuals randomly initialised.

Fig. 5 (Continued)

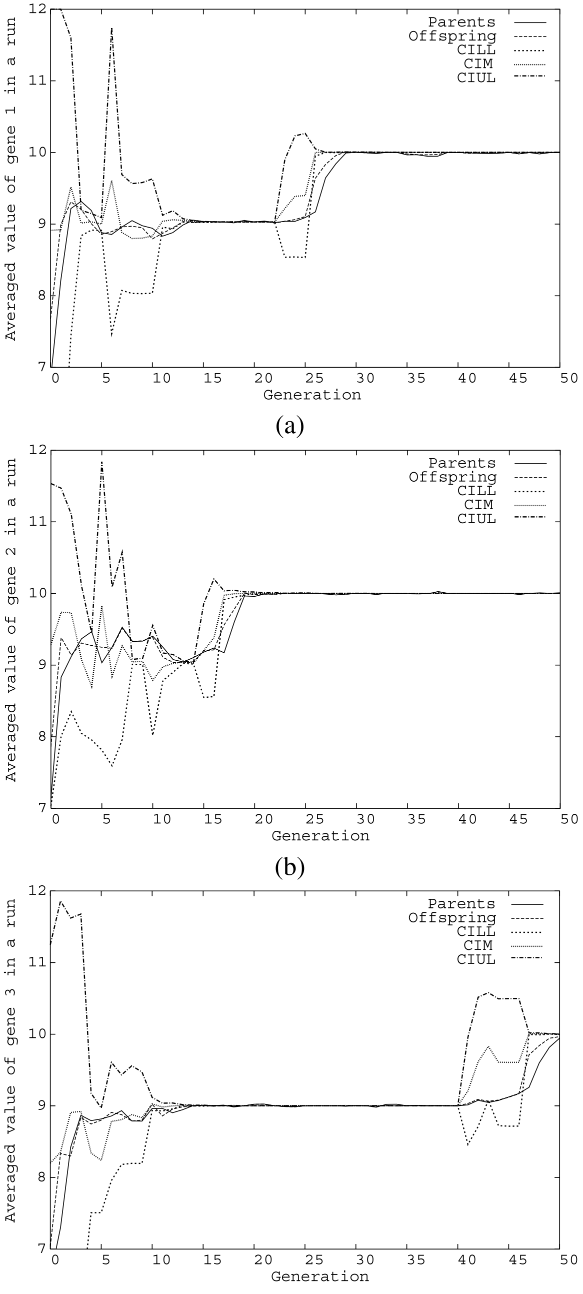


(d)

(e)

(f)

In (Zhang and Kim 2000) an experimental comparative study was carried out of the performance of different selection methods. The study concludes that the methods of ranking and tournament selection obtain better results than the methods of proportional and Genitor selection. We chose the binary tournament selection for its



(a)

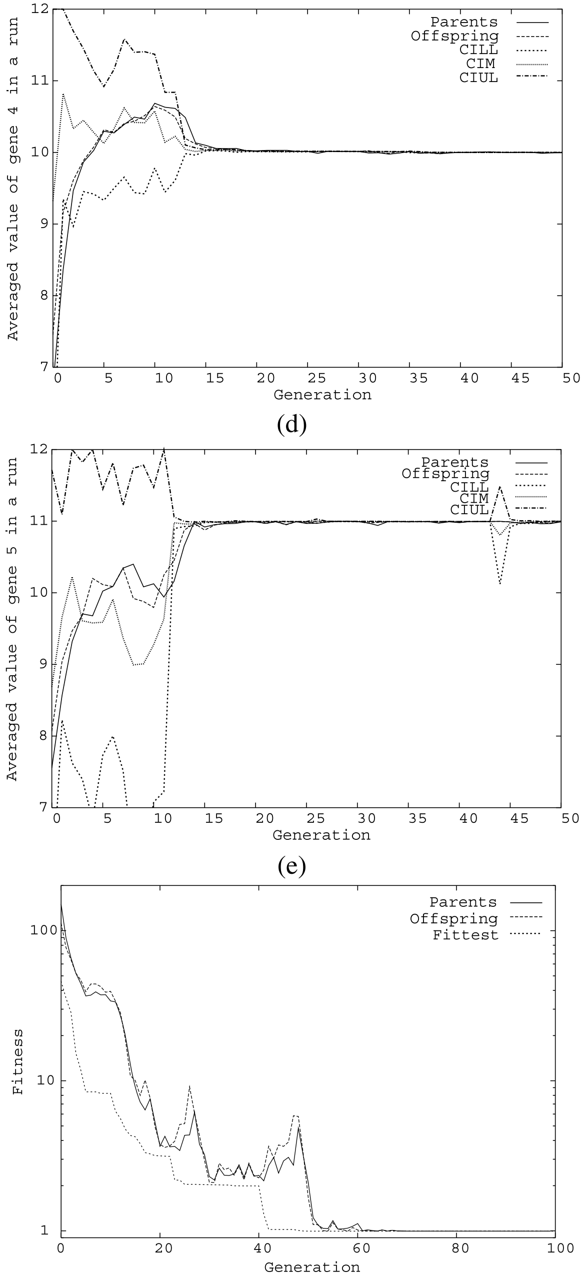
(b)

(c)

Fig. 6 Average values of the 5 genes of the parents and offspring together with values of the three virtual parents (**a**–**e**) and fitness of the fittest individual and average fitness of the parents and offspring (**f**) for Rastrigin using BLXL2VP

simplicity. Tournament selection runs a tournament between two individuals and selects the winner. In order to assure that the best individuals always survive to the next generation, we use elitism, the best individual of the population in generation t is always included in the population in generation $t + 1$. The convenience of the use

Fig. 6 (Continued)



(d)

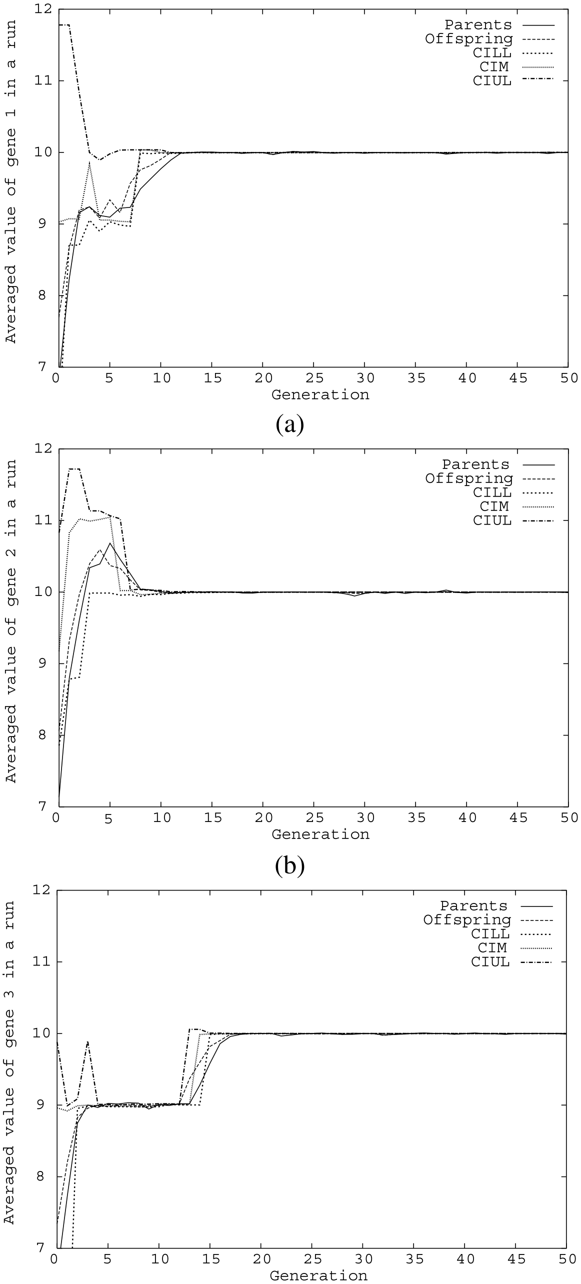
(e)

(f)

of elitism has been proved, both theoretically (Rudolph 1994) and empirically (Bäck 1996; Michalewicz 1992; Zhang and Kim 2000).

As mutation operator we have chosen the non-uniform mutation with parameter $b = 5$ (Michalewicz 1992) as its dynamic nature makes it very suitable for a wide

Fig. 7 Average values of the 5 genes of the parents and offspring together with values of the three virtual parents (**a**–**e**) and fitness of the fittest individual and average fitness of the parents and offspring (**f**) for Rastrigin using BLXL1VP



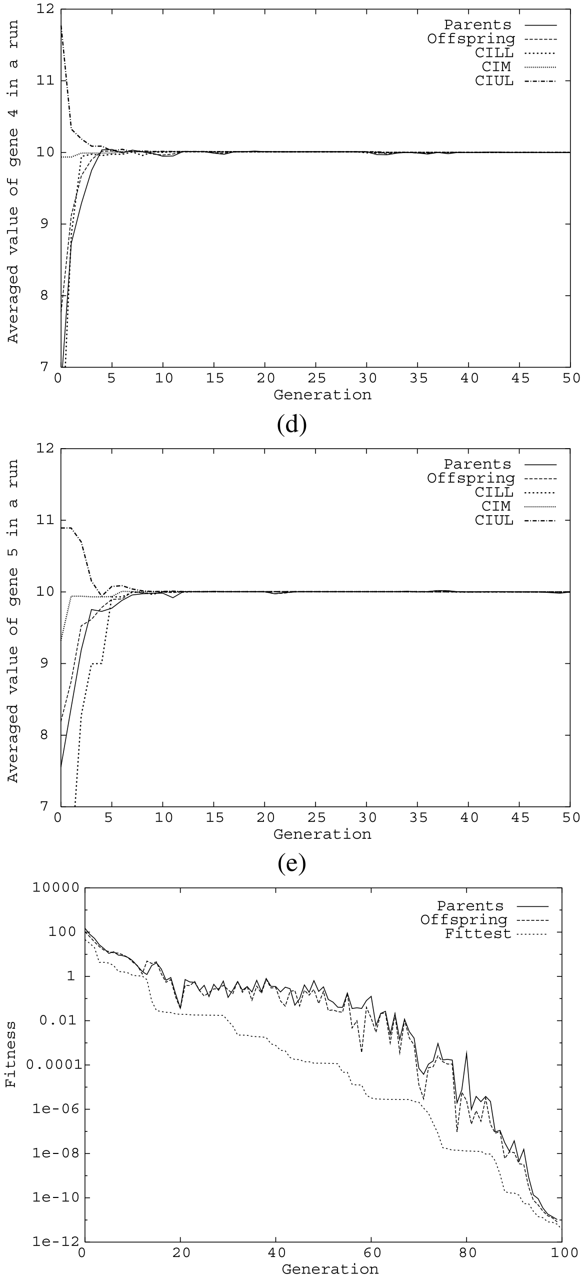
(a)

(b)

(c)

variety of problems (Herrera and Lozano 2000). This operator performs a uniform search in the initial stages of the evolution, and a localised search in the final stages allowing fine tuning of the solution. For the mutation probability, values in the interval $p_m \in [0.001, 0.1]$ are usual (De Jong 1975; Herrera et al. 1998; Michalewicz

Fig. 7 (Continued)



(d)

(e)

(f)

1992; Bäck 1996). We have chosen a value of $p_m = 0.05$ for both models. We have used a crossover probability of $p_c = 0.6$ because is commonly used in the literature (De Jong 1975; Herrera et al. 1998). The general structure of the genetic algorithm is shown in Fig. 8.

