Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
European Journal of Operational Research 185 (2008) 1088–1113
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Global and local real-coded genetic algorithms based on parent-centric crossover operators

C. García-Martínez [a,*], M. Lozano [b], F. Herrera [b], D. Molina [c], A.M. Sánchez [d]

[a] *Department of Computing and Numerical Analysis, University of Córdoba, 14071 Córdoba, Spain*
[b] *Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain*
[c] *Department of Software Engineering, University of Cádiz, 11002 Cádiz, Spain*
[d] *Department of Software Engineering, University of Granada, 18071 Granada, Spain*

Received 1 June 2005; accepted 1 June 2006
Available online 18 October 2006

Abstract

Parent-centric real-parameter crossover operators create the offspring in the neighbourhood of one of the parents, the female parent. The other parent, the male one, defines the range of the neighbourhood. With the aim of improving the behaviour of these crossover operators, we present three processes that are performed before their application. First, a female and male differentiation process determines the individuals in the population that may become female or/and male parents. Then, two different selection mechanisms choose the female and male parents from each group. In addition, we tackle the election of the most adequate evolution model to take out profit from these parent selection mechanisms. The experimental results confirm that these three processes may enhance the operation of the parent-centric crossover operators.

Keywords: Real-coded genetic algorithms; Steady-state genetic algorithms; Parent selection mechanism; Parent-centric crossover operators; Chromosome differentiation; Hybrid real-coded genetic algorithms

1. Introduction

In the initial formulation of the *Genetic Algorithms* (GAs) (Goldberg, 1989; Holland, 1975), the candidate solutions were coded using the binary alphabet, however, other coding types, such as the real coding, have also been taken into account to deal with the representation of the problem. The *real coding* approach seems to be adequate when tackling optimization problems of parameters with variables in continuous domains (Davis, 1991; Deb, 2001, 2005; Herrera et al., 1998; Michalewicz, 1992). GAs based on real-number representation are called *real-coded* GAs (RCGAs). Over the past few years, many researchers have been paying attention to RCGAs (Chelouah and Siarry, 2000; Herrera et al., 2005; Hervás-Martínez and Ortiz-Boyer, 2005; Someya and Yamamura, 2005; Winter

---

[*] Corresponding author.
E-mail addresses: cgarcia@uco.es (C. García-Martínez), lozano@decsai.ugr.es (M. Lozano), herrera@decsai.ugr.es (F. Herrera), daniel.molina@uca.es (D. Molina), amlopez@ugr.es (A.M. Sánchez).

0377-2217/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2006.06.043

et al., 2005; Yang and Kao, 2000), and recently, there has been an increasing interest in solving real-world optimization problems using these algorithms.

The crossover operator has always been regarded as one of the main search operators in GAs (De Jong and Spears, 1992; Kita, 2001) because it exploits the available information in previous samples to influence future searches. This is why most RCGA research has been focused on developing effective real-parameter crossover operators, and as a result, many different possibilities have been proposed (Deb, 2001; Herrera et al., 1998, 2003). Parent-centric crossover operators (PCCOs) are a family of real-parameter crossover operators that has currently received special attention. They include fuzzy recombination (Voigt et al., 1995), SBX (Deb and Agrawal, 1995), PCX (Deb et al., 2002), XLM (Takahashi and Kobayashi, 2001), vSBX (Ballester and Carter, 2003), PNX (Ballester and Carter, 2004), and PBX (Lozano et al., 2004). In general, these operators use a probability distribution to create offspring in a restricted search space around the region marked by one of the parent, *the female parent*. The range of this probability distribution depends on the distance among the female parent and the other parent involved in the crossover, *the male parent*. Experiments carried out in Deb et al. (2002) have shown that PCCOs arise as a meaningful and efficient way of solving real-parameter optimization problems. Thus, the study of these operators becomes a topic of major interest for RCGA research.

So far, PCCO practitioners have assumed that every chromosome in the population may become either a female parent or a male parent. However, it is very important to emphasize that female and male parents have two differentiated roles:

- female parents *point* to the search areas that will receive sampling points, whereas
- male parents are used to determine the *extent* of these areas.

At this point, it is reasonable to think that some chromosomes may be well-suited to act either as female parents or as male parents. This means that we may make use of different ways to select the female and male parents. Thus, with the aim of improving the behaviour of PCCOs, in this paper, we propose three processes that are performed before their application:

1. *The female and male differentiation* (FMD) *process*. It creates two different groups: (1) G_F with the N_F chromosomes in the population that can be female parents; and (2) G_M with the N_M individuals that can be selected as male parents (N_F and N_M are tuneable parameters). This process may be considered as a *preselection mechanism*.
2. *The selection of the female parent*. We present a new method for the selection of the female parent, the *uniform fertility selection* (UFS), which attempts to assign a fair number of offspring to the chromosomes that visit the population, with the aim of providing a widespread search. In order to do that, it selects, as female parent, the individual in G_F with the lowest number of offspring generated.
3. *The selection of the male parent.* We consider the *negative assortative mating* (NAM) (Fernandes and Rosa, 2001; Matsui, 1999) for the selection of the male parent. From a set of candidate mates (from G_M), this procedure chooses, as male parent, the chromosome with the highest Euclidean distance from the female parent. With this strategy, we force the use of wide probability distributions that favour the creation of offspring very dissimilar to the female parent.

An important feature of the FMD process is that two different types of specialized RCGAs may be obtained by adjusting the N_F and N_M parameters: *global* RCGAs, which may offer *reliability*, and *local* RCGAs, which may provide *accuracy*. In addition, with the aim of producing a robust operation, we propose a *hybrid* RCGA *method* that combines these algorithms. First, it applies a global RCGA, and then, a local RCGA. The best individuals in population of the former become the individuals in the initial population of the last one.

The paper is organized as follows. In Section 2, we introduce relevant issues related to PCCOs and describe the one considered in this paper, the PBX-α operator (Lozano et al., 2004). In Section 3, we describe the basic algorithm step of the *Steady-State Genetic Algorithms* (SSGAs), which is considered as the most adequate model to take out profit from PCCOs (Ballester and Carter, 2003, 2004; Deb et al., 2002). This election was made because SSGAs may supply high selection pressure, which becomes well-suited for the meaningful operation of these operators. In Section 4, we propose the new method, UFS, for the selection of the female parent, and its effectiveness is compared with other parent selection mechanisms presented in the literature that may be used for female selection. In Section 5, we examine the effects produced by the male selection mechanism suggested, NAM, on the operation of PBX-α. We should point out that UFS and NAM are presented and studied before the FMD process in order to facilitate their analysis (the initial experimental study of UFS and NAM (Sections 4 and 5) was made without applying the FMD process). In Section 6, we present the FMD process and carry out experiments to investigate its behaviour when is incorporated in an RCGA along with UFS and NAM. In Section 7, we explain the way that global and local RCGAs may be designed by adjusting the parameters associated with the FMD process (N_F and N_M), and in addition, we make up a *hybrid* RCGA model that combines global and local RCGAs, and compare its performance with other specific *metaheuristics* proposed in the literature for continuous optimization. Finally, we draw some conclusions in Section 8. In Appendix A, we present an abbreviation index of the terms used in the paper. In Appendix B, we provide a detailed discussion about the selection mechanisms and replacement strategies for SSGAs. In Appendix C, we include the features of the test suite used for the experiments. This appendix explains, as well, the way the executed algorithms were started with a *skewed* population not bracketing the global optimum. And, in Appendix D and E, we present the results of the studied algorithms.

2. Parent-centric crossover operators

PCCOs assign more probability to create offspring near the female parent than anywhere in the search space. In particular, they determine the genes of the offspring extracting values from intervals defined in neighbourhoods associated with the genes of the female parent, throughout probability distributions. The ranges of these probability distributions depend on the distance among the genes of the female parent and the genes of the male parent. Examples are fuzzy recombination (Voigt et al., 1995), SBX (Deb and Agrawal, 1995), XLM (Takahashi and Kobayashi, 2001), PCX (Deb et al., 2002), PNX (Ballester and Carter, 2004), and PBX (Lozano et al., 2004).

In this section, we deal with the main aspects of PCCOs. In Section 2.1, we describe the PCCO instance that is used in this work and in Section 2.2, we discuss different advantages of PCCOs and explain why they are like self-adaptive real-parameter mutation operators.

2.1. The PBX-α crossover operator

In this paper, we use the *parent-centric* BLX-α crossover operator (PBX-α) (Lozano et al., 2004), which is described as follows. Let us assume that $X = (x_1 \cdots x_n)$ and $Y = (y_1 \cdots y_n)$ $(x_i, y_i \in [a_i, b_i] \subseteq \Re,\ i = 1 \ldots n)$ are two real-coded chromosomes that have been selected to apply the crossover operator to them. PBX-α generates the offspring $Z = (z_1 \cdots z_n)$, where z_i is a randomly (uniformly) chosen number from the $[l_i, u_i]$ interval, with $l_i = \max\{a_i, x_i - I \cdot \alpha\}$, $u_i = \max\{b_i, x_i + I \cdot \alpha\}$, and $I = |x_i - y_i|$. The parents X and Y, will be named differently:

- X will be called *female parent*, and
- Y will be denominated *male parent*.

The effects of this crossover operator may be observed in Fig. 1. The role of the female parents is to point to the search areas that will receive sampling points. On the other hand, the male parents are used to determine the extent of these areas, because the ranges of the probability distributions used by PBX-α depend on the distance among the female parent and this parent. α determines the spread associated with the probability distributions used to create offspring. It uses to get values from the interval [0.5, 1].

2.2. Advantages of the parent-centric crossover operators

Experiments carried out in Deb et al. (2002) have shown that PCCOs arise as a meaningful and efficient way of solving real-parameter optimization problems. We think that these results are due to those PCCOs combine two advantageous features:

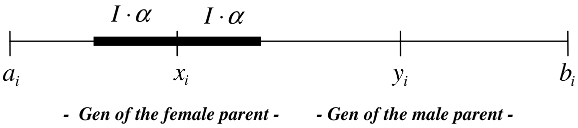

Fig. 1. Effects of the PBX-α operator.

- *PCCOs behave like a mutation operator*. PCCOs generate solutions that are close to the female parent. In this way, they may be seen as a special type of mutation. In fact, it is interesting to highlight that most RCGA models based on PCCOs do not use additional mutation operators (Ballester and Carter, 2003, 2004; Deb et al., 2002).
  On the other hand, two of the most important avenues of research in evolutionary algorithms that use real coding pay attention on mutation as the primal operation to generate novel search points. They are the *evolution strategies* (ESs) (Beyer and Schwefel, 2002; Schwefel, 1995) and the *evolutionary programming* (EP) (Fogel, 1995). They simulate evolution as a phenotypic process, that is, a process emphasizing the behavioural link between parents and offspring, rather than their genetic link. In this way, the emphasis is placed on the use of mutation operators that generate a continuous range of behavioural diversity and keep a strong correlation between the behaviour of the parent and its offspring (Fogel, 1994).
  Deb adopts a similar idea to justify the work of PCCOs (Deb, 2005): since each parent is carefully picked by the selection mechanism, for most real-parameter optimization problems it can be assumed that solutions *close* to these parents are also likely to be very good candidates. From this claim, we may remark an additional outstanding comment: the operation of PCCOs may become particularly promising when they are applied to highly fit individuals. This explains that most RCGAs based on PCCOs appeared in the literature are steady-state GAs, because they may attain higher selection pressure levels than generational GAs (De Jong and Sarma, 1993).
- *PCCOs are self-adaptive crossover operators*. PCCOs define a probability distribution of offspring solutions based on some measure of distance among the parent solutions. If the parents are located closely to each other, the offspring generated by the crossover might be distributed densely around the female parent. On the other hand, if the parents are located far away from each other, then the offspring will be sparsely distributed around it. Therefore, PCCOs may fit their action range depending on the diversity of the population by using specific information held by the parents. In this way, depending on the current level of diversity in the population, they may favour the production of additional diversity (divergence) or the refinement of the solutions (convergence). This behaviour is achieved without requiring an external adaptive mechanism.
  In fact, in the recent past, RCGAs with crossover operators with this feature have been demonstrated to exhibit self-adaptive behaviour similar to that observed in ESs and EP (Deb and Beyer, 2001; Kita, 2001). *Differential Evolution* (Storn and Price, 1997) is another example of an Evolutionary Algorithm that exhibits self-adaptive behaviour. Moreover, Beyer et al. (Beyer and Deb, 2001) argue that a variation operator that harnesses the difference of the parents in the search space is essential for the resulting evolutionary algorithm to exhibit self-adaptive behaviour on the population level.