Fig. 8 Structure of the genetic algorithm, where t is the current generation

```
Genetic algorithm
begin
    $t \leftarrow 0$
    initialize $\boldsymbol{\beta}(t)$
    evaluate $\boldsymbol{\beta}(t)$
    while (not stop-criterion) do
    begin
        $t \leftarrow t + 1$
        select $\boldsymbol{\beta}(t)$ from $\boldsymbol{\beta}(t-1)$
        crossover $\boldsymbol{\beta}(t)$
        mutate $\boldsymbol{\beta}(t)$
        evaluate $\boldsymbol{\beta}(t)$
    end
end
```

The parameters of the operator are common to the standard version and when virtual parents are used. Following the literature with use SBX with $\eta = 5$, Fuzzy crossover with $d = 0.5$, Arithmetical crossover with $\lambda = 0.25$, linear BGA crossover with $\text{rang}_i = 0.5(b_i - a_i)$ and BLX crossover with $\alpha = 0.5$. For constructing the confidence intervals we use $n = 5$. Although in multimodal functions a larger value might be suitable, previous works (Ortiz-Boyer et al. 2005; Hervás-Martínez and Ortiz-Boyer 2005; Hervás-Martínez et al. 2003) show that a value of $n = 5$ is large enough to identify the most promising regions of the search space. For obtaining the confidence interval of the localisation estimator we use a confidence coefficient of $1 - \alpha = 0.99$.

Each experiment is repeated 30 times with different random seeds following the recommendations of (Czarn et al. 2004) for avoiding noise due to seed. We have used a different random series for each implemented item: initialisation of the population, selection, crossover, and mutation. For each set of experiments with the same algorithm we use a different set of seeds. The sets of seeds is common for all the experiments. That is, seeds for experiment 1 of algorithm a is different from the set of seeds for experiment 2 of algorithm a, but experiment 1 of algorithm b uses the same seeds as experiment 1 of algorithm a.

The stopping criterion is the number of evaluations of the fitness function and we use a limit of 300,000 evaluations (Eiben et al. 1998; De Jong et al. 1998). The precision of the solutions is bounded by the precision of the data type used in the implementation of the genetic algorithm. We have used a double precision data type of 64 bits following the specification ANSI/IEEE STD 754-1985 (IEEE Standard for Binary Floating-Point Arithmetic). This data type has a precision of 15–17 digits.

For the evaluation of the performance of the proposed methodology with respect to the problem typology we have used a set of well characterised functions instead of a large number of functions. This set is made up of functions with different modality, separability and regularity. In this way, we try to discover whether the use of virtual parents can improve the performance of a RCGA in specific types of functions.

Taking into account that dimensionality is a factor that has effects on the complexity of the functions (Friedman et al. 1994), and in order to establish the same degree of difficulty, we have chosen a dimensionality of $p = 30$ for all the functions. Table 2 shows the definition of the functions, with their main features and the localisation of the optima.

The Sphere function (De Jong 1975) is a simple and strongly convex function. The Schwefel's double sum function (Schwefel 1995) is a function whose gradient is not oriented along their axis due to the correlation among their variables. In this way, the algorithms that use the gradient method converge very slowly. The Rosembrock function (Rosenbrock 1960), defined as a two-dimensional function, has a deep valley with the shape of a parabola of the form $x_1^2 = x_2$ that leads to the global minimum. The non-linearity of the valley makes many algorithms converge slowly because they change the direction of the search repeatedly. The extended version of this function that we use was proposed by Spedicato (1975). Many authors considered this function as challenging for any optimisation algorithm (Schlierkamp-Voosen 1994). Its difficulty is mainly due to the non-linear interaction among its variables.

The Rastrigin function (1974) has a contour made up of a large number of local minima whose value increases with the distance to the global minimum. The Schwefel function (1981) is a function whose main difficulty is the existence of a second best minimum far from the global minimum, where many search algorithms are trapped. Moreover, the global minimum is near the bounds of the domain.

The Weierstrass function (1872) is highly multimodal, continuous everywhere but differentiable nowhere. This shows the fractal nature, that is, it doesn't get any "smoother" as one looks closer (a prerequisite for differentiability).

The expanded Schaffer's F6, or sine envelop sine wave a function (Schaffer et al. 1989), which we call Schaffer function or f_{Schaf}, is highly multimodal and nonseparable. Its global minimum is encircled by countless second minimum points. Since the second minimum points have a larger absorbing basin, they are extremely deceptive.

The Ackley (1987) function has an exponential term that covers its surface with numerous local minima. The complexity of this function is moderate. An algorithm that only uses the steepest gradient descent will be trapped in a local optima, but any search strategy that analyses a wider region will be able to cross the valley among the optima and achieve better results. The Griewangk function (Bäck et al. 1997) has a product term that introduces strong interdependence among the variables. As in the Ackley function, the optima of the Griewangk function are regularly distributed.

The functions of Fletcher–Powell (1963) and Langerman (Bersini et al. 1996) are highly multimodal, as Ackley and Griewangk, but they are non-symmetrical and their local optima are randomly distributed. In this way, the objective function has no implicit symmetry advantages that might simplify the optimisation for certain algorithms. Fletcher–Powell function achieves the random distribution of the optima choosing randomly the values of the matrices **a** and **b**, and of the vector α. We have used the values provided in (Bäck 1996). For the Langerman function, we have used the values of **a** and **c** referenced in (Eiben and Bäck 1997). In contrast with the rest of functions, Langerman is a maximisation problem.

6 Experimental results

This work is focused on the improvement of one of the components of a RCGA, so it is not the aim of this paper to achieve the highest quality solutions for the considered

Table 3 Average value and standard deviation of the 30 runs for each function with respect to the original crossover, X, and their corresponding alternatives XL2VP and XL1VP. Columns α^{X}_{XL2VP}, α^{XL1VP}_{X} and α^{XL2VP}_{XL1VP} show the p-values of the multiple comparison tests between the average fitness of the original crossover X and XL2VP, the original crossover X and XL1VP, and between XL2VP and XL1VP. Values are marked with "*" when the Levene test suggested a Bonferroni test. In other cases a Tamhane test is used. "+" and "−" show the results of the sign test, "+" meaning better and "−" worse

Crossover	Mean	St. Dev.	α^{X}_{XL2VP}	Mean	St. Dev.	α^{XL2VP}_{XL1VP}	Mean	St. Dev.	α^{XL1VP}_{X}
	X (Original)			XL2VP			XL1VP		
f_{Sph}									
SBX	5.180e–12^3	2.739e–12	0.000*	2.590e–16^1 +	8.417e–17	1.000*	4.589e–16^2 +	9.987e–17	0.000*
Fuzzy	1.007e–15^3	6.108e–16		2.220e–16^1 +	2.507e–32		4.071e–16^2 +	1.315e–16	
Arithmetical	2.746e–15^3	1.071e–15		2.813e–16^1 +	9.987e–17		5.107e–16^2 +	1.323e–16	
BGA	4.773e–13^2	2.996e–13	0.000*	1.058e–12^3 –	5.747e–13	0.000*	4.361e–14^1 +	2.509e–14	0.000*
BLX	2.590e–16^2	8.417e–17		1.998e–16^1 +	6.775e–17		3.553e–16^3 –	1.251e–16	
Flat	2.428e–15^3	1.405e–15	0.000*	3.109e–16^1 +	1.251e–16	0.013*	5.995e–16^2 +	1.764e–16	0.000*
f_{SchDS}									
SBX	4.536e–01^3	4.772e–01	0.000	9.935e–03^1 +	1.078e–02	0.000	4.755e–02^2 +	4.524e–02	0.000
Fuzzy	3.577e+01^3	1.752e+01	0.000	5.792e–03^1 +	5.050e–03	0.000	3.848e–02^2 +	2.699e–02	0.000
Arithmetical	2.145e+00^3	1.153e+00	0.000	7.873e–03^1 +	5.596e–03	0.000	3.105e–01^2 +	2.169e–01	0.000
BGA	1.697e–02^1	1.486e–02	0.130	2.386e–02^2 –	1.088e–02	0.000	5.824e–02^3 –	3.460e–02	0.000
BLX	1.430e–01^3	9.285e–02	0.000	5.233e–03^1 +	4.882e–03	0.000	5.078e–02^2 +	2.771e–02	0.000
Flat	1.514e+00^3	9.672e–01	0.000	1.099e–02^1 +	9.547e–03	0.000	2.688e–01^2 +	2.097e–01	0.000
f_{Ros}									
SBX	3.848e+01^3	2.362e+01	0.010	2.469e+01^1 +	6.382e–01	0.823	2.682e+01^2 +	1.487e+01	0.077
Fuzzy	2.510e+01^2	5.160e+00	0.965	2.469e+01^1 +	1.026e+00	0.998	2.521e+01^3 –	1.816e+01	1.000
Arithmetical	2.663e+01^2	1.096e–01	0.000	2.520e+01^1 +	2.847e–01	0.656	2.760e+01^3 –	1.246e+01	0.965
BGA	2.715e+01^3	2.728e+01	0.731	2.242e+01^2 +	4.020e+00	0.800	2.059e+01^1 +	1.141e+01	0.547
BLX	2.570e+01^2	1.163e+00	0.001	2.484e+01^1 +	3.218e–01	0.914	2.639e+01^3 –	1.431e+01	0.992
Flat	2.655e+01^3	1.192e–01	0.000	2.521e+01^2 +	2.264e–01	0.001	2.371e+01^1 +	2.067e+00	0.000

Table 3 (Continued)

Crossover	Mean	St. Dev.	α^{X}_{XL2VP}	Mean	St. Dev.	α^{XL2VP}_{XL1VP}	Mean	St. Dev.	α^{XL1VP}_{X}
	X (Original)			XL2VP			XL1VP		
f_{Ras}									
SBX	1.479e+01^{3}	4.003e+00	0.000	1.895e–15^{1}	+1.038e–14	0.000	3.032e–14^{2}	+2.884e–14	0.000
Fuzzy	2.365e+01^{3}	6.076e+00	0.000	8.955e–01^{2}	+9.547e–01	0.000	2.084e–14^{1}	+2.786e–14	0.000
Arithmetical	1.350e+01^{3}	3.394e+00	0.000	2.686e+00^{2}	+1.614e+400	0.000	3.317e–02^{1}	+1.817e–01	0.000
BGA	3.648e–01^{2}	4.877e–01	0.962	4.315e–01^{3}	–6.764e–01	0.005	9.633e–12^{1}	+6.210e–12	0.001
BLX	2.643e+401^{3}	6.777e+00	0.000	2.255e+00^{2}	+1.651e+00	0.000	2.274e–14^{1}	+3.202e–14	0.000
Flat	1.489e+01^{3}	4.321e+00	0.000	2.653e+00^{2}	+1.575e+00	0.000	4.926e–14^{1}	+4.151e–14	0.000
f_{Sch}									
SBX	9.370e+02^{3}	2.990e+02	0.000	1.380e+02^{2}	+1.527e+02	0.000	3.818e–04^{1}	+9.230e–13	0.000
Fuzzy	1.408e+02^{2}	1.746e+02	0.000	3.325e+02^{3}	–1.864e+02	0.000	3.818e–04^{1}	+8.478e–13	0.000
Arithmetical	3.116e+03^{3}	4.235e+02	0.000	7.816e+02^{2}	+2.380e+02	0.000	3.948e+00^{1}	+2.162e+01	0.000
BGA	3.447e+01^{2}	6.142e+01	0.007	9.552e+01^{3}	–8.550e+01	0.000	3.818e–04^{1}	+4.565e–11	0.014
BLX	1.598e+03^{3}	3.507e+02	0.000	5.173e+02^{2}	+2.335e+02	0.000	3.818e–04^{1}	+9.250e–13	0.000
Flat	2.248e+03^{3}	4.955e+02	0.000	7.366e+02^{2}	+2.591e+02	0.000	3.818e–04^{1}	+1.105e–12	0.000

Table 4 Average value and standard deviation of the 30 runs for each function with respect to the original crossover, X, and their corresponding alternatives XL2VP and XL1VP. Columns $\alpha^{\mathrm{X}}_{\mathrm{XL2VP}}$, $\alpha^{\mathrm{XL1VP}}_{\mathrm{X}}$ and $\alpha^{\mathrm{XL2VP}}_{\mathrm{XL1VP}}$ show the p-values of the multiple comparison tests between the average fitness of the original crossover X and XL2VP, the original crossover X and XL1VP, and between XL2VP and XL1VP. Values are marked with "*" when the Levene test suggested a Bonferroni test. In other case a Tamhane test is used. "+" and "−" show the results of the sign test, "+" meaning better and "−" worse

Crossover	Mean	St. Dev.	$\alpha^{\mathrm{X}}_{\mathrm{XL2VP}}$	Mean	St. Dev.	$\alpha^{\mathrm{XL2VP}}_{\mathrm{XL1VP}}$	Mean	St. Dev.	$\alpha^{\mathrm{XL1VP}}_{\mathrm{X}}$
	X (Original)			XL2VP			XL1VP		
f_{Wei}									
SBX	7.116e–02^3	2.727e–01	0.415	1.269e–06^2	+ 9.108e–07	0.000	2.503e–07^1	+ 1.373e–07	0.415
Fuzzy	2.448e–05^3	1.355e–05	0.000	3.929e–06^2	+ 3.714e–06	0.000	4.602e–07^1	+ 1.793e–07	0.000
Arithmetical	4.633e–04^3	3.945e–04	0.000	2.652e–05^2	+ 1.494e–05	0.000	7.491e–06^1	+ 6.443e–06	0.000
BGA	1.227e–03^2	5.290e–04	0.000	2.190e–03^3	– 6.813e–04	0.000	3.580e–04^1	+ 1.195e–04	0.000
BLX	3.210e–05^3	6.658e–06	0.000	7.169e–06^2	+ 4.335e–06	0.000	1.702e–06^1	+ 2.293e–06	0.000
Flat	2.964e–04^3	1.599e–04	0.000	1.895e–05^2	+ 1.056e–05	0.000	5.950e–06^1	+ 3.957e–06	0.000
f_{Schaf}									
SBX	4.756e + 00^3	8.465e–01	0.000	9.762e–01^2	+ 4.262e–01	0.879	9.066e–01^1	+ 3.753e–01	0.000
Fuzzy	7.475e + 00^3	9.761e–01	0.000	2.058e + 00^2	+ 7.777e–01	0.000	8.153e–01^1	+ 4.075e–01	0.000
Arithmetical	4.671e + 00^3	1.143e + 00	0.000	3.489e + 00^2	+ 8.008e–01	0.000	8.363e–01^1	+ 2.898e–01	0.000
BGA	9.612e–01^2	4.573e–01	1.000*	9.891e–01^3	– 3.845e–01	0.015*	6.969e–01^1	+ 3.225e–01	0.032*
BLX	3.711e + 00^3	1.171e + 00	0.092	3.559e + 00^2	+ 7.808e–01	0.000	8.416e–01^1	+ 4.130e–01	0.000
Flat	5.247e + 00^3	1.027e + 00	0.000	3.285e + 00^2	+ 8.541e–01	0.000	8.045e–01^1	+ 3.437e–01	0.000
f_{Ack}									
SBX	9.672e–06^3	3.598e–06	0.000	6.910e–09^2	+ 2.256e–09	0.000	4.177e–09^1	+ 1.367e–09	0.000
Fuzzy	1.448e–07^3	6.210e–08	0.000	6.851e–09^1	+ 1.894e–09	0.719	7.294e–09^2	+ 1.698e–09	0.000
Arithmetical	1.971e–07^3	6.188e–08	0.000	1.196e–08^1	+ 2.993e–09	0.003	1.643e–08^2	+ 6.254e–09	0.000
BGA	2.923e–06^2	7.226e–07	0.071	3.636e–06^3	– 1.510e–06	0.000	8.210e–07^1	+ 2.728e–07	0.000
BLX	6.425e–08^3	1.524e–08	0.000	6.952e–09^1	+ 2.402e–09	0.000	1.044e–08^2	+ 3.565e–09	0.000
Flat	1.818e–07^3	6.079e–08	0.000	1.326e–08^1	+ 3.182e–09	0.033	1.635e–08^2	+ 5.523e–09	0.000

Table 4 (Continued)

Crossover	Mean	St. Dev.	$\alpha^{\mathrm{X}}_{\mathrm{XL2VP}}$	Mean	St. Dev.	$\alpha^{\mathrm{XL2VP}}_{\mathrm{XL1VP}}$	Mean	St. Dev.	$\alpha^{\mathrm{XL1VP}}_{\mathrm{X}}$
	X (Original)			XL2VP			XL1VP		
f_{Gri}									
SBX	2.865e–02^{3}	3.442e–02	0.472	1.842e–02^{2} +	2.473e–02	0.114	8.043e–03^{1} +	1.003e–02	0.010
Fuzzy	1.971e–03^{1}	4.815e–03	0.001	1.483e–02^{3} –	1.639e–02	0.999	1.434e–02^{2} –	1.780e–02	0.003
Arithmetical	4.841e–03^{1}	8.510e–03	0.074	1.050e–02^{2} –	1.048e–02	0.042	2.607e–02^{3} –	3.134e–02	0.003
BGA	5.454e–02^{3}	4.207e–02	0.000	1.827e–02^{1} +	2.405e–02	0.975	2.055e–02^{2} +	2.300e–02	0.001
BLX	1.479e–03^{1}	3.861e–03	0.003	1.090e–02^{2} –	1.369e–02	0.018	3.008e–02^{3} –	3.332e–02	0.000
Flat	7.534e–03^{1}	1.305e–02	0.810	1.041e–02^{2} –	1.464e–02	0.027	2.303e–02^{3} –	2.094e–02	0.004
f_{Fle}									
SBX	3.182e + 04^{3}	2.062e + 04	0.004	1.650e+04^{2} +	1.366e+04	0.496	1.230e+04^{1} +	1.163e+04	0.000
Fuzzy	1.482e + 04^{2}	1.318e + 04	1.000*	1.671e+04^{3} –	1.520e+04	0.418*	1.177e+04^{1} +	9.374e+03	0.000*
Arithmetical	8.589e + 03^{2}	9.971e + 03	1.000*	1.067e+04^{3} –	1.135e+04	0.979*	8.173e+03^{1} +	7.785e+03	1.000*
BGA	1.847e + 04^{3}	1.289e + 04	1.000*	1.614e+04^{2} +	1.455e+04	0.427*	1.132e+04^{1} +	9.882e+03	0.092*
BLX	5.193e + 03^{1}	5.756e + 03	0.029	1.273e+04^{3} –	1.408e+04	0.924	1.095e+04^{2} –	1.016e+04	0.029
Flat	8.580e + 03^{1}	5.539e + 03	0.227	1.435e+04^{3} –	1.678e+04	0.998	1.373e+04^{2} –	1.542e+04	0.255
f_{Lan}									
SBX	–1.861e–01^{2}	8.004e–02	0.005	–2.960e–01^{1} +	1.602e–01	0.000	–1.407e–01^{3} –	6.111e–02	0.049
Fuzzy	–9.978e–02^{3}	3.549e–02	0.000	–2.670e–01^{1} +	1.490e–01	0.084	–1.905e–01^{2} +	1.123e–01	0.001
Arithmetical	–2.247e–05^{3}	1.023e–04	0.000	–2.209e–01^{1} +	1.243e–01	0.928	–2.033e–01^{2} +	1.230e–01	0.000
BGA	–2.082e–01^{2}	8.647e–02	0.015	–3.011e–01^{1} +	1.500e–01	0.000	–1.701e–01^{3} –	8.025e–02	0.227
BLX	–2.729e–02^{3}	1.914e–02	0.000	–2.780e–01^{1} +	1.446e–01	0.045	–1.930e–01^{2} +	1.162e–01	0.000
Flat	–5.794e–05^{3}	1.229e–04	0.000	–2.344e–01^{1} +	1.218e–01	0.784	–2.084e–01^{2} +	1.156e–01	0.000