To sum up, we may conclude that PCCOs may be seen as *self-adaptive real-parameter mutation operators*. Several self-adaptive mutation techniques have been proposed for ESs and EP as well (see Bäck, 1996). However, there exists a clear difference:

- ESs and EP evolve the parameters of this operator, such as *standard deviations*, simultaneously with the decision variables.
- PCCOs calculate implicitly the standard deviations using information about the distribution of the individuals in the population.

Finally, we should point out that since PCCOs work like self-adaptive mutation operators, they are amenable for the design of effective *local search procedures*. In fact, in Lozano et al. (2004), a *crossover hill-climbing* based on PBX-α is proposed as local searcher of a real-coded memetic algorithm.

3. Steady-state genetic algorithms

The generational GA creates new offspring from the members of an old population using the genetic operators and places these individuals in a new population which becomes the old population when the whole new population is created. The *Steady-state GA* (SSGA) is different to the generational model in that there is typically one single new member inserted into the new population at any one time. A *replacement/deletion* strategy defines which member in the current population is forced to perish (or vacate a slot) in order to make room for the new offspring to compete (or, occupy a slot) in the next iteration. SSGAs are overlapping systems, since parents and offspring compete for survival. The basic algorithm step of the SSGA (Syswerda, 1989; Whitley, 1989) considered is shown below. It is adapted for the application of the PBX-α operator.

```
1. Select a female parent.
2. Select a male parent.
3. Create an offspring applying PBX-α on
   the parents.
4. Evaluate the offspring with the fit-
   ness function.
5. Select an individual in the popula-
   tion, which may be replaced by the
   offspring.
6. Decide if this individual will be
   replaced.
```

These steps are repeated until a termination condition is achieved. In step 5, one can choose the *replacement strategy* (e.g., replacement of the worst, the oldest, or a randomly chosen individual). In step 6, one can choose the *replacement condition* (e.g., replacement if the new individual is better or unconditional replacement). In Appendix B, we describe different techniques for the parent selection and the replacement phase of the SSGA.

4. Female selection mechanism: The uniform fertility selection

Many replacement strategies presented for SSGAs (Appendix B.2) may introduce a possible drawback: *some individuals may reside in the population during a long time*. This situation may cause that particular search zones are excessively exploited, at the expense of ignoring other areas showing promise. One way to avoid the risks derived from this problem lies in the limitation of the number of offspring the chromosomes may receive during their lifetime. Different parent selection schemes have been presented to implement this idea (Branke et al., 1999; De Jong and Sarma, 1993; Ghosh et al., 1998).

In this section, we present a new parent selection mechanism, called *uniform fertility selection* (UFS), which restricts, as well, the number of offspring allowed to the chromosomes in the population. UFS takes into account the number of times that a chromosome has been selected as female parent. Then, *it chooses the less times selected chromosome*.

UFS is designed to favour diversity only; it does not cause a search bias towards the fittest individuals. Thus, it should be combined with a replacement strategy that may yield some degree of selection pressure. In particular, we proposed to apply UFS along with the replace worst strategy (RW) (Section B.2 in Appendix B), which introduces *high selection pressure* because it maintains the best individuals appearing so far. In this way, all the individuals in the population represent promising search zones that deserve to be explored by means of PBX-α. This justifies the way UFS works: it induces a widespread search by providing the same opportunities to the chromosomes in the population of being female parents.

With the combination of UFS and the RW strategy (UFS&RW), we put together a diversification parent selection scheme that favours the production of offspring representing very different search zones with a replacement method that introduces high selection pressure. Other authors, with the aim of providing an effective search, have suggested GA schemes that connect techniques with high exploration and techniques with high exploitation. For example, in Shimodaira (1996), an algorithm is implemented employing large mutation rates and population-elitist selection. In Eshelman (1991), a GA is proposed, which combines a disruptive crossover operator with a conservative selection strategy. Finally, in Kemenade et al. (1995), authors suggest that higher selection pressures allow the application of more disruptive recombination operators.

In Section 4.1, we carry out an experimental study of the performance of the combination UFS&RW by means of its comparison against other possible combinations of selection mechanisms and replacement strategies. In Section 4.2, we investigate the behaviour of UFS and its influence on the number of offspring that receives the chromosomes that visit the population.

4.1. Study of the combination of UFS and the RW strategy

The performance of SSGAs will be strongly determined by the exploration/exploitation balance derived from the combination between the parent selection mechanism and the replacement strategy being applied. The objective of this section is to detect whether the balance maintained by the combination UFS&RW produces beneficial effects on the SSGA performance. In order to do this, we compare empirically this combination against other combinations of the selection mechanisms and the

Table 1
Combinations of selection mechanisms and replacement strategies

Selection mechanism (TS, RS, DS, FUSS, and DOS)	Replacement strategy (RW, FIFO, and RTS)		
	Selection pressure	Diversity	Div. & Sel. pres.
Selection pressure	TS&RW	TS&FIFO	TS&RTS
Diversity	RS&RW	–	RS&RTS
Div. & Sel. pres.	DS&RW	DS&FIFO	DS&RTS
	FUSS&RW	FUSS&FIFO	FUSS&RTS
	DOS&RW	DOS&FIFO	DOS&RTS

replacement strategies described in Appendix B. They are shown in Table 1. The ω parameter associated with RTS was set to 5, and n_T, in TS, was set to 2.

The eight types of combinations in Table 1 allow us to analyze the effects derived from the union of different ways of managing diversity and selection pressure through the selection mechanism and the replacement strategy. The combination between selection mechanism and replacement strategy favouring diversity only has not been considered; because it reduces to random search (the selection pressure is critical in ensuring sufficient optimization progress).

We have carried out minimisation experiments on six representative problems of the test suite described in Appendix C. They include:

- Three *unimodal* functions, f_{sph}, f_{Ros}, and f_{Sch}.
- Two *multimodal* functions, f_{Ras} and f_{Gri}, and
- One complex real-world problem, P_{sle}.

We considered this reduced set of test functions in order to facilitate the analysis of UFS. We have implemented several SSGAs that are distinguished uniquely by the selection mechanism and the replacement policy. They use real coding and apply the PBX-α operator with a fixed value for α ($\alpha = 1$). The population size was set to 60 individuals. No mutation operator was used. The selection mechanisms considered are used to determine the female parents. The male parents are selected at random. And, all the algorithms were executed 50 times, each one with a maximum of 100,000 evaluations.

Table 2 shows the results obtained. The performance measure used is the average of the best fitness function found at the end of each run. In addition, a two-sided t-test at 0.05 level of significance was applied in order to ascertain if the differences in the performance for UFS&RW are significant when compared against the ones for the other algorithms. We denote the direction of any significant differences as follows:

- A plus sign (+): the performance of UFS&RW is better than the one of the corresponding algorithm.

Table 2
Results for the SSGAs compared

Algorithm	f_{sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}	+	∼	−
TS&RW	4.11e−61−	1.84e+01+	3.18e−04+	9.65e+01+	1.02e−02+	2.48e+02+	5	0	1
TS&FIFO	2.06e+01+	6.27e+03+	5.77e+03+	1.78e+02+	1.95e+01+	6.27e+02+	6	0	0
TS&RTS	1.77e−26+	1.87e+01+	1.91e+00+	4.76e+01+	2.32e−02+	8.74e+01−	5	0	1
RS&RW	1.04e−39+	1.96e+01+	7.22e−02+	5.30e+01+	1.11e−02+	1.40e+02∼	5	1	0
RS&RTS	2.48e−15+	1.94e+01+	5.71e+01+	3.48e+01+	1.53e−02+	7.72e+01−	5	0	1
DS&RW	4.23e−55+	1.90e+01+	8.62e−04+	9.10e+01+	1.74e−02+	1.95e+02+	6	0	0
DS&FIFO	2.96e+01+	1.01e+04+	6.84e+03+	1.92e+02+	3.17e+01+	8.10e+02+	6	0	0
DS&RTS	2.19e−22+	2.21e+01+	1.42e+01+	3.32e+01+	1.42e−02+	9.48e+01∼	5	1	0
FUSS&RW	1.42e−75−	1.58e+01∼	7.78e−06∼	2.05e+02+	2.98e−02+	3.67e+02+	3	2	1
FUSS&FIFO	2.45e+01+	1.20e+04+	1.28e+04+	1.77e+02+	2.71e+01+	8.77e+02+	6	0	0
FUSS&RTS	2.41e−08+	2.06e+01+	6.67e+02+	4.76e+01+	1.64e−02+	9.65e+01∼	5	1	0
DOS&RW	4.92e−43+	1.79e+01+	7.45e−02+	7.62e+01+	3.02e−02+	3.25e+02+	6	0	0
DOS&FIFO	2.70e+01+	9.54e+03+	6.91e+03+	1.89e+02+	2.87e+01+	8.32+02+	6	0	0
DOS&RTS	6.43e−16+	2.72e+01+	9.88e+01+	5.50e+01+	3.40e−02+	1.71e+02+	6	0	0
UFS&RW	2.86e−56	1.56e+01	8.90e−08	2.64e+01	3.20e−03	1.20e+02			

- A minus sign (−): the algorithm improves the performance of UFS&RW.
- An approximate sign (∼): non significant differences.

The places where these signs do not appear correspond with the performance values for UFS&RW. We have included in the Table 2 the last three columns to facilitate the analysis of the results. They have the number of improvements, reductions, and non-differences (according to the *t*-test) in the performance for UFS&RW with regards to the one for the other algorithms.

We may observe that UFS&RW consistently outperforms all the other algorithms, because it achieves remarkable amount of improvements and few reductions. In addition, the number of non-differences is too small. These promising results reveal that UFS and RW interact adequately to allow the best solutions to be reached. With the application of RW, the population accumulates information about the location of the most promising regions, and with UFS, an exhaustive exploration of these zones is carried out.

Next, we examine the characteristics of the combinations that reach the best solutions. Table 3 displays the five algorithms that returned the best results for each test problem.

An inspection of Table 3 allows us to make the following comments:

- The combination UFS&RW arises as the most robust one. In fact, it provides the best results for four out six test problems.
- In general, the RW replacement strategy is very successful. Most of the best algorithms are based on this strategy.
- The RTS replacement strategy becomes competitive for the complex problems, f_{Ras}, f_{Gri}, and P_{sle}. This is due to its ability to maintain population diversity (it is a *crowding* method). In particular, the union of RTS and RS (selection mechanism that favours diversity only) provides a robust operation for these problems. This combination was already suggested in Harik (1995).
- FUSS takes part of two combinations that are worthy of remark: FUSS&RW and FUSS&RTS. The first achieves promising results for the unimodal problems. The second appears among the best performing algorithms for the two complex problems f_{Ras} and P_{sle}. FUSS prefers individuals that belong to sparsely populated fitness regions. In this way, FUSS shares with UFS the aim of providing a uniform sampling on the search areas represented in the current population. The good results of FUSS and UFS allow us to conclude that this idea represents a promising way to improve SSGA performance.
- The combination TS&RW has been advantageous for the unimodal problems, f_{sph}, f_{Ros}, and f_{Sch}. However, it does not appear among the best performing algorithms for the complex problems. This is reasonable, because both TS and RW are focused on the production of selection pressure only, which is beneficial for the unimodal problems and detrimental for the complex ones.