Table 5 Sign test result for each crossover operator

Crossover	XL2VP				XL1VP			
	Win	Draw	Loss	Sign Test	Win	Draw	Loss	Sign Test
SBX	11	0	0	0.000	10	0	1	0.012
Fuzzy	8	0	3	0.227	9	0	2	0.065
Arithmetical	9	0	2	0.065	9	0	2	0.065
BGA	4	0	7	0.549	9	0	2	0.065
BLX	9	0	2	0.065	7	0	4	0.549
Flat	9	0	2	0.065	9	0	2	0.065

benchmarks. For recent results on the proposed problems the reader is referred to (Affenzeller and Wagner 2004; Auger and Hansen 2005).

The analysis of the results of this work aims at two objectives: to determine whether given a crossover X, its alternatives XL2VP and XL1VP are able to outperform it, and study its influence on the performance of the RCGA in function of the features of the problem to solve.

From the results of Tables 3 and 4, where the average and standard deviations of best individuals are shown, we have performed a sign test (Hollander and Wolfe 1973) over the win/draw/loss record of each original crossover X and its alternatives XL2VP and XL1VP. The results of these sign tests are shown in Table 5. With a confidence level of 90%, SBX, Arithmetical and Flat crossover are improved with both new strategies. For Fuzzy crossover the improvement is significant for FuzzyL1VP. For BLX the improvement is significant for BLXL2VP. Only for BGA the alternative XL2VP does not improve the performance of the operator. However, the results obtained with BGAL1VP are significantly better than those obtained with the original BGA. As a summary, the best alternative is XL1VP that significantly improves all the operators, with the exception of BLX.

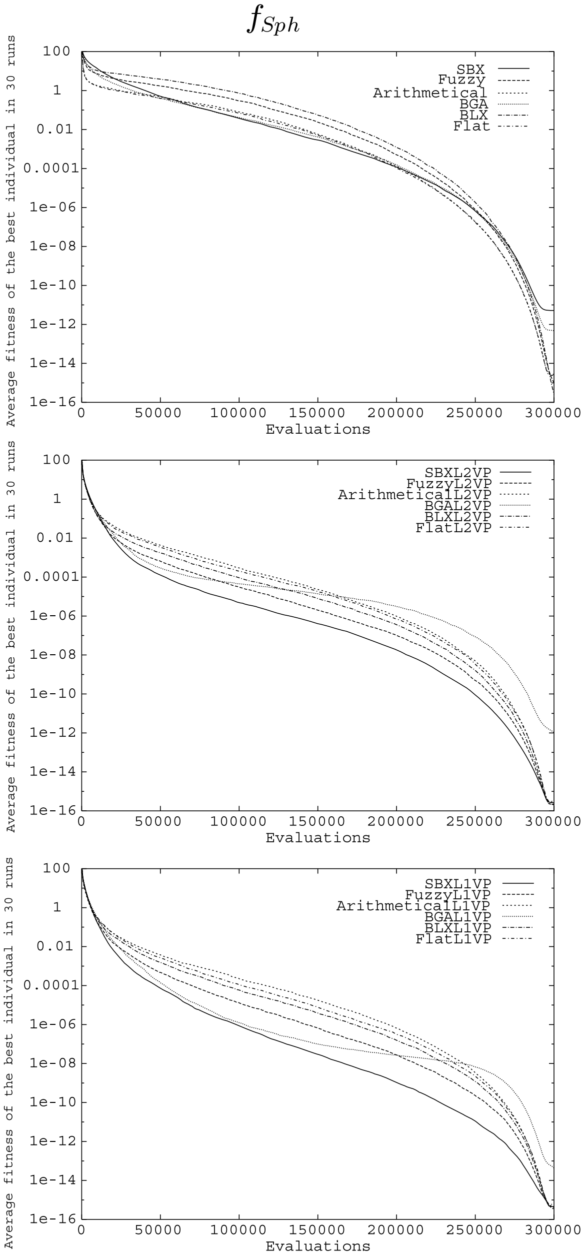
These results show that the use of virtual parents involves, as a general rule, an improvement on the performance of the RCGA, especially when XL1VP is used. Nevertheless, a further analysis, taking into account the type of problem posed by each function, is recommendable to test the validity of that statement. So, for each function we have performed a multiple comparison test between the original crossover and its two alternatives. First, we carry out a Levene test (Miller 1996; Levene 1960) with a confidence level of 95% for evaluating the equality of variances. If the hypothesis that the variances are equal is accepted, we perform a Bonferroni test (Miller 1996) for ranking the means. If the test of Levene results in rejecting the equality of covariance matrixes, we perform a Tamhane test (Tamhane and Dunlop 2000).

Tables 3 and 4 show the results obtained following the above methodology. The superscript of the average fitness shows the rank of the result with respect to the other methods. Columns $\alpha^{\mathrm{X}}_{\mathrm{XL2VP}}$, $\alpha^{\mathrm{XL1VP}}_{\mathrm{X}}$ and $\alpha^{\mathrm{XL2VP}}_{\mathrm{XL1VP}}$ show the significant level of the differences between the average fitness of the original crossover and XL2VP, the original crossover and XL1VP, and between XL2VP and XL1VP. The values are marked with an "*" when the Levene test suggested a Bonferroni test. Additionally, for determining the influence on the convergence speed of the crossover, Figs. 9, 10,

Table 6 Average value and standard deviation of the 30 runs for each function with CIXL2, CIXL1, and the best crossover of Tables 3 and 4

Function	Mean	St. Dev.	Mean	St. Dev.		Mean	St. Dev.
	CIXL2		CIXL1			Best	
f_{Sph}	1.998e–16	6.775e–17	1.858e–15	7.614e–16	BLXL2VP	1.998e–16	6.775e–17
f_{SchDS}	1.305e–02	1.303e–02	3.705e–02	3.452e–02	BLXL2VP	5.233e–03	4.882e–03
f_{Ros}	2.463e+01	1.330e+00	4.051e+01	2.813e+01	BGAL1VP	2.059e+01	1.141e+01
f_{Ras}	7.164e+00	2.579e+00	2.653e–01	5.804e–01	SBXL2VP	1.895e–15	1.038e–14
f_{Sch}	2.844e+03	4.168e+02	2.098e+02	1.278e+02	FuzzyL1VP	3.818e–04	8.478e–13
f_{Wei}	2.907e–05	3.252e–05	7.077e–05	3.221e–05	SBXVL1VP	2.503e–07	1.373e–07
f_{Schaf}	5.400e+00	9.647e–01	8.316e–01	3.449e–01	BGAL1VP	6.969e–01	3.225e–01
f_{Ack}	5.188e–09	2.883e–09	9.950e–08	3.738e–08	SBXL1VP	4.177e–09	1.367e–09
f_{Gri}	1.285e–02	1.801e–02	4.466e–02	3.168e–02	BLX	1.479e–03	3.861e–03
f_{Fle}	1.527e+04	1.362e+04	1.637e+04	1.595e+04	SBX	3.182e+04	2.062e+04
f_{Lan}	–2.803e–01	1.350e–01	–1.669e–01	8.825e–02	BLXL2VP	–3.011e–01	1.500e–01

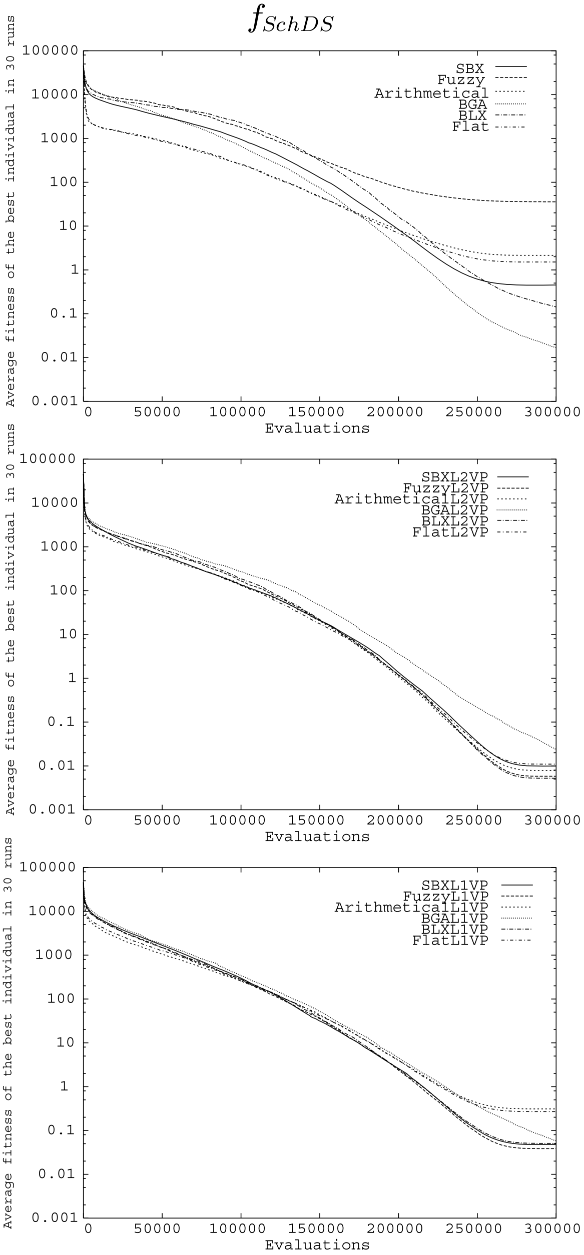
Fig. 9 Evolution of the average fitness of the best individual in 30 runs, in logarithmic scale, using original crossovers X and their alternatives XL2VP and XL1VP for functions f_{Sph}



11, 12, 13 and 14 show the evolution of the average fitness of the best individuals over 30 runs of the algorithm, in logarithmic scale.