4.2. Effects of UFS

In this section, we attempt to discover those behavioural characteristics that allow UFS to decisively affect SSGA performance. In order to do this, we have introduced Fig. 2, which compares UFS and the selection mechanisms used in Section 4.1. It displays the percentage of chromosomes that, during their lifetime, either they were never selected as female parent or once or twice or so on, when tackling with f_{sph}. The RW replacement is used as well.

We may remark the following facts:

- Working with UFS, most chromosomes that went through the population have produced

Table 3
Algorithms that provide the best results

f_{sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}
FUSS&RW	UFS&RW	UFS&RW	UFS&RW	UFS&RW	RS&RTS
TS&RW	FUSS&RW	FUSS&RW	DS&RTS	TS&RW	TS&RTS
UFS&RW	DOS&RW	TS&RW	RS&RTS	RS&RW	DS&RTS
DS&RW	TS&RW	DS&RW	TS&RTS	DS&RTS	FUSS&RTS
DOS&RW	TS&RTS	RS&RW	FUSS&RTS	RS&RTS	UFS&RW

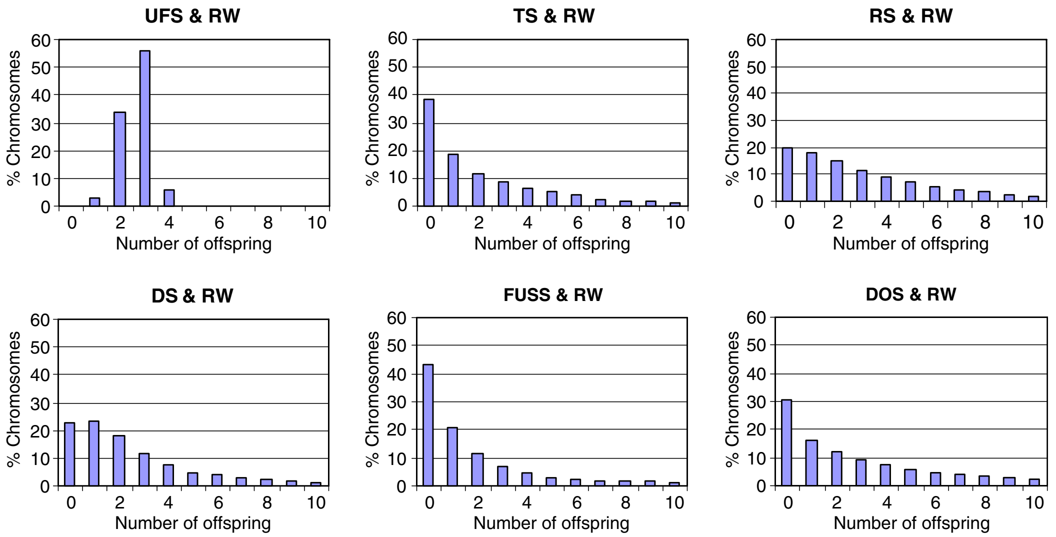

Fig. 2. Distributions of the number of offspring.

two or three offspring. Thus, all the zones represented in the population have been equally explored. However, a remarkable circumstance that occurs with all the other selection methods is that considerable percentage of chromosomes did not generate offspring during their stay in the population. This means that certain promising search regions that deserve attention are not sampled with the enough interest and fall into oblivion. This may be very detrimental for the complex problems.

- TS, FUSS, and DOS assign many offspring to few chromosomes and few offspring to many chromosomes. This is caused, firstly, because the selection method show a high tendency to select the best individuals in the population as female parents, and secondly, because of the application of RW, which retains the best individuals in the population, giving them the opportunity of producing offspring during long time periods.
- RS and DS slow down the imbalance caused by the previous selection mechanism. On the one hand, RS eliminates the bias towards the election of the fittest individuals. On the other hand, DS favours the selection of both better and worse solutions.

5. Male selection mechanism: The negative assortative mating

In this section, we undertake the study of the selection of the male parent. This task may be carried out by means of a *mating selection* mechanism, which determines the way the chromosomes are mated to apply the crossover to them. In the conventional GA, no mating strategy is applied to the results of selection; that is, parents are approved without any further examination after they are chosen at random or just by fitness. Indeed, mating in nature is more complicated. Inspired by this observation, a number of mating strategies for GAs have been proposed to deal with the issues of population diversity or selection pressure in a natural way (Ting et al., 2003; Bandyopadhyay et al., 1998; Craighurst and Martin, 1995; Fernandes and Rosa, 2001; Matsui, 1999).

We have chosen the *Negative Assortative Mating* (NAM) (Fernandes and Rosa, 2001) as the technique for the male parent selection. Assortative mating is the natural occurrence of mating between individuals of similar phenotype more or less often than expected by chance. Mating between individuals with similar phenotype more often is called positive assortative mating and less often is called negative assortative mating. Fernandes and Rosa (2001) implement these ideas to design two mating selection mechanisms. A first parent is selected by the roulette wheel method and n_{ass} chromosomes are selected with the same method (in our experiments n_{ass} is equal to 5, and all the candidates are selected at random). Then, the similarity between each of these chromosomes and the first parent is computed (similarity between two real-coded chromosomes is defined as the Euclidean distance between them). If assortative mating is *negative* (NAM), then the one with less similarity is chosen. If it is *positive*, the genome more similar to the first parent is chosen to be the second parent. Clearly, NAM may increase genetic diversity in the population by mating dissimilar genomes with higher probability. There exists a mechanism similar to NAM, which is called *correlative tournament selection* (Matsui, 1999). It chooses the mate with higher fitness and Hamming distance from a set of candidates.

We have chose NAM as technique for the male parent selection due to two reasons:

- NAM forces the creation of diversity following a direct way. Most of the mating selection mechanisms presented in the literature (Craighurst and Martin, 1995; Eshelman and Schaffer, 1991; Bandyopadhyay et al., 1998) are really *decision-making* strategies, because they determine whether two parents may be crossed over or not. Their main ideas are to wait that an adequate pair of parents (fulfilling a particular condition) could arrive at any moment. On the other hand, NAM acts directly; it finds the pair of parent that really will promote the generation of diversifying offspring.
- In an SSGA with the RW replacement, the successive application of NAM and PBX-α may produce a *self-adaptive diversification processing* that favours the generation of *useful diversity* (i.e., population diversity that in some way helps produce good solutions (Mahfoud, 1995)).

In Section 5.1, we explain the second of these reasons in detail and in Section 5.2, we compare the performance of NAM with the one of random mating, which was applied up to now in these experiments.

5.1. Self-adaptive diversification by means of NAM and PBX-α

An SSGA with the RW replacement maintains a population with the best elements appeared so far (*elite population*). There are two important kinds of information in an elite population that may be exploited to generate new elements: (1) where the best elements are located and (2) how these elements are distributed (close together, disperse, etc.). Precisely, some real-parameter crossovers may obtain their self-adaptive ability because they make use of the second type of information.

NAM manages the second type of information as well, with the aim of influencing the extent of the areas on which PBX-α samples the offspring. In particular, it extracts information from the elite population about how far away promising individuals are distributed located from each other. Then, PBX-α may exploit this information with two intentions:

- To use the widest areas for the generation of offspring. The idea is to create offspring the most distant from their parents as possible, in order to contribute diversity to the population.
- To produce offspring with high fitness quality. Since we use information about the distribution of the best individuals to generate new chromosomes, we may expect that these ones will show suitable fitness values.

In this way, the successive application of NAM and PBX-α allows the two main objectives of a GA to be tackled simultaneously: *to obtain high-quality solutions and to enhance population diversity, that is, to promote useful diversity*. In addition, we should point out that since the diversity levels provided by NAM and PBX-α depend directly on the distribution of the individuals in the population, we may say that they carry out a *self-adaptive diversification processing*.

5.2. Empirical analysis of NAM

In this section, we analyse the way NAM affects the performance of an SSGA that applies the RW replacement. In addition, we are particularly interested on ascertaining the influence of the α parameter associated with PBX-α on the action of NAM, because α determines the spread associated with the probability distributions used by this operator to create offspring. In order to do this, we have introduced Table 4. It compares the performance of algorithms that apply random mating for male parent selection with others that incorporate NAM. Different values for α were tried, $\alpha = 0.7$, 0.8, 0.9, and 1. We have experimented with other values for this parameter; however they offered very poor results and thus were discarded. All the algorithms consider UFS as female parent selection and the RW replacement. The first type of algorithms will be denoted as RM-α (random mating) and the second NAM-α. We indicate, in bold, the result that becomes significantly the best (according to a *t*-test). If there are not any differences between the corresponding algorithms, neither of them appears remarked.

We clearly notice that, in general, with $\alpha = 0.7$ and $\alpha = 0.8$, NAM provides meaning improvements on most test problems with regards to the corresponding RM algorithms. With $\alpha = 0.9$ and $\alpha = 1$, it attains the best performance as well for the complex f_{Ros}, the multimodal f_{Ras}, and the complex real-world problem P_{sle}. These two facts suggest that NAM is a promising diversification tool that is able to empower the performance of SSGAs. This affirmation may be reinforced with Table 5, which outlines the five algorithms that returned the best solutions for each problem. We point out two important remarks from this table:

- For most problems, the best performing algorithm is based on NAM.
- In general, the best solution for each problem is reached by NAM using different α values. For the multimodal problems, f_{Ras} and f_{Gri}, and the complex P_{sle} the most effective α value is 1, whereas for the unimodal problems lower values become more beneficial. This indicates that α has an important influence on the effectiveness of NAM.
- Despite the previous fact, we may consider that the algorithm based on NAM with $\alpha = 0.8$, NAM-0.8 (boldfaced in Table 5), achieves an acceptable *robustness* with regard to the other algorithms, because it appears among the best algorithms for all the problems considered in our experimentation. None of the remaining algorithms allows a better operation to be obtained.

Another interesting observation in Table 4 is that the results of NAM with $\alpha = 0.9$ and $\alpha = 1$ are worse than those for the corresponding RM algorithms for the unimodal f_{sph} and f_{Sch}, and for the non complex multimodal f_{Gri}. The application of NAM along with the use of high values for α forces PBX-α to use ranges for the probability distributions that are too long (see Fig. 1). This may be suitable for complex problems where the diversity can help to reach promising search zones (as we have observed, $\alpha = 1$ is the best choice for these problems); however, this excessive amount of exploration becomes disadvantageous for problems with the features of the aforementioned test functions.

6. Female and male differentiation process

In this section, we present a FMD process that determines the individuals in the current population

Table 4
Results obtained using NAM

Algorithm	f_{sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}
RM-0.7	1.97e−066	2.24e+001	5.41e−003	1.65e+002	2.51e−002	6.60e+002
NAM-0.7	**9.88e−095**	1.91e+001	**7.97e−011**	**7.49e+001**	**5.17e−003**	**1.89e+002**
RM-0.8	9.89e−080	1.98e+001	5.29e−008	8.09e+001	4.58e−003	3.39e+002
NAM-0.8	1.44e−081	**1.47e+001**	6.05e−008	**3.08e+001**	6.84e−003	**7.14e+001**
RM-0.9	**4.35e−071**	1.56e+001	**5.87e−009**	4.08e+001	**4.04e−003**	2.07e+002
NAM-0.9	3.67e−060	**1.42e+001**	2.68e−003	**1.45e+001**	8.81e−003	**3.50e+001**
RM-1.0	**2.86e−056**	1.56e+001	**8.90e−008**	2.64e+001	**3.20e−003**	1.20e+002
NAM-1.0	2.10e−041	**1.48e+001**	3.28e+000	**1.04e+001**	1.00e−002	**2.25e+001**

Table 5
Algorithms that provide the best results

f_{sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}
NAM-0.7	NAM-0.9	NAM-0.7	NAM-1.0	RM-1.0	NAM-1.0
NAM-0.8	**NAM-0.8**	RM-0.9	NAM-0.9	RM-0.9	NAM-0.9
RM-0.8	NAM-1.0	RM-0.8	RM-1.0	RM-0.8	**NAM-0.8**
RM-0.9	RM-0.9	**NAM-0.8**	**NAM-0.8**	NAM-0.7	RM-1.0
RM-0.7	RM-1.0	RM-1.0	RM-0.9	**NAM-0.8**	NAM-0.7

that may become female parents only, or male parents only, or, even, both female and male parents. This process should be carried out before the application of the female and male selection mechanisms (which will be applied on the corresponding groups). In this way, it may be considered as a *pre-selection mechanism*.