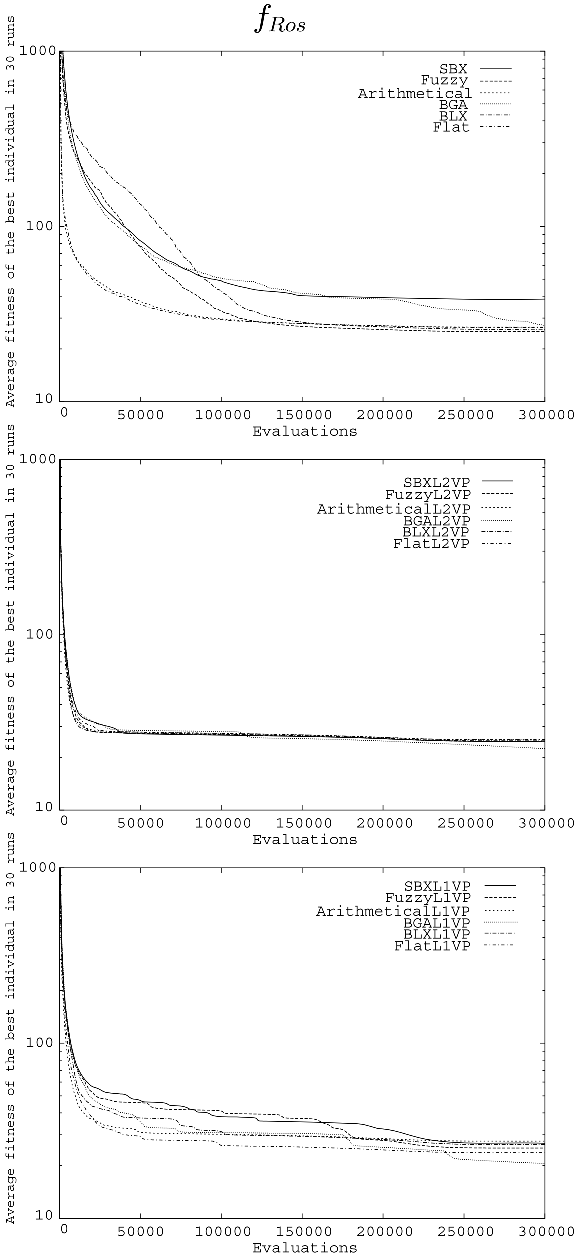
For Sphere function, the fact that it is easy to optimise and the fitness behaves as a singular random variable with sample variance near 0, prevents performing the comparison test for several crossovers. However, the best results are obtained using XL2VP, except for BGA for which the best alternative is BGAL1VP. The behaviour

Fig. 10 Evolution of the average fitness of the best individual in 30 runs, in logarithmic scale, using original crossovers X and their alternatives XL2VP and XL1VP for functions f_{SchDS} and f_{Ros}



of f_{SchDS} is similar. The significantly best results are obtained, at a confidence level of 95%, with, in that order, XL2VP, XL1VP and the original crossover, except for BGA where the best results are obtained with the original crossover but without significant differences with BGAL2VP. For f_{Ros} for 4 of the 6 operators the best alternative is XL2VP. For BGA, BGAL2VP is the second best operator after BGAL1VP, but without significant differences. The same happens for Fuzzy crossover, but in this

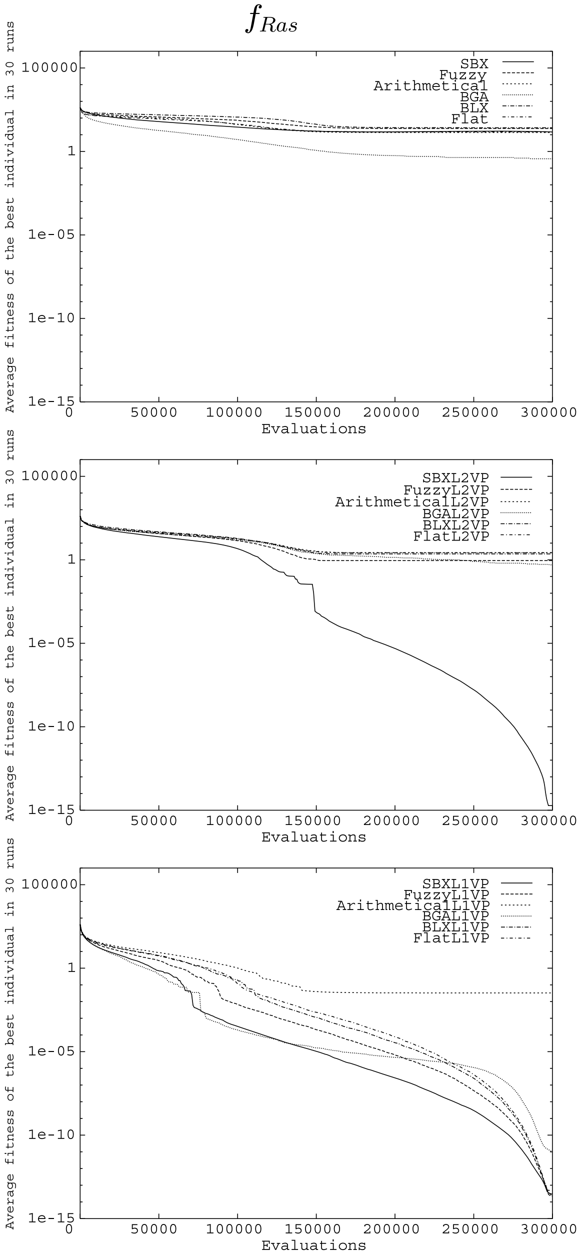
Fig. 10 (Continued)



case the differences are significant. Figs. 9 and 10 show that for these three functions the use of virtual parents makes the convergence faster.

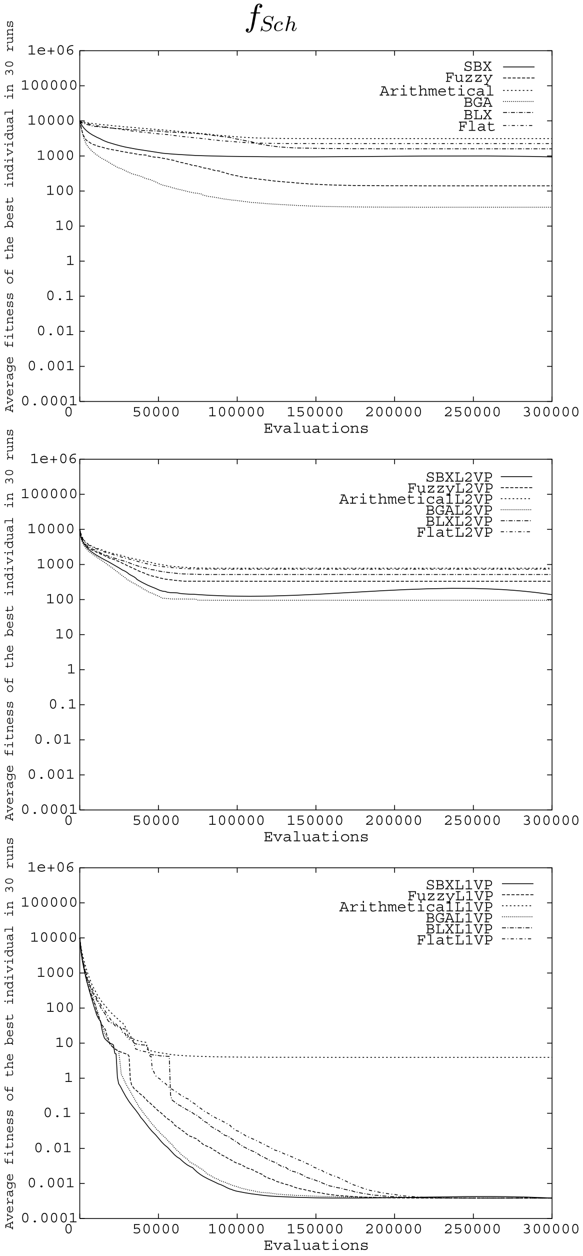
If we take into account that in unimodal functions the probability that the best individuals follow a normal distribution is high, it is reasonable that the performance of XL2VP would be better than the performance of XL1VP. That is the case for f_{Sph}, f_{SchDS} and f_{Ros}.