The proposed FMD process needs two parameters, N_F and N_M, with $N_F \leqslant N$ and $N_M \leqslant N$ (N is the population size) and obtains G_F and G_M as follows (Fig. 3):

- G_F consists of the N_F best individuals in the population, and
- G_M is made of the N_M best individuals in the population.

In addition, it should be ensured that either $N_F = N$ or $N_M = N$ is fulfilled. Next, we provide two remarks derived from this definition:

1. In the case of $N_F = N_M$, there is not female and male differentiation, reaching the standard way of applying PCCOs.
2. $G_F \cap G_M \neq \emptyset$, which means that some individuals may be both female and male parents (see Fig. 3). In particular, the $N_{\min}$ best individuals in the population have this characteristic, where $N_{\min} = \min\{N_F, N_M\}$. We have assumed that these individuals may be well-suited to act as both female and male parents.

Another important feature of this FMD process is that it introduces *selective pressure* in the processes of selection of the female and male parents (which are applied later). In addition, we may point out that:

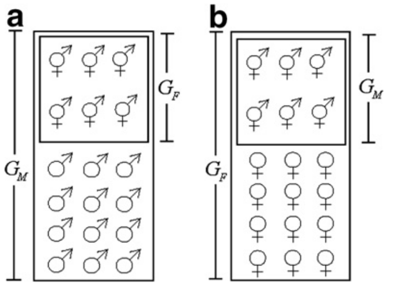

Fig. 3. Female and male differentiation imposed by the FMD process, considering: (a) $N_F < N_M$ and (b) $N_F > N_M$. (We assume that the two populations are ordered based on the fitness values of the chromosomes.)

- The impact of the N_F parameter on this selective pressure is simple and predictable.
- The range of selective pressure that can be made by varying the N_F parameter is very large.

These are two desirable features for a selection process (Bäck, 1994). On the one hand, when N_F is low, high selection pressure degrees are achieved, which forces the search process to be very focused in the best regions. On the other hand, if N_F is high, the selection pressure is softened, providing an extensive sampling on the search areas represented in the current population.

Now, we may outline our RCGA model based on PCCOs, which will be called FMD-U&N. The FMD process is performed before the application of UFS and NAM and returns G_F and G_M. Then, UFS selects a female parent from G_F and NAM picks a male parent from G_M. The remaining steps are accomplished following the usual way. Its basic algorithm step is the following:

```
1. Construct $G_F$ and $G_M$ applying FMD process.
2. Select a female parent from $G_F$ using UFS.
3. Select a male parent from $G_M$ using NAM.
4. Create an offspring applying PBX-$\alpha$ to the parents.
5. Evaluate the offspring with the fitness function.
6. Introduce the offspring in the population using the RW strategy.
```

We should recognize that the idea of incorporating chromosome differentiation in GAs is not new. Other authors (Bandyopadhyay et al., 1998; Goh et al., 2003) added a sexual differentiation process in order either to investigate new models or to have a clear and balanced separation of functions between exploration and exploitation. The main difference with the model proposed in this paper comes from taking into account the different roles of the parents in the PCCOs.

Next, we investigate the influence of the parameters associated with the FMD process, N_F and N_M, on the performance of the FMD-U&N algorithm. We have carried out experiments with this algorithm on the test problems used in Sections 4.1 and 5.2, considering different values for N_F and N_M ($N_F = 1$, 5, 25, 50, 100, 200, 300, and 400 individuals, and $N_M = 25$, 50, 100, 200, 300, and

400 individuals). All the possible combinations of these values were investigated. Table 11 (Appendix D) shows the results obtained.

Now, we may examine the characteristics of the N_F and N_M combinations that reach the best solutions. Table 6 displays the fifteen combinations that returned the best results for each test problem in a sorted way. We have underlined, in boldface, the best combinations with $N_F = N_M$, which represent the standard way of implementing the RCGAs based on PCCOs.

For every test function, the combination that achieves the best results fulfils that $N_F \neq N_M$, that is, it implies female and male differentiation. We have applied a *t*-test in order to ascertain if differences in the performance of the best combinations with $N_F \neq N_M$ are significant when compared against the one for the best combinations with $N_F = N_M$ (we have introduced a '*' sign when this occurs). Table 7 has the results. For most test functions, we may see that the female and male differentiation allows the performance of FMD-U&N to be improved. Only for f_{Gri}, a comparable behaviour is achieved.

Table 7
Best combinations with $N_F \neq N_M$ versus best combinations with $N_F = N_M$

Test problem	N_F,N_M	Average best fitness
f_{sph}	5, 100*	9.98e−187
	50, 50	6.19e−095
f_{Ros}	5, 100*	1.56e+000
	50, 50	1.46e+001
f_{Sch}	25, 50*	1.01e−012
	50, 50	3.54e−010
f_{Ras}	400, 100*	2.60e+000
	200, 200	3.58e+000
f_{Gri}	400, 300	3.48e−004
	300, 300	5.42e−004
P_{sle}	200, 400*	5.45e+000
	400, 400	7.84e+000

Another important observation, from Table 6, is that the best solution for each problem is reached by using different values for N_F and N_M. In particular, we may remark the big differences between the best N_F values for the multimodal and complex problems (f_{Ras}, f_{Gri}, and P_{sle}) and the ones for the unimodal problems (f_{sph}, f_{Ros}, and f_{Sch}):

- Low N_F values (e.g. 5 or 25 individuals) induce very high selection pressures, which may be suitable to obtain *accuracy* on unimodal problems. However, this is not the unique determinant aspect to achieve success for this type of problems (and in particular, in our case, since we follow a skewed initialisation (Section C.1 in Appendix C)). The use of high N_M values (e.g. 50 or 100 individuals) enlarges the sampling zones. This circumstance has produced synergetic effects with the selection pressure, ensuring that the algorithm may progress towards better zones.
- The use of high N_F values along with the application of UFS induces a scattered search, due to different female parents become the centre of attention of the PCCO. This high exploration of the search space is essential to provide *reliability* for multimodal and complex problems. In

Table 6
Best N_F and N_M values used by the FMD-U&N algorithm

f_{sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}
5, 100	5, 100	25, 50	400, 100	400, 300	200, 400
5, 50	5, 200	**50, 50**	300, 100	200, 400	300, 400
1, 200	5, 300	5, 100	**200, 200**	**300, 300**	**400, 400**
25, 50	25 100	100, 25	400, 50	300, 400	50, 400
5, 200	25 200	25, 100	100, 400	400, 200	400, 300
1, 300	25 300	5, 200	200, 100	100, 300	300, 300
25, 100	5 400	200, 25	100, 300	300, 200	400, 200
5, 300	25 400	100, 50	100, 200	200, 300	100, 400
1, 400	25 50	5, 300	300, 200	100, 400	200, 300
50, 50	50 100	50, 100	200, 300	300, 100	300, 200
100, 25	50 200	300, 25	300, 50	50, 400	100, 300
5, 400	50 300	25, 200	50, 400	200, 200	50, 300
200, 25	**50, 50**	5, 400	200, 50	100, 200	25, 400
50, 25	50, 400	400, 25	400, 25	400, 400	200, 200
25, 200	100, 50	200, 50	50, 300	25, 400	100, 200

addition, the application of very high N_{M} values (e.g. 400 individuals) reinforces this exploration ability, inducing a promising performance on the most complex problems (e.g. P_{sle}).

Thus, we may conclude that the FMD process let us easily tune the SSGA in order to obtain promising results on unimodal problems, by using low values for the N_{F} parameter, or on multimodal and complex problems, by using higher values for this parameter.

7. Global and local RCGAs

An important conclusion obtained from Section 6 is that the FDM process allows the application of PCCOs to be more effective. This is possible due to:

- The large range of selective pressure obtained by varying N_{F}, and
- The possibility of supplying adequate exploration levels by controlling N_{M}.

These two characteristics allow us to design two different kind of specialized search algorithms:

- RCGAs that offer *reliable* solutions when they attempt to solve multimodal and complex problems. An example may be the FMD-U&N algorithm adopting $N_{\mathrm{F}} = 200$ and $N_{\mathrm{M}} = 400$ (see Table 6), and
- RCGAs that reach *accurate* solutions when they deal with unimodal problems. An example may be the FMD-U&N algorithm using $N_{\mathrm{F}} = 5$ and $N_{\mathrm{M}} = 100$ (see Table 6).

In this paper, these algorithms will be termed *global* RCGAs and *local* RCGAs, respectively. In order to achieve a *robust* operation for problems with different characteristics, global and local RCGAs should be hybridized in such a way that their beneficial advantages might be offered simultaneously, allowing the most promising search space regions to be reached and refined. Thus, the objective of this section is the design of a *hybrid* RCGA model that might be suited to most practical problems.

7.1. The conflict between accuracy and reliability

There exists a fundamental conflict between *accuracy* and *reliability* when searching for the global optimum in most practical problems (Renders and Flasse, 1996). Traditionally, this conflict was tackled by means of advanced genetic operators (e.g. the non-uniform mutation operator proposed in (Michalewicz, 1992), adaptation of GA control parameters (Eiben et al., 1999), heterogeneous distributed populations (Herrera and Lozano, 2000), etc.). Nowadays, an alternative that receives special attention is the *hybridization* of GAs with other search techniques. Three important examples are:

- *Memetic algorithms* (Moscato, 1999). They are GAs that apply a separate local search process (which searches efficiently only for a local optimum) to members of the population after recombination and mutation. In this case, the local search procedure works within the GA.
- *Continuous hybrid algorithm* (Chelouah and Siarry, 2003). It comprises two main stages. The first stage involves the run of a RCGA. The second stage involves the application of a local search process to the best individual found by the RCGA.
- *Hybridization of GAs with different purposes.* Kazarlis et al. (2001) propose the use of a *microgenetic algorithm* (MGA) (GA with small population that evolves for a few generations) as a generalized hill-climbing operator. They combine a standard GA with the MGA to produce a hybrid genetic scheme. In contrast to conventional hill climbers that attempt independent steps along each axis, a MGA operator performs a genetic local search. The authors claimed that the MGA operator is capable of evolving paths of arbitrary direction leading to better solutions and following potential ridges in the search space regardless of their direction, width, or even discontinuities.
  Lozano et al. (2004) present a real-coded memetic algorithm that applies a *real-parameter crossover hill-climbing*. This hill-climbing maintains a pair of parents and performs repeatedly the PBX-α operator on this pair until some number of offspring is reached. Then, the best offspring is selected and replaces the worst parent only if it is better. The authors claimed that this process may be conceived as a *micro selecto-recombinative* RCGA.