Fig. 11 Evolution of the average fitness of the best individual in 30 runs, in logarithmic scale, using original crossovers X and their alternatives XL2VP and XL1VP for functions f_{Ras} and f_{Sch}



For multimodal functions, both separable, f_{Ras} and f_{Sch}, and non-separable, f_{Wei} and f_{Schaf}, XL1VP is significantly the best, except for f_{Ras} where SBXL2VP performs significantly better than SBXL1VP (although both perform better than SBX). Figures 11 and 12 show that the use of virtual parents not only involves a clear improvement in the convergence but also a better final solution of the RCGA. The results

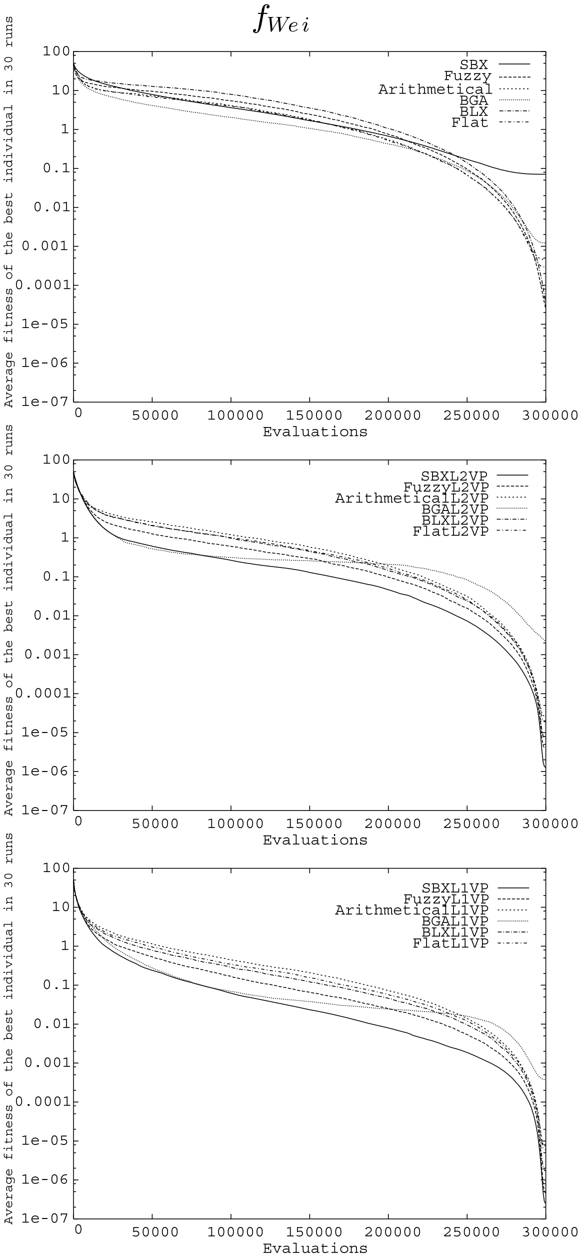
Fig. 11 (Continued)



show that the independence from the distribution of the best individuals of XL1VP is an advantage when dealing with multimodal functions.

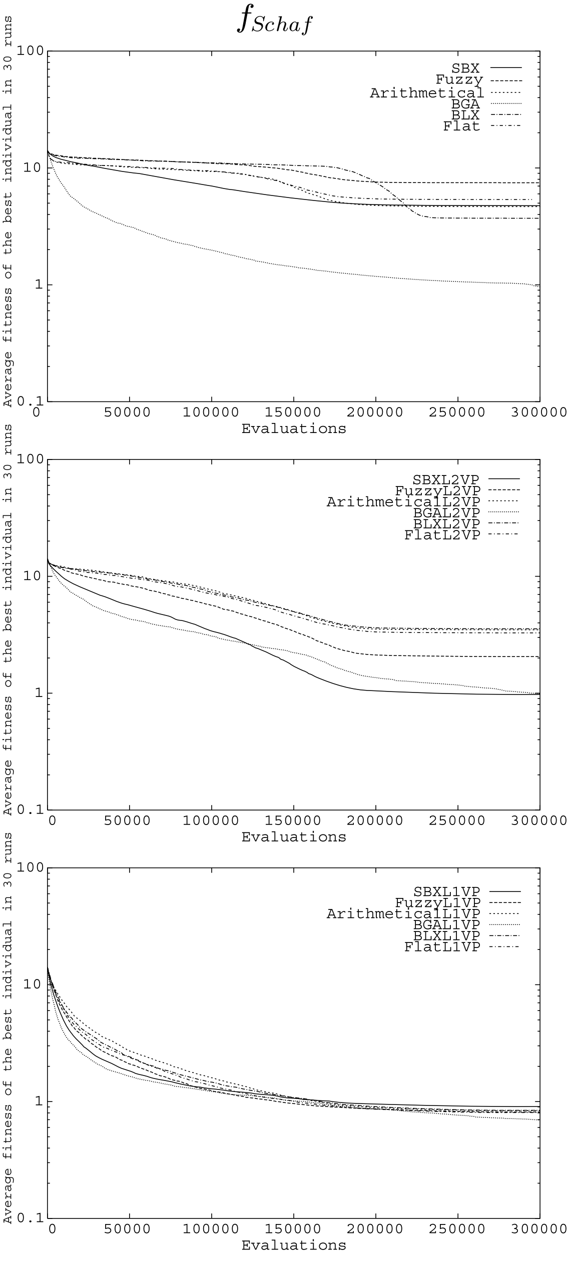
For f_{Ack} and f_{Gri}, although they are also multimodal and non-separable, the regularity with which the maxima are distributed makes their study more difficult. For f_{Ack} the use of virtual parents involves a significant improvement of the results with respect to the original operator but whether alternative, XL2VP or XL1VP, achieves

Fig. 12 Evolution of the average fitness of the best individual in 30 runs, in logarithmic scale, using original crossover X and their alternatives XL2VP and XL1VP for functions f_{Wei} and f_{Schaf}



better results depends on the basic crossover. Figure 13 shows that both methods, XL2VP and XL1VP, improve the convergence of the RCGA. For f_{Gri} the use of virtual parents only improves the results for SBX and BGA crossovers. Figure 13 shows that XL2VP and XL1VP converge faster until generation 200,000, however from that point a change in the slope of the curve makes four of the original operators converge

Fig. 12 (Continued)



to better solutions. This behaviour of f_{Gri} might mean the necessity of using more individuals to localise the most promising search region.

Due to their complexity, f_{Fle} and f_{Lan} functions are challenging for any algorithm. For f_{Fle} XL1VP involves an improvement for four of the original crossover, although this improvement is not always significant. For f_{Lan} the application of XL2VP is always the best performing one and is a significant improvement with respect to the

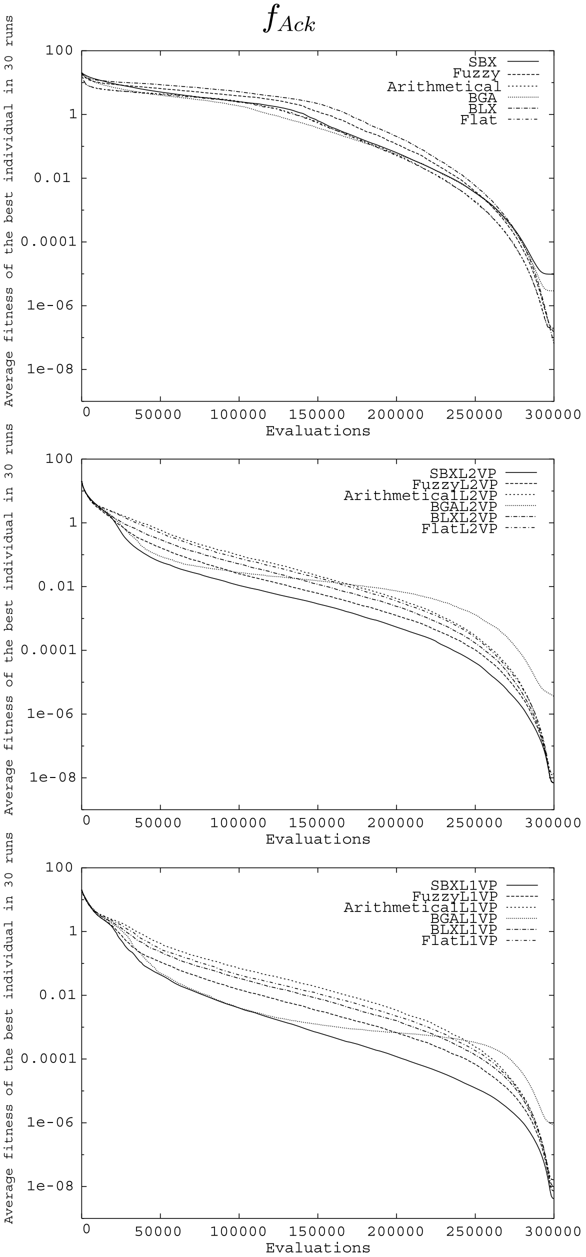
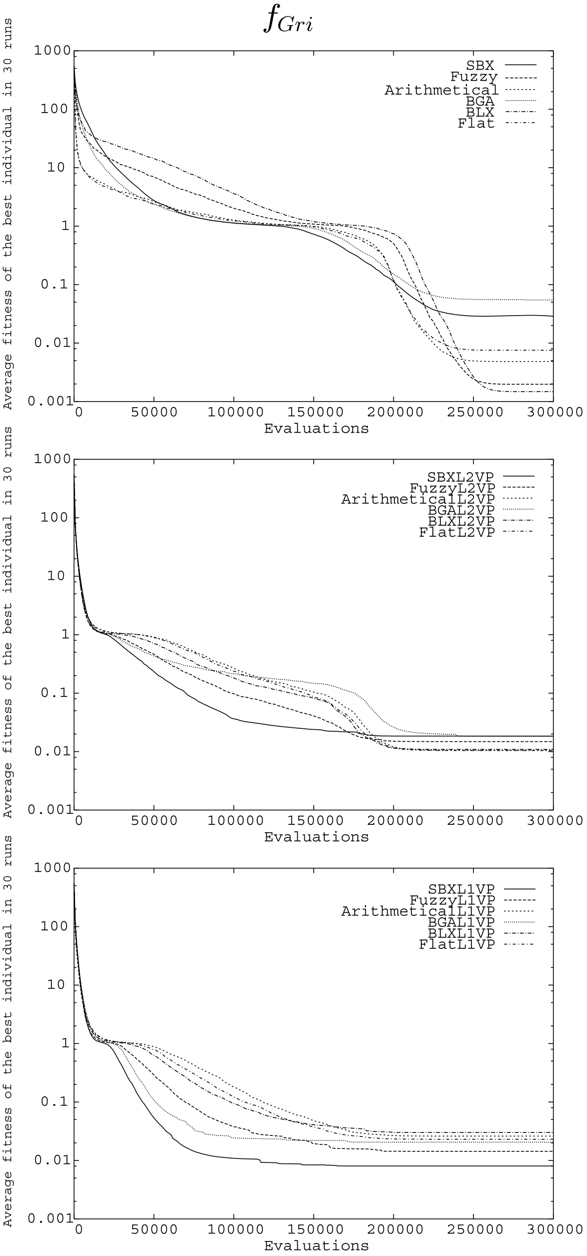


Fig. 13 Evolution of the average fitness of the best individual in 30 runs, in logarithmic scale, using original crossover X and their alternatives XL2VP and XL1VP for functions f_{Ack} and f_{Gri}

original operator. Figure 14 shows that the use of virtual parents improves the convergence speed. Due to the difficulties all algorithms have on these two functions, it is not feasible to extract generalisable conclusions from these results.

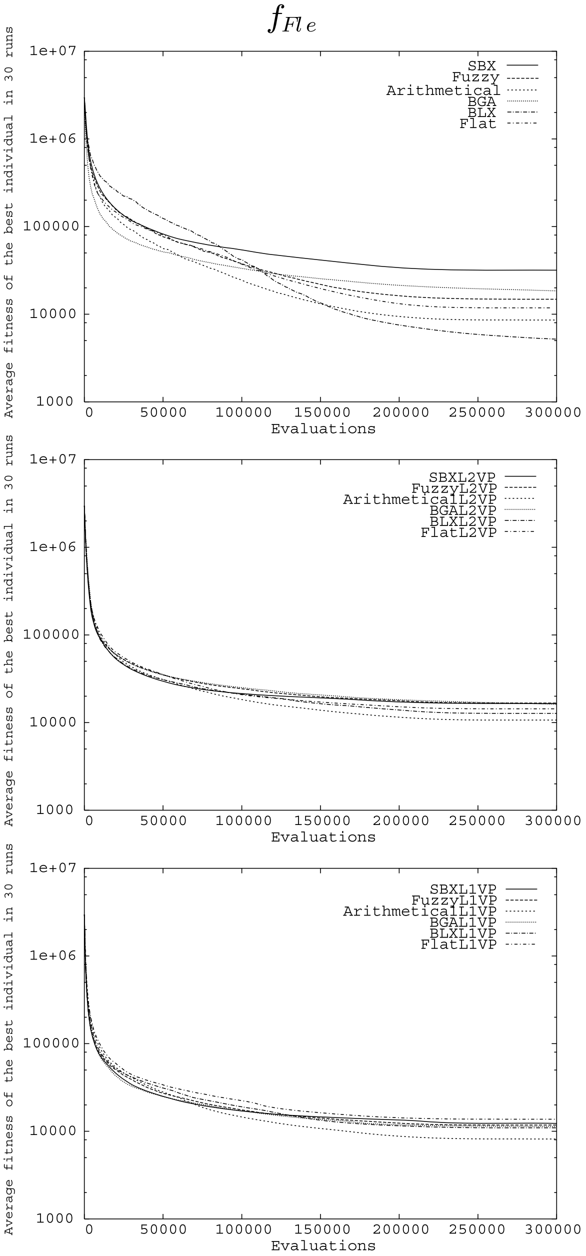
Table 6 shows a comparison between the results obtained using CIXL1 and CIXL2 and the best crossover of the previous analysis. In 9 out of 11 functions, the best result is obtained using a crossover that implements the proposed philosophy, for

Fig. 13 (Continued)



the other 2 functions the best results are obtained with CIXL2 for one of them and BLX for the other. Hence, the proposed methodology is advantageous not only to the original operators but also it is able to improve the original implementation proposed in CIXL1 and CIXL2.

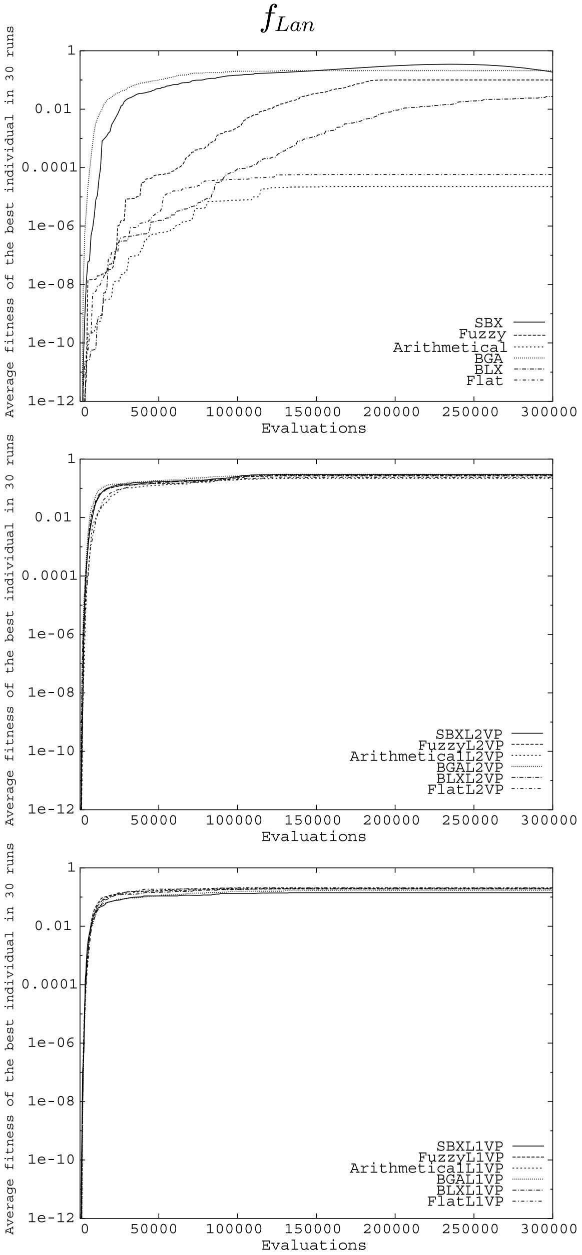
Fig. 14 Evolution of the average fitness of the best individual in 30 runs, in logarithmic scale, using original crossover X and their alternatives XL2VP and XL1VP for functions f_{Fle} and f_{Lan}



7 Conclusions

In this work we have proposed a new strategy for crossover based on the use of three virtual parents created from the features of localisation and dispersion of the best individuals of the population. This strategy can be incorporated into any operator, and in this work we present versions of SBX, Fuzzy, Arithmetical, BGA, BLX and

Fig. 14 (Continued)



Flat operators using this strategy. For each operator we have defined two alternatives: XL2VP and XL1VP. For XL2VP the estimators of localisation and dispersion assume a normal distribution of the individuals, and XL1VP is a non-parametric method.

The reported results show that either XL2VP or XL1VP improve, most of the time, the results using the original crossover. As a general rule, the high probability of the best individuals of the population of following a normal distribution in unimodal problems makes XL2VP more suitable for this kind of problem. On the other hand, in multimodal problems where it is more difficult to make any assumption about the distribution of the genes of the best individuals, using XL1VP is more advisable, as it uses a non-parametric confidence interval, since the confidence coefficient is determined from the binomial distribution without making any assumption of the underlying gene distribution. Nevertheless, the experiments show that the original crossover and the regularity of the distribution of optima may have an effect on the performance of both methods, XL2VP and XL1VP.

Although using virtual parents that drive the population towards the most promising search regions might suggest a greater likelihood of being trapped in local minima, especially in multimodal functions, this circumstance has not been observed in the experiments reported here. The crossover strategy proposed allows the generation of descendants near the parents from the population only if the fitness of those parents is better than the fitness of the virtual parents. In this way the shifting of the population towards the region where the virtual parents are placed is prevented unless those virtual parents mean a significant improvement in the fitness.

Additional research is very interesting in some areas of this work, especially the selection of the individuals from the population to make up the virtual parents, taking into account the problem to be solved, to improve the identification of the most promising regions of the search space. In such a case it will be necessary to study the possible noise induced by the selection method in the distribution of the genes of the best individuals.

In multimodal, nonseparable with many chaotically scattered optima functions it might be interesting to divide the population into different clusters and obtain different virtual individuals from each cluster. In this way, we can implement a multidirectional search strategy.

Acknowledgements This work has been financed in part by the projects TIC2002-04036-C05-02 and TIN2005-08386-C05-02 of the Spanish Inter-Ministerial Commission of Science and Technology (CICYT) and FEDER funds.

References

Ackley, D.H.: An empirical study of bit vector function optimization. In: Genetic Algorithms and Simulated Annealing, pp. 170–215. Kaufmann, San Mateo (1987)

Affenzeller, M., Wagner, S.: A self-adaptive model for selective pressure handling within the theory of genetic algorithms. In: Computer Aided Systems Theory: EUROCAST 2003. Lecture Notes in Computer Science, vol. 2809, pp. 384–393. Springer, Berlin (2003)

Affenzeller, M., Wagner, S.: Sasegasa: A new generic parallel evolutionary algorithm for achieving highest quality results. J. Heuristics **10**, 239–263 (2004). Special Issue on New Advances on Parallel Meta-Heuristics for Complex Problems

Antonisse, J.: A new interpretation of schema notation that overturns the binary encoding constraint. In: Schaffer, J.D. (ed.) Third International Conference on Genetic Algorithms, pp. 86–91. Kaufmann, San Mateo (1989)

Auger, A., Hansen, N.: A restart cma evolution strategy with increasing population size. In: IEEE Congress on Evolutionary Computation (CEC'05), vol. 2, pp. 1769–1776. IEEE Press, Napier University, Edinburgh, UK (2005)

Bebis, G., Georgiopoulos, M., Kasparis, T.: Coupling weight elimination with genetic algorithms to reduce network size and preserve generalization. Neurocomputing **17**, 167–194 (1997)

Bersini, H., Dorigo, M., Langerman, S., Seront, G., Gambardella, L.M.: Results of the first international contest on evolutionary optimisation (1st ICEO). In: Proceedings of IEEE International Conference on Evolutionary Computation, IEEE-EC 96, pp. 611–615. IEEE Press, Nagoya (1996)

Beyer, H.-G., Deb, K.: On self-adapting features in real-parameter evolutionary algorithms. IEEE Trans. Evol. Comput. **5**(3), 250–270 (2001)

Bäck, J.H.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, Oxford (1996)