7.2. Combining global and local RCGAs

In this section, we propose a hybrid RCGA model that combines a global RCGA (with $N_{\mathrm{F}}^{\mathrm{G}}$

and N_M^G as N_F and N_M values, respectively) and a local RCGA (with N_F^L and N_M^L as N_F and N_M values, respectively). The hybridization is accomplished following the idea of the *continuous hybrid algorithm* (Chelouah and Siarry, 2003) because is very simple: first, it runs the global RCGA during the P_G% of the available evaluations, and then, it performs the local RCGA. The initial population for the local RCGA consists of the N_{max} best individuals in the final population of the global RCGA, where $N_{max} = \max\{N_F^L, N_M^L\}$.

This hybridization follows a classical heuristic: "*to protect the exploration in the initial stages and the exploitation later*". This heuristic has been considered to design other search techniques, such as *simulated annealing* (Kirkpatric et al., 1983). With the initial exploration, the diversity became greater, increasing the probability of finding zones which are close to optimal solutions. Then, supposing that the population has information about these zones, the convergence towards the optimum is produced through exploitation.

We have implemented three instances of this hybrid model. The global RCGA is the FMD-U&N algorithm adopting $N_F^G = 200$ and $N_M^G = 400$ and the local RCGA is the same algorithm but with $N_F^L = 5$ and $N_M^L = 100$. The instances are distinguished with regards to the values considered for P_G ($P_G = 25\%$, 50%, and 75%). They will be called GL-25, GL-50, and GL-75, respectively. Their results are outlined in Table 8. We have included the results for the global and local RCGAs when they are executed independently.

We should point out that, for four problems, there exist instances of our hybrid RCGA model that perform better than the sole usage of the corresponding global and local RCGAs: f_{Ros} (GL-25), f_{Ras} (GL-50 and GL-75), f_{Gri} (GL-25, GL-50, and GL-75) and P_{sle} (GL-25, GL-50, and GL-75). This means that the hybrid technique proposed is a suitable way to achieve *synergy* between global and local RCGAs.

Another important observation is that GL-25 and GL-75 have returned solutions for f_{Ros} and f_{Gri}, respectively, which are the best ones as compared with the solutions achieved by all the algorithms previously studied (see Table 7).

7.3. Comparison with other algorithms

The main aim of this section is to compare our hybrid RCGA method with other *metaheuristics for continuous optimization* presented in the literature:

- Two RCGAs based on PCCOs: SPC-PNX (Ballester and Carter, 2004) and G3-PCX (Deb et al., 2002).
  - The SPC-PNX algorithm is a steady-state RCGA model that uses the PNX crossover operator. We have considered four instances that use different population size ($N = 40$, 60, 100 and 200). They will be called SPC-PNX-N.
  - The G3-PCX algorithm is a steady-state RCGA model that uses the PCX crossover operator. We have implemented several instances that use different values for the λ parameter of the PCX operator ($\lambda = 2$, 3, and 4), $\mu = 3$, and the population size is 150 individuals. The other parameters for the PCX operator are: $\sigma_\varsigma^2 = 0.1$ and $\sigma_\eta^2 = 0.1$. These algorithms will be referred as G3-PCX-λ.
- Two hybrid RCGAs: the *real-coded memetic algorithm with crossover hill-climbing* (RCMA-XHC) (Lozano et al., 2004) and a hybrid algorithm that combines the CHC algorithm (Eshelman, 1991) with the Solis and Wets' algorithm, which is a local searcher, (CHC-SW) (Solis and Wets, 1981).
  - The RCMA-XHC, whose authors claimed that this algorithm improves the performance of other real-coded memetic algorithms appeared in the literature (Lozano et al., 2004).

Table 8
Results for the hybrid RCGA model

Algorithm	f_{sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}
Local RCGA	9.98e−187	1.56e+000	1.74e−009	2.90e+002	1.27e−002	1.64e+002
GL-25	3.17e−147	7.61e−001	1.61e−007	1.33e+001	2.22e−017	4.69e+000
GL-50	1.29e−104	6.03e+000	1.22e−005	8.26e+000	1.33e−017	3.25e+000
GL-75	3.94e−061	1.22e+001	3.26e−002	3.74e+000	0.00e+000	2.72e+000
Global RCGA	2.95e−018	1.91e+001	3.12e+001	1.92e+001	4.93e−004	5.45e+000

  - CHC-SW follows the idea of the continuous hybrid algorithm (Chelouah and Siarry, 2003). First, the CHC is performed during the P_G% of the available evaluations, and then, the local searcher refines the best individual returned by CHC. We have considered $P_G = 25\%$, 50%, and 75%. These algorithms will be termed CHC-SW-P_G. We have included the CHC algorithm as well.
- *Enhanced simulated annealing* (ESA) (Siarry et al., 1997). It is a continuous variant of the well-known Simulated Annealing (Kirkpatric et al., 1983). All the parameters were set as the authors suggest in Siarry et al., 1997. In addition, the normalization guidelines proposed in that work have been followed. We have implemented a simple linear variable change $X_{\text{norm}} = \alpha \cdot X_{\text{real}} + \beta$ that brings X_{norm} into the $[-1, 1]$ range.
- *Differential evolution algorithm* (DE[1]) (Storn and Price, 1997). It is an Evolutionary Algorithm that borrows the idea from *Nelder &Mead's simplex algorithm*. It evolves a population of solution by combining its individuals. We have considered three instances:
  - DE uses the *DE/rand/1/bin* strategy, with 10· Dimension members in the population, $F = 0.8$, and $\text{CR} = 0.5$, as the code suggests; and
  - two instances with 60 and 100 individuals, which use the *DE/Rand/1/Exp* strategy, $F = 0.5$, and $\text{CR} = 0.8$. They are DE-60 and DE-100 respectively.
- *Comprehensive learning particle swarm optimizer* (CLPSO[2]) (Liang et al., 2004a,b). It is a variant of the original particle swarm optimizer (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995), which simulates the behaviours of the birds flocking. We have compared two instances: CLPSO that uses 10 particles, as Liang et al. (2004b) suggests, and CLPSO-30 that uses 30.
- *Evolution strategy with covariance matrix adaptation* (CMA-ES[3]) (Hansen and Ostermeier, 2001; Hansen et al., 2003). It represents the state of the art of Evolution Strategies and it is a referent in the continuous optimization field. The initial step size $\sigma^{(0)}$ is set to half of the initialisation interval.
- *Evolutionary programming with Levy probability distribution based mutations* (LEP) (Lee and Yao, 2004) and the variant called adaptive LEP (ALEP) (Lee and Yao, 2004). LEP is a recent Evolutionary Programming and ALEP is a LEP algorithm that considers four different values for the α parameter, simultaneously. We have implemented the ALEP algorithm and a LEP instance with $\alpha = 1.4$. In addition, we have tried two different values for the initial step sizes σ_0 (1 with $\sigma_0 = 0.015 \cdot D_{\text{width}}$, and 2 with $\sigma_0 = 0.01$).

For this comparison, we will use all the 18 test functions described in Appendix C. We have chosen GL-25 (Table 8) for the comparison, because it shows an acceptable level of robustness. The stop criterion of every algorithm has been set to 100.000 objective function evaluations. The results for all these algorithms are included in Tables 12–14 (Appendix E). A t-test was applied to detect significant differences between GL-25 and the other algorithms.

We have introduced the Fig. 4 in order to facilitate the analysis of these results. It shows the percentage of improvements, reductions, and non-differences, according to the t-test, obtained when comparing the GL-25 algorithm with the other algorithms for all the test problems.

We can draw the following conclusions from Fig. 4:

- GL-25 performs better than all the other algorithms for more than the 60% of the test problems. Thus, we may conclude that the hybrid RCGA model proposed arises as a promising algorithm to deal with continuous optimisation problems.
- CMA-ES, ESA, CLPSO, CLPSO-30, and CHC-SW arise as the most competitive algorithms for GL-25.

In order to compare the behaviour of CMA-ES, ESA, CLPSO and CHC-SW with the one of GL-25, we examine their results differentiating two groups of test problems: unimodal functions ($f_{\text{sph}} - f_{\text{QNoise}}$), and multimodal ones ($f_{\text{Ras}} - f_{\text{Schaffer}}$ and P_{sle} and P_{fms}). Fig. 5 shows the results:

---

[1] We have used the matlab code offered at http://www.icsi.berkeley.edu/~storn/code.html#matl.

[2] The code is offered at http://www.ntu.edu.sg/home/EPNSugan/.

[3] We have used the MATLAB code, Version 2.34a, available from http://www.bionik.tu-berlin.de/user/niko/index.html.

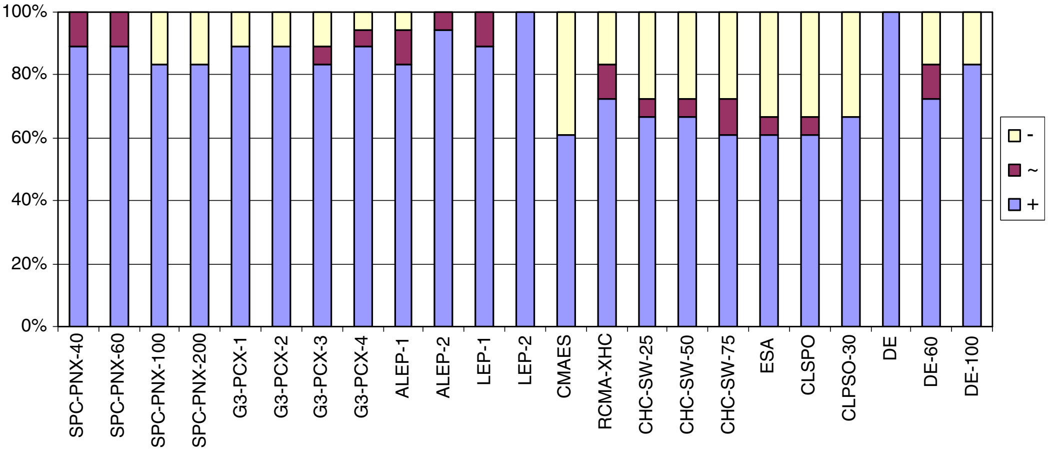

Fig. 4. Comparison of GL-25 with the other algorithms.

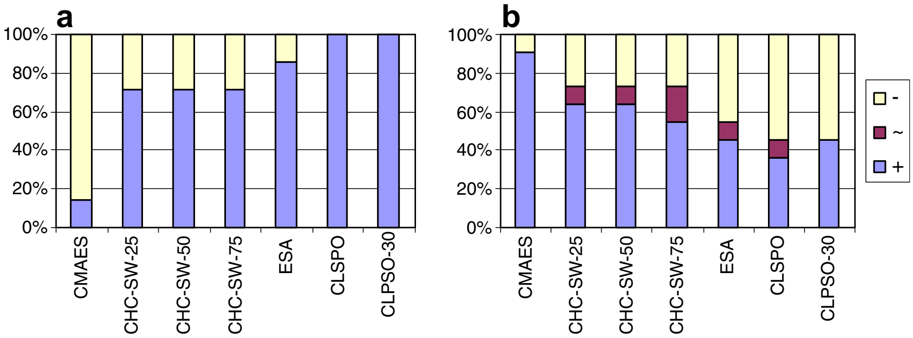

Fig. 5. Comparison of GL-25 with CMAES, CHC-SW instances, ESA, and PSO, considering: (a) unimodal functions, and (b) multimodal functions.

- On the one hand, CMA-ES outperforms the GL-25 algorithm on most unimodal functions (Fig. 5a). However, GL-25 outperforms CMA-ES on most multimodal functions (Fig. 5b). Since, CMA-ES was interpreted as a robust local search strategy (Hansen and Ostermeier, 1996), it achieves very good results on unimodal functions.
- On the other hand, ESA, CLPSO, and CLPSO-30 outperform the GL-25 algorithm on most multimodal functions (Fig. 5b). However, GL-25 obtains better results on the unimodal ones (Fig. 5a). We can see in Tables 12–14 that ESA, CLPSO, and CLPSO-30 obtain outstanding results on multimodal functions (for example, see the results obtained by ESA, CLPSO, and CLPSO-30 for f_{SRas} and f_{Boh}), whereas that does not occur on unimodal functions. This reveals that the ESA, CLPSO, and CLPSO-30 algorithms prefer the global search.
- Although GL-25 outperforms the CHC-SW instances, they show a similar behaviour, with regards to GL-25, for both kinds of test functions. This may be due to these algorithms are designed in order to achieve a balanced behaviour between global and local search, by combining the CHC algorithm with the Solis and Wets' local searcher.