Bäck, T., Fogel, D., Michalewicz, Z.: Handbook of Evolutionary Computation. Institute of Physics Publishing Ltd/Oxford University Press, Bristol/New York (1997)

Czarn, A., MacNish, C., Vijayan, K., Turlach, B., Gupta, R.: Statistical exploratory analysis of genetic algorithms. IEEE Trans. Evol. Comput. **8**(4) (2004)

De Jong, K.D.: An analysis of the behavior of a class of genetic adaptive systems. Ph.D. thesis, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor (1975)

De Jong, K., Spears, W.: A formal analysis of the role of multi-point crossover in genetic algorithms. Ann. Math. Artif. Intell. **5**(1), 1–26 (1992)

De Jong, M.B., Kosters, W.: Solving 3-SAT using adaptive sampling. In: Poutré, H., van den Herik, J. (eds.) Proceedings of the Tenth Dutch/Belgian Artificial Intelligence Conference, pp. 221–228 (1998)

Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Syst. **9**, 115–148 (1995)

Deb, K., Beyer, H.: Self-adaptive genetic algorithms with simulated binary crossover. Evol. Comput. **9**(2), 195–219 (2001)

Eiben, A.E., Bäck, T.: An empirical investigation of multi-parent recombination operators in evolution strategies. Evol. Comput. **5**(3), 347–365 (1997)

Eiben, A., van der Hauw, J., van Hemert, J.: Graph coloring with adaptive evolutionary algorithms. J. Heuristics **4**(1), 25–46 (1998)

Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In: Whitley, L.D. (ed.) Foundation of Genetic Algorithms 2, 187C3.3.7:1–C3.3.7:8.–202, Kaufmann, San Mateo (1993)

Eshelman, L.J., Caruana, A., Schaffer, J.D.: Biases in the crossover landscape. In: Schaffer, J.D. (ed.) Third International Conference on Genetic Algorithms, pp. 86–91. Kaufmann, San Mateo (1989)

Fletcher, R., Powell, M.J.D.: A rapidly convergent descent method for minimization. Comput. J. **6**, 163–168 (1963)

Friedman, J.H., An overview of predictive learning and function approximation. In: Cherkassky, V., Friedman, J.H., Wechsler, H. (eds.) From Statistics to Neural Networks, Theory and Pattern Recognition Applications. NATO ASI Series F, vol. 136, pp. 1–61. Springer, Berlin (1994)

García-Pedrajas, N., Hervás-Martínez, C., Ortiz-Boyer, D.: Cooperative coevolution of artificial neural network ensembles for pattern classification. IEEE Trans. Evol. Comput. **9**(3), 271–302 (2005a)

García-Pedrajas, N., Ortiz-Boyer, D., Hervas-Martínez, C.: An alternative approach for neural network evolution with a genetic algorithm: Crossover by combinatorial optimization. Neural Netw. **19**, 514–528 (2005b)

Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison–Wesley, New York (1989)

Goldberg, D.E.: Real-coded genetic algorithms, virtual alphabets, and blocking. Complex Syst. **5**, 139–167 (1991)

Hajela, P.: Soft computing in multidisciplinary aerospace design-new direction for research. Prog. Aerosp. Sci. **38**(1), 1–21 (2002)

Herrera, F., Lozano, M.: Gradual distributed real-coded genetic algorithms. IEEE Trans. Evol. Comput. **4**(1), 43–63 (2000)

Herrera, F., Lozano, M., Verdegay, J.L.: Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. In: Artificial Inteligence Review, pp. 265–319. Kluwer Academic, Netherlands (1998)

Herrera, F., Lozano, M., Sánchez, A.M.: A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. Int. J. Intell. Syst. **18**, 309–338 (2003)

Hervás-Martínez, C., Ortiz-Boyer, D.: Analizing the statistical features of CIXL2 crossover offspring. Soft Comput. **9**(4), 270–279 (2005)

Hervás-Martínez, C., García-Pedrajas, N., Ortiz-Boyer, D.: Confidence interval based crossover using a L1 norm localization estimator for real-coded genetic algorithms. In: Benitez, J., Cordón, O., Hoffmann, F., Roy, R. (eds.) Advances in Soft Computing, pp. 297–305. Springer, Berlin (2003)

Hettmansperger, T.P., McKean, J.W.: Robust Nonparametric Statistical Methods. Arnold John/Wiley, London (1998)

Holland, J.H.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor (1975)

Hollander, M., Wolfe, D.: Nonparametric Statistical Methods. Wiley, New York (1973)

Kendall, M., Stuart, S.: The Advanced Theory of Statistics, vol. 1. Charles GriOEn & Company (1977)

Kita, H.: A comparison study of self-adaptation in evolution strategies and real-code genetic algorithms. Evol. Comput. **9**(2), 223–241 (2001)

Koza, J.R.: Genetic Programming. MIT Press, Cambridge (1992)

Levene, H.: Essays in Honor of Harold Hotelling. In: Contributions to Probability and Statistics, pp. 278–292. Stanford University Press, Stanford (1960)

Liepins, G.E., Vose, M.D.: Characterizing crossover in genetic algorithms. Ann. Math. Artif. Intell. **5**, 27–34 (1992)

McNeils, J.D.P.: Approximating and simulating the stochastic growth model: Parameterized expectations, neural networks, and the genetic algorithm. J. Econ. Dyn. Control **25**(9), 1273–1303 (2001)

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

Miller, R.G.: Beyond ANOVA, Basics of Applied Statistics, 2nd edn. Chapman & Hall, London (1996)

Mühlebein, H., Schlierkamp-Voosen, D.: Predictive models for breeder genetic algorithm i. continuous parameter optimization. Evol. Comput. **1**, 25–49 (1993)

Neyman, J.: Outline of a theory of statistical estimation based on the classical theory of probability. Philos. Trans. Roy. Soc. Lond. A **236**, 333–380 (1937)

Ortiz-Boyer, D., Hervás-Martínez, C., García-Pedrajas, N.: Crossover operator effect in function optimization with constraints. In: Merello, J., Adamidis, P., Beyer, H.-G., Fernandez, J.L., Schwefel, H.P. (eds.) The 7th Conference on Parallel Problem Solving from Nature. Lecture Notes in Computer Science, vol. 2439, pp. 184–193. Springer, Granada (2002)

Ortiz-Boyer, D., Hervás-Martínez, C., Muñoz-Pérez, J.: Study of genetic algorithms with crossover based on confidence intervals as an alternative to classic least squares estimation methods for non-linear models. In: Resende, M.G.C., de Sousa, J.P. (eds.) Metaheuristics: Computer Decision-Making, pp. 127–151. Kluwer Academic, Dordrecht (2003)

Ortiz-Boyer, D., Hervás-Martínez, C., García-Pedrajas, N.: Cixl2: A crossover operator for evolutionary algorithms based on population features. J. Artif. Intell. Res. **24**, 1–48 (2005)

Périauz, J., Sefioui, M., Stoufflet, B., Mantel, B., Laporte, E.: Robust genetic algorithm for optimization problems in aerodynamic design. In: Winter, G., Periaux, J., Galan, M., Cuesta, P. (eds.) Genetic Algorithms in Engineering and Computer Science, pp. 370–396. Wiley, New York (1995)

Radcliffe, N.J.: Equivalence class analysis of genetic algorithms. Complex Syst. **2**(5), 183–205 (1991)

Radcliffe, N.J.: Non-linear genetic representations. In: Männer, R., Manderick, B. (eds.) Second International Conference on Parallel Problem Solving from Nature, pp. 259–268. Elsevier, Amsterdam (1992)

Rastrigin, L.A.: Extremal control systems. In: Theoretical Foundations of Engineering Cybernetics Series, vol. 3. Nauka, Moscow (1974)

Rosenbrock, H.H.: An automatic method for finding the greatest or least value of a function. Comput. J. 175–184 (1960)

Roubos, J., van Straten, G., van Boxtel, A.: An evolutionary strategy for fed-batch bioreactor optimization; concepts and performance. J. Biotechnol. **67**(2-3), 173–187 (1999)

Rudolph, G.: Convergence analysis of canonical genetic algorithms. IEEE Trans. Neural Netw. **5**(1), 96–101 (1994)

Schaffer, J., Caruana, R., Eshelman, L., Das, R.: A study of control parameters affecting online performance of genetic algorithms for function optimization. In: Schaffer, J. (ed.) 3rd International Conference on Genetic Algorithms, pp. 51–60. Kaufmann, San Mateo (1989)

Schlierkamp-Voosen, D.: Strategy adaptation by competition. In: Second European Congress on Intelligent Techniques and Soft Computing, pp. 1270–1274 (1994)

Schwefel, H.P.: Numerical Optimization of Computer Models. Wiley, New York (1981)

Schwefel, H.P.: Evolution and Optimum Seeking. Wiley, New York (1995)

Spedicato, E.: Computational experience with quasi-newton algorithms for minimization problems of moderately large size, CISE-N-175, Centro Informazioni Studi Esperienze, Segrate (Milano), Italy (1975)

Syswerda, G.: Uniform crossover in genetic algorithms. In: Schasffer, J. (ed.) 3rd International Conference on Genetic Algorithm, pp. 2–9. Kaufmann, San Mateo (1989)

Tamhane, A.C., Dunlop, D.D.: Statistics and Data Analysis. Prentice Hall, New York (2000)

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

Voigt, H.M., Mühlenbein, H., Cvetkovic, D.: Fuzzy recombination for the breeder genetic algorithms. In: Eshelman, L. (ed.) The 6th International Conference Genetic Algorithms, pp. 104–111. Kaufmann, San Mateo (1995)

Weierstrass, F.: Über continuirlichefunctionen eines reellen arguments die für keinen werth des letzteren einen bestimmter differentialquotienten besitzen. Math. Werke **II**, 71–72 (1872)

Wright, A.: Genetic algorithms for real parameter optimization. In: Rawlin, G.J.E. (ed.) Foundations of Genetic Algorithms 1, pp. 205–218. Kaufmann, San Mateo (1991)

Zhang, B.T., Kim, J.J.: Comparison of selection methods for evolutionary optimization. Evol. Optim. **2**(1), 55–70 (2000)