Finally, we introduce Fig. 6 with the aim of determining the general performance of GL-25 on every test problem. It shows the percentage of improvements, reductions and non-differences obtained by GL-25 for each function (with regards to the other algorithms).

Several remarks are worth being mentioned from Fig. 6:

- GL-25 achieves better results than the 60% of all the algorithms on all the functions, except on

f_{MSch}, *EF10*, and f_{Schaffer}, where it obtains better or equal results than more than the 50%. Thus, we may conclude that the balance between global and local search obtained by our hybrid method may produce outstanding benefits.

- GL-25 returns the best results for function f_{QNoise}, which is *highly noisy*.
- GL-25 is one of the best algorithm on the multimodal functions f_{Gri} and f_{Boh}, the two real-world problems, P_{sle} and P_{fms}, and the *Rotated Rastrigin's function*, f_{RRas}.
- Although GL-25 performs well on all the functions, the main difficulties appear on f_{sph}, f_{Sch}, f_{MSch}, *EF10*, and f_{Schaffer}. On the one hand, f_{sph} and f_{Sch} are non complex unimodal functions, which require a strong local behaviour in order to be adequately solved, whereas f_{MSch}, *EF10* and f_{Schaffer} are very complex problems that need extra global search to obtain reliable results. Since GL-25 attempts to keep a balance between local and global search, it may not achieve highly accurate results for these problems.

To sum up, we may conclude that our proposal of combining global and local RCGAs is very competitive with the state-of-the-art on *metaheuristics for continuous optimization*, because it may produce a robust operation for test problems with different characteristics.

8. Conclusions

This paper presented a FMD process and two parent selection mechanisms, UFS and NAM, with the aim of improving the performance of the PCCOs:

- The FMD process assigns the role of female or/and male parent to the chromosomes in the population.
- UFS broads the search because it obliges PCCOs to consider different zones for the generation of offspring. Thus, it encourages an exhaustive coverage of the regions represented by the female parent.
- NAM diversifies because it forces PCCOs to create offspring very dissimilar from their parents. In this way, it helps PCCOs to give rise to useful diversity.

In addition, the paper tackled the election of the most adequate evolution model to take out profit from UFS and NAM. SSGAs with the RW replacement were finally selected, because they supply high selection pressure that becomes well-suited for the meaningful operation of these two selection mechanisms.

An experimental study carried out with the PBX-α operator has shown that these three processes may empower the work of PCCOs. Another important conclusion is that the two parameters associated with the FMD process, N_{F} and N_{M}, may be adequately adjusted to design reliable global RCGAs and accurate local RCGAs. With the aim of achieving a *robust* operation, we have followed a simple hybridization technique to put together these specialized search algorithms. We have confirmed empirically that: (1) this technique allows synergy

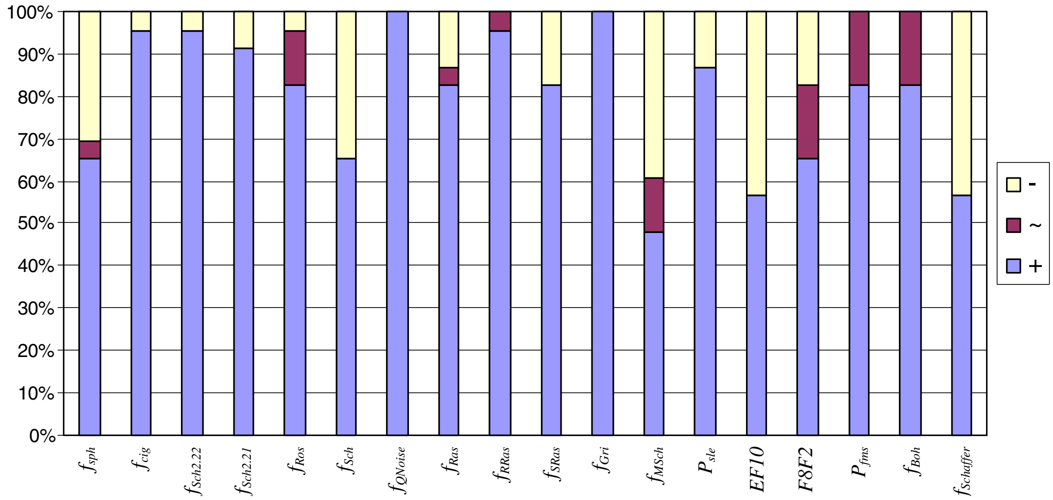

Fig. 6. Performance of GL-25 on every test problem.

to occur between global and local RCGAs, that is, their combination performs better than the sole usage of any of them, and (2) it is very competitive with *state-of-the-art on metaheuristics for continuous optimization*.

In essence, the research line initiated with the present work is indeed worth of further studies. We are currently extending our investigation to different test-suites and real-world problems. Also we intend to: (1) design adaptive FMD processes that select female and male parents depending on the current state of the search, (2) study the effectiveness of the FMD process, UFS, and NAM when *multiparent* crossover operators are applied (e.g. PCX (Deb et al., 2002; Deb, 2005)), and (3) study adaptive mechanisms to adjust the P_G parameter required by the hybrid RCGA model proposed.

Acknowledgements

This research has been supported by the project TIN2005-08386-C05-01 and a scholarship from the Education and Universities Spanish Government Secretariat given to the author C. García-Martínez. In addition, the authors would like to thank the anonymous reviewers for their constructive comments and suggestions.

Appendix A. Abbreviation index

See Table 9.

Appendix B. Diversity and selection pressure in SSGAs

There are two primary factors in the search carried out by a GA (Whitley, 1989):

- *Selection pressure*. In order to have an effective search there must be a search criterion (the fitness function) and a selection pressure that gives individuals with higher fitness a higher chance of being selected for reproduction, mutation, and survival. Without selection pressure, the search process becomes random and promising regions of the search space would not be favoured over non-promising regions.
- *Population diversity*. It is crucial to a GA's ability in order to continue the fruitful exploration of the search space.

Selection pressure and population diversity are inversely related. Increasing selection pressure results in a faster loss of population diversity, while maintaining population diversity offsets the effect of increasing selection pressure. These two factors should be controlled in order to obtain their beneficial advantages simultaneously, allowing the most promising search space regions to be reached and refined.

The population diversity versus selection pressure problem has been tackled considering both the parent selection and the replacement phases of a SSGA. In fact, different studies have shown that improved performance in SSGAs on generational GAs is because of their higher selection pressure and changes in the exploration/exploitation balance caused by using different parent selection and replacement strategies, and is not because of the use of an overlapping model (De Jong and Sarma, 1993).

There are parent selection mechanisms and replacement strategies that:

- promote diversity only, or
- select pressure only, or
- both diversity and selection pressure are included.

Different examples of these mechanisms are shown in Sections B.1 and B.2, respectively.

B.1. Parent selection mechanisms

In this section, we describe different parent selection mechanisms that belong to each one of the aforementioned categories.

B.1.1. Diversity only

Random Selection (RS). RS selects an individual at random from the population.

B.1.2. Selection pressure only

Tournament Selection (TS). TS is one of the more commonly used parent selection schemes, perhaps because of its simplicity. The basic idea of TS scheme is quite straightforward. A group of n_T individuals is selected randomly from the population. The individuals in this group are then compared with each other, with the fittest among the group becoming the selected individual. Typical

Table 9
Abbreviation index

Abbreviation	Meaning	Abbreviation	Meaning
(A)LEP	*Evolutionary Programming* defined in Section 7.3	NAM	*Negative Assortative Mating* (Section 5)
CHC-SW-P_G	*Combination of CHC and SW* defined in Section 7.3	N_{F}	*Size of* G_{F} (Section 6)
CLPSO	*Comprehensive Learning Particle Swarm Optimizer* (Section 7.3)	N_{M}	*Size of* G_{M} (Section 6)
CMAES	*Evolution Strategy with Covariance Matrix Adaptation* (Section 7.3)	$N_{\mathrm{F}}^{\mathrm{G}}$	*Size of* G_{F} *of the Global Genetic Algorithm* (Section 7)
DE	*Differential Evolution* (Section 7.3)	$N_{\mathrm{M}}^{\mathrm{G}}$	*Size of* G_{M} *of the Global Genetic Algorithm* (Section 7)
DOS	*Diversity-Oriented Selection* (Appendix B.1)	$N_{\mathrm{F}}^{\mathrm{L}}$	*Size of the* G_{F} *of the Local Genetic Algorithm* (Section 7)
DS	*Disruptive Selection* (Appendix B.1)	$N_{\mathrm{M}}^{\mathrm{L}}$	*Size of the* G_{M}*of the Local Genetic Algorithm* (Section 7)
EP	*Evolutionary Programming* (Fogel, 1995)	P	*Population of a GA*
ES	*Evolution Strategy* (Schwefel, 1995)	PBX-α	*Parent-Centric Crossover Operator* defined in Section 2.1
ESA	*Enhanced Simulated Annealing* (Section 7.3)	PCCO	*Parent-Centric Crossover Operator* (Section 2)
FIFO	*First-In-First-Out Replacement Strategy* (Appendix B.2)	RCGA	*Real-Coded Genetic Algorithm*
FMD	*Female and Male Differentiation process* (Section 6)	RCMA-XHC	*A Memetic RCGA* defined in Section 7.3
FMD-U&N	*FMD with UFS and NAM selection mechanisms* (Section 6)	RS	*Random Selection* (Appendix B.1)
FUSS	*Fitness Uniform Selection Scheme* (Appendix B.1)	RTS	*Restricted Tournament Selection* (Appendix B.2)
G3-PCX-λ	A *RCGA* defined in Section 7.3	RW	*Replace Worst* (Appendix B.2)
GA	*Genetic Algorithm* (Goldberg, 1989; Holland, 1975)	SPC-PNX-N	*A RCGA* defined in Section 7.3
G_{F}	*Group of Female Parents* (Section 6)	SSGA	*Steady-State Genetic Algorithm* (Section 3)
GL-P_G	*Combination of a Global and a Local Algorithm* (Section 7)	TS	*Tournament Selection* (Appendix B.1)
G_{M}	*Group of Male Parents* (Section 6)	UFS	*Uniform Fertility Selection* (Section 4)
N	*Size of the population of a GA*		

implementation of TS involves picking only two individuals for comparison (*binary* TS).

B.1.3. Diversity and selection pressure

This group is represented by three instances because many selection mechanisms presented in the GA literature may be included in it.

Disruptive Selection (DS) (Kuo and Hwang, 1996). Unlike conventional selection mechanisms, DS devotes more trials to both better and worse solutions than it does to moderate solutions. This is carried out by modifying the objective function of each chromosome, C, as follows:

$$f'(C) = |f(C) - \bar{f}|,$$

where $\bar{f}$ is the average value of the objective function of the individuals in the population. Tournament selection may be applied considering this new objective function to select an individual from the population.

Fitness uniform selection scheme (FUSS) (Hutter, 2002). FUSS generates selection pressure towards sparsely populated fitness regions, not necessarily towards higher fitness. It is defined as follows: if $f_{\min}$ and $f_{\max}$ are the lowest and highest fitness values in the current population, respectively, we select a fitness value uniformly in the interval $[f_{\min}, f_{\max}]$. Then, the individual in the population with fitness nearest to this value is selected. FUSS results in high selection pressure towards higher fitness if there are only a few fit individuals and the selection pressure is automatically reduced when the number of fit individuals increases. In a typical FUSS population, there are many unfit and only a few fit individuals. Fit individuals are effectively favoured until the population becomes fitness uniform. Occasionally, a new higher fitness level is discovered and occupied by a new individual, which then, again, is favoured.

Diversity-oriented selection (DOS) (Shimodaira, 1999). There exists an important group of parent selection mechanisms that are defined using both the fitness function and a measure that evaluates the diversity introduced by the competing chromosomes in the population (Bonham and Parmee, 1999; De Jong et al., 2001; Ichikawa and Ishiiolland, 1993; Lee, 2003; Mori et al., 1995; Shimodaira, 1999; Toffolo and Benini, 2003). Their purpose is to favour the individuals with both high fitness function values and high diversity contributions. For example, in DOS, the chromosomes for the next generation are selected from a merged population of parents and their children. The chromosomes are selected in their fitness values order, through a selection probability that is calculated using the Hamming distance between the candidate chromosomes and the chromosome with the best fitness value, being larger for chromosomes with larger Hamming distances.

We have designed a parent selection mechanism for SSGAs that follows the main idea of DOS. In particular, we run two times binary TS to choose two highly-fit chromosomes; then the one more dissimilar to the best chromosome in the population, according to the Euclidean distance, is selected.

Table 10
Used test functions $f_{\mathrm{sph}} - f_{\mathrm{QNoise}}$ are unimodal and $f_{\mathrm{Ras}} - f_{\mathrm{Schaffer}}$ are multimodal

Test function	D	Domain	Init	f^*
Sphere $f_{\mathrm{sph}} = \sum_{i=1}^{n} x_i^2$	25	$[-5.12,\ 5.12]$	$[4,5]$	0
Cigar-Tablet $f_{\mathrm{cig}} = x_1^2 + 1e8 \cdot x_n^2 + 1e4 \cdot \sum_{i=2}^{n-1} x_i^2$	25	$[-7,7]$	$[5,7]$	0
Schwefel's Problem 2.22 $f_{\mathrm{Sch2.22}} = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	25	$[-10,10]$	$[8,10]$	0
Schwefel's Problem 2.21 $f_{\mathrm{Sch2.21}} = \max\{\lvert x_i \rvert, 1 \leqslant i \leqslant n\}$	100	$[-100,100]$	$[80,100]$	0
Generalized Rosenbrock's Function $f_{\mathrm{Ros}} = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	25	$[-5.12,5.12]$	$[-5,-4]$	0
Schwefel's Problem 1.2 $f_{\mathrm{Sch}}(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	25	$[-65.536,65.536]$	$[60,65]$	0
Quartic Function i.e. Noise $f_{\mathrm{QNoise}} = random[0,1) + \sum_{i=1}^{n} i \cdot x_i^4$	25	$[-1.28,1.28]$	$[-1.28,\ 1.28]$	0
Generalized Rastrigin's Function $f_{\mathrm{Ras}}(x) = 10 \cdot n + \sum_{i=1}^{n} x_i^2 - 10 \cdot \cos(2\pi \cdot x_i)$	25	$[-5.12,\ 5.12]$	$[4,5]$	0
Rotated Generalized Rastrigin's Function $f_{\mathrm{RRas}} = 10n + \sum_{i=1}^{n} (z_i^2 - 10\cos(2\pi \cdot z_i))$ with $z = Ax$ and $A_{ij} = \begin{cases} 4/5 & \text{if } i = j \\ (-1)^{i+1} \cdot 3/5 & \text{if } \lvert i - j \rvert = 1 \\ 0 & \text{otherwise} \end{cases}$	25	$[-5.12,5.12]$	$[4,5]$	0
Scaled Generalized Rastrigin's Function $f_{\mathrm{SRas}} = 10n + \sum_{i=1}^{n} \left(\left(10^{\frac{i-1}{n-1}} \cdot x_i \right)^2 - 10\cos\left(2\pi \cdot 10^{\frac{i-1}{n-1}} \cdot x_i \right) \right)$	25	$[-5.12,5.12]$	$[4,5]$	0
Generalized Griewank Function $f_{\mathrm{Gri}}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \Pi_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	15	$[-600,600]$	$[580,600]$	0
Generalized Schwefel's Problem 2.26 $f_{\mathrm{MSch}} = -\sum_{i=1}^{n} x_i \cdot \sin\left(\sqrt{\lvert x_i \rvert} \right)$	30	$[-500,500]$	$[-500,300]$	−12.569.5
Expanded F10 $EF10 = F10(x_n, x_1) + \sum_{i=1}^{n-1} F10(x_i, x_{i+1})$ with $F10(x,y) = (x^2 + y^2)^{0.25} \cdot (\sin^2(50 \cdot (x^2 + y^2)^{0.1}) + 1)$	15	$[-100,100]$	$[-100,\ 100]$	0
Composed fGri (fRos (x)) $F8F2 = f_{\mathrm{Gri}}(f_{\mathrm{Ros}}(x_n, x_1)) + \sum_{i=1}^{n-1} f_{\mathrm{Gri}}(f_{\mathrm{Ros}}(x_i, x_{i+1}))$	10	$[-5,5]$	$[-5,5]$	0
Bohachevsky $f_{\mathrm{Boh}} = \sum_{i=1}^{n-1} (x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi \cdot x_i) - 0.4\cos(4\pi \cdot x_{i+1}) + 0.7)$	25	$[-15,15]$	$[10,15]$	0
Schaffer $f_{\mathrm{Schaffer}} = \sum_{i=1}^{n-1} (x_i^2 + x_{i+1}^2)^{0.25} \cdot [\sin^2(50 \cdot (x_i^2 + x_{i+1}^2)^{0.1}) + 1]$	25	$[-100,100]$	$[50,100]$	0

B.2. Replacement strategies

Next, we review three different replacement strategies aimed to either promote diversity, or produce selection pressure, or ensure both of them.

B.2.1. Diversity only

First-in-first-out strategy (FIFO) (De Jong and Sarma, 1993). The offspring replaces the oldest individual in the population.

B.2.2. Selection pressure only

Replace worst strategy (RW). This strategy replaces the worst individual in the population only if the new individual is better. Goldberg et al. (Goldberg and Deb, 1991) suggested that the deletion of the worst individuals induced a high selection pressure, even when the parents were selected randomly.

B.2.3. Diversity and selection pressure

Restricted tournament selection (RTS) (Harik, 1995). Let us suppose that A is the individual to be included in the population. Then, RTS scans ω (*window size*) members of the population and picks the individual that most closely resembles A from those ω elements. A then competes with this element, and if A wins, it is allowed to enter the population. RTS is a *crowding method* (Cedeño et al., 1995; De Jong, 1975; Mahfoud, 1995). New individuals are more likely to replace existing individuals in the population that are similar to themselves based on genotypic similarity. In this manner, the population does not build up an excess of similar solutions.

Appendix C. Test suite

The test suite that we have used for the experiments consists of sixteen test functions and two real-world problems. They are described in Sections C.1 and C.2, respectively.

C.1. Test functions

Table 10 describes the sixteen test functions considered in the experiments. It includes number of decision variables of each problem (D), the decision variables' domains, the ranges where the population of the algorithms is initialized, and the fitness value of the optimal solution (f^*). In addition, it indicates the functions which are non-separable.

We have applied a skewed initialization that forces the initial population to be away from the global basin due to two reasons (Deb et al., 2002):

- To prevent the advantage enjoyed by algorithms which have inherent tendency to create solutions near the centroid of the parents.
- To make sure that an algorithm must overcome a number of local minima to reach the global basin when dealing with multimodal functions.

C.2. Real-world problems

C.2.1. Systems of linear equations (Eshelman et al., 1997)

The problem to be solved is to obtain the elements of a vector X, given the matrix A and vector B in the expression: $A \cdot X = B$. The evaluation function used for these experiments is:

$$P_{\text{sle}}(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} |(a_{ij} \cdot x_j) - b_j|.$$

We have studied an example of a ten-parameter problem. We have considered that $-127 \leqslant x_i \leqslant 127$ and the following matrices:

$$\begin{vmatrix} 5 & 4 & 5 & 2 & 9 & 5 & 4 & 2 & 3 & 1 \\ 9 & 7 & 1 & 1 & 7 & 2 & 2 & 6 & 6 & 9 \\ 3 & 1 & 8 & 6 & 9 & 7 & 4 & 2 & 1 & 6 \\ 8 & 3 & 7 & 3 & 7 & 5 & 3 & 9 & 9 & 5 \\ 9 & 5 & 1 & 6 & 3 & 4 & 2 & 3 & 3 & 9 \\ 1 & 2 & 3 & 1 & 7 & 6 & 6 & 3 & 3 & 3 \\ 1 & 5 & 7 & 8 & 1 & 4 & 7 & 8 & 4 & 8 \\ 9 & 3 & 8 & 6 & 3 & 4 & 7 & 1 & 8 & 1 \\ 8 & 2 & 8 & 5 & 3 & 8 & 7 & 2 & 7 & 5 \\ 2 & 1 & 2 & 2 & 9 & 8 & 7 & 4 & 4 & 1 \end{vmatrix} \begin{vmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{vmatrix} = \begin{vmatrix} 40 \\ 50 \\ 47 \\ 59 \\ 45 \\ 35 \\ 53 \\ 50 \\ 55 \\ 40 \end{vmatrix}.$$

Clearly, the best value for this objective function is $P_{\text{sle}}(x^*) = 0$. Inter-parameter linkage (i.e., non-linearity) is controlled easily in systems of linear equations; their non-linearity does not deteriorate as increasing numbers of parameters are used, and they have proven to be quite difficult.

The initialization of the population of the algorithms have been made in the range $[-120, -100]$.

Table 11
Achieve results with different combinations of values for N_F and N_M

N_F	N_M	f_{sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}
1	25	1.90e+002	3.36e+005	5.70e+006	5.04e+002	5.90e+002	7.59e+003
1	50	3.19e+001	1.85e+004	1.94e+006	4.61e+002	1.46e+002	4.11e+003
1	100	2.86e−005	4.92e+001	2.29e+005	4.41e+002	2.80e+001	2.21e+003
1	200	2.46e−144	2.13e+001	2.61e+004	4.34e+002	3.57e+000	9.85e+002
1	300	4.92e−116	1.59e+001	8.34e+003	4.16e+002	5.21e+000	7.03e+002
1	400	5.10e−095	1.95e+001	3.06e+003	4.19e+002	7.44e−002	3.90e+002
5	25	9.17e+001	1.75e+005	1.17e+005	4.00e+002	3.14e+002	2.70e+003
5	50	6.21e−146	2.89e+001	2.81e+003	3.50e+002	1.04e+001	5.94e+002
5	100	9.98e−187	1.56e+000	1.74e−009	2.90e+002	1.27e−002	1.64e+002
5	200	2.76e−129	2.33e+000	3.09e−009	2.22e+002	1.41e−002	8.86e+001
5	300	8.99e−098	7.67e+000	1.05e−006	2.06e+002	1.18e−002	5.09e+001
5	400	2.53e−079	1.16e+001	7.60e−005	1.85e+002	1.49e−002	3.82e+001
25	25	9.24e−007	3.27e+003	3.69e+003	2.02e+002	2.22e+001	5.78e+002
25	50	1.64e−130	1.33e+001	1.01e−012	1.34e+002	1.07e−002	1.47e+002
25	100	3.31e−104	9.48e+000	2.68e−009	6.62e+001	6.70e−003	6.50e+001
25	200	1.05e−074	9.80e+000	6.66e−005	4.48e+001	2.56e−003	3.85e+001
25	300	1.24e−059	1.08e+001	3.82e−003	2.37e+001	3.45e−003	2.10e+001
25	400	4.18e−050	1.23e+001	3.43e−002	1.84e+001	2.51e−003	1.46e+001
50	25	5.98e−076	2.53e+001	1.74e−003	1.21e+002	2.67e+000	4.50e+002
50	50	6.19e−095	1.46e+001	3.54e−010	4.97e+001	6.45e−003	8.96e+001
50	100	1.29e−072	1.37e+001	3.85e−006	2.39e+001	4.73e−003	5.61e+001
50	200	1.15e−053	1.38e+001	4.15e−003	1.26e+001	2.86e−003	2.77e+001
50	300	9.83e−044	1.45e+001	9.00e−002	1.04e+001	2.71e−003	1.43e+001
50	400	1.82e−037	1.51e+001	5.36e−001	7.80e+000	1.38e−003	8.59e+000
100	25	1.15e−088	2.45e+001	2.45e−009	5.37e+001	1.84e−002	2.50e+002
100	50	1.29e−071	1.55e+001	1.00e−006	1.94e+001	5.07e−003	1.19e+002
100	100	2.28e−049	1.60e+001	2.20e−003	1.10e+001	3.30e−003	3.32e+001
100	200	5.59e−037	1.65e+001	2.34e−001	6.27e+000	1.58e−003	1.79e+001
100	300	9.01e−031	1.71e+001	1.45e+000	5.32e+000	9.37e−004	1.33e+001
100	400	1.06e−026	1.75e+001	4.58e+000	4.24e+000	1.18e−003	1.05e+001
200	25	2.19e−078	2.53e+001	3.96e−007	1.86e+001	6.99e−003	1.49e+002
200	50	8.56e−052	1.65e+001	6.07e−004	7.92e+000	4.04e−003	6.90e+001
200	100	6.31e−035	1.73e+001	9.10e−002	5.15e+000	2.76e−003	2.02e+001
200	200	1.22e−024	1.83e+001	3.40e+000	3.58e+000	1.53e−003	1.61e+001
200	300	9.95e−021	1.88e+001	1.45e+001	6.64e+000	1.04e−003	1.07e+001
200	400	2.95e−018	1.91e+001	3.12e+001	1.92e+001	4.93e−004	5.45e+000
300	25	3.22e−068	1.85e+001	1.37e−005	1.13e+001	4.78e−003	1.28e+002
300	50	2.72e−044	1.70e+001	1.40e−002	6.81e+000	3.74e−003	5.04e+001
300	100	1.09e−027	1.81e+001	6.89e−001	3.33e+000	1.23e−003	1.94e+001
300	200	5.19e−020	1.89e+001	1.18e+001	6.32e+000	9.86e−004	1.16e+001
300	300	3.17e−016	1.96e+001	3.76e+001	3.80e+001	5.42e−004	9.13e+000
300	400	2.53e−014	1.98e+001	7.35e+001	6.58e+001	5.92e−004	7.45e+000
400	25	3.47e−061	1.87e+001	2.96e−004	8.08e+000	7.44e−003	1.10e+002
400	50	1.28e−039	1.74e+001	9.27e−002	4.16e+000	2.86e−003	3.73e+001
400	100	1.71e−024	1.85e+001	2.53e+000	2.60e+000	3.10e−003	2.43e+001
400	200	7.73e−017	1.94e+001	2.35e+001	3.45e+001	6.41e−004	1.01e+001
400	300	4.93e−014	1.99e+001	7.18e+001	6.72e+001	3.48e−004	8.83e+000
400	400	5.35e−012	2.03e+001	1.38e+002	9.71e+001	1.77e−003	7.84e+000

C.2.2. Frequency modulation sound parameter identification (Tsutsui and Fujimoto, 1993)

The problem is to specify six parameters a_1, w_1, a_2, w_2, a_3, w_3 of the frequency modulation sound model represented by

$$y(t) = a_1 \cdot \sin(w_1 \cdot t \cdot \theta + a_2 \cdot \sin(w_2 \cdot t \cdot \theta + a_3 \cdot \sin(w_3 \cdot t \cdot \theta))),$$

with $\theta = (2 \cdot \pi/100)$. The fitness function is defined as the sum of square errors between the evolved data and the model data as follows:

$$P_{\text{fms}}(a_1, w_1, a_2, w_2, a_3, w_3) = \sum_{t=0}^{100} (y(t) - y_0(t))^2,$$

where the model data are given by the following equation:

Table 12
Comparison of the hybrid RCGA with other continuous metaheuristics ($f_{\text{sph}} - f_{\text{Sch}}$)

Algorithm	f_{sph}	f_{cig}	$f_{\text{Sch2.22}}$	$f_{\text{Sch2.21}}$	f_{Ros}	f_{Sch}
SPC-PNX-40	5.42e−040+	8.49e−037+	4.95e−025+	8.92e+001+	2.22e+001+	2.43e−001+
SPC-PNX-60	2.37e−029+	1.35e−027+	1.20e−017+	8.62e+001+	1.99e+001+	9.75e−001+
SPC-PNX-100	8.62e−018+	4.91e−015+	4.65e−010+	7.62e+001+	2.20e+001+	1.62e+001+
SPC-PNX-200	2.94e−008+	5.91e−005+	2.96e−004+	6.74e+001+	2.09e+001+	3.55e+002+
G3-PCX-1	4.21e−203−	6.81e+001+	1.63e+002+	9.26e+001+	9.83e+000+	3.74e−029−
G3-PCX-2	3.55e−178−	5.41e+001+	1.56e+002+	9.23e+001+	3.30e+000+	5.78e−031−
G3-PCX-3	8.10e−148−	7.40e+001+	1.54e+002+	9.14e+001+	4.84e−001∼	8.88e−032−
G3-PCX-4	4.47e−127+	7.60e+001+	1.46e+002+	9.07e+001+	7.18e−001∼	2.95e−030−
ALEP-1	1.15e−003∼	1.62e+003+	1.47e+000+	9.77e+001+	4.05e+001+	1.35e+003+
ALEP-2	1.55e+000+	7.47e+004+	9.37e+000+	9.38e+001+	9.92e+002∼	1.51e+004+
LEP-1	3.31e−002+	1.08e+003+	2.65e+000+	9.89e+001+	4.17e+001+	1.70e+003+
LEP-2	3.60e+000+	1.39e+005+	2.21e+001+	9.45e+001+	7.50e+002+	3.21e+004+
CMAES	3.18e−293−	5.25e−243−	1.32e−124−	3.17e−005−	1.59e−001−	8.73e−267−
RCMA-XHC	9.47e−100+	5.73e−094+	3.56e−041+	7.67e+001+	2.85e+000+	9.48e−007+
CHC-SW-25	3.41e−322−	5.01e−005+	7.02e−005+	8.42e+001+	5.59e+000+	1.91e−031−
CHC-SW-50	3.80e−322−	2.98e−013+	4.79e−010+	8.31e+001+	8.99e+000+	8.51e−022−
CHC-SW-75	5.30e−205−	1.66e−022+	1.13e−015+	8.32e+001+	1.31e+001+	3.97e−012−
ESA	1.46e−023+	3.48e−018+	6.73e−009+	2.12e+001−	9.40e+000+	1.80e−005+
CLPSO	2.21e−028+	9.68e−026+	4.05e−017+	9.44e+001+	6.69e+000+	5.80e+002+
CLPSO-30	1.03e−011+	2.71e−007+	1.07e−005+	9.61e+001+	2.34e+001+	2.47e+003+
DE	2.86e+000+	1.04e+005+	5.21e+001+	9.84e+001+	2.94e+003+	1.07e+005+
DE-60	6.81e−024+	6.68e−010+	2.63e−011+	8.97e+001+	3.38e+000+	2.28e−004+
DE-100	1.03e−011+	1.83e−007+	4.34e−005+	8.71e+001+	8.52e+000+	2.84e+000+
GL-25	4.36e−147	1.03e−122	9.85e−077	3.25e+001	8.22e−001	3.43e−009

Table 13
Comparison of the hybrid RCGA with other continuous metaheuristics ($f_{\text{QNoise}} - f_{\text{MSch}}$)

Algorithm	f_{QNoise}	f_{Ras}	f_{RRas}	f_{SRas}	f_{Gri}	f_{MSch}
SPC-PNX-40	8.33e−003+	1.29e+002+	1.66e+002+	1.05e+002+	2.62e−002+	−9.16e+003∼
SPC-PNX-60	7.80e−003+	7.30e+001+	8.81e+001+	6.09e+001+	1.96e−002+	−9.47e+003∼
SPC-PNX-100	9.81e−003+	3.68e+001+	3.80e+001+	3.12e+001+	2.02e−002+	−1.01e+004−
SPC-PNX-200	1.53e−002+	3.13e+001+	4.76e+001+	2.51e+001+	9.25e−003+	−1.12e+004−
G3-PCX-1	1.03e+000+	4.86e+002+	4.78e+002+	1.56e+003+	1.44e−001+	−7.89e+003+
G3-PCX-2	5.83e−001+	4.85e+002+	4.72e+002+	1.18e+003+	7.32e−002+	−7.97e+003+
G3-PCX-3	3.41e−001+	4.88e+002+	4.76e+002+	9.94e+002+	4.98e−002+	−8.20e+003+
G3-PCX-4	2.63e−001+	4.83e+002+	4.73e+002+	7.42e+002+	4.51e−002+	−8.25e+003+
ALEP-1	5.92e−002+	4.84e+001+	7.72e+001+	5.92e+001+	6.79e−002+	−9.82e+003−
ALEP-2	9.87e−002+	4.59e+001+	1.26e+002+	1.09e+002+	8.66e+001+	−8.80e+003+
LEP-1	4.55e−002+	6.94e+001+	9.09e+001+	1.18e+002+	1.12e−001+	−9.31e+003∼
LEP-2	5.14e−002+	8.39e+001+	1.65e+002+	2.40e+002+	2.38e+002+	−7.30e+003+
CMAES	2.23e−001+	5.24e+001+	5.15e+001+	6.77e+001+	4.14e−003+	−6.89e+003+
RCMA-XHC	5.55e−003+	1.24e+001∼	1.93e+001∼	1.05e+001−	4.71e−002+	−1.06e+004−
CHC-SW-25	1.79e−002+	2.61e+001+	3.60e+001+	2.56e+001+	5.27e−003+	−1.11e+004−
CHC-SW-50	1.10e−002+	2.72e+001+	3.42e+001+	2.29e+001+	4.93e−003+	−1.11e+004−
CHC-SW-75	7.93e−003+	2.15e+001+	2.76e+001+	2.16e+001+	5.91e−003+	−1.08e+004−
ESA	2.37e−002+	1.99e−002−	2.46e+001+	0.00e+000−	3.14e−002+	−5.58e+003+
CLPSO	6.12e−003+	7.20e+000−	3.64e+001+	4.40e+000−	9.86e−004+	−1.20e+004−
CLPSO-30	1.05e−002+	5.57e−001−	3.29e+001+	2.32e−006−	6.09e−009+	−1.25e+004−
DE	6.16e−001+	1.63e+002+	1.93e+002+	2.41e+002+	3.37e−001+	−6.28e+003+
DE-60	7.00e−003+	1.23e+002+	1.30e+002+	1.23e+002+	4.43e−004+	−6.70e+003+
DE-100	1.40e−002+	1.31e+002+	1.39e+002+	1.32e+002+	3.24e−003+	−5.85e+003+
GL-25	1.67e−003	1.38e+001	1.90e+001	1.35e+001	2.26e−018	−9.36e+003

Table 14
Comparison of the hybrid RCGA with other continuous metaheuristics ($P_{\text{sle}} - f_{\text{Schaffer}}$)

Algorithm	P_{sle}	EF10	F8F2	P_{Sound}	f_{Boh}	f_{Schaffer}
SPC-PNX-40	2.64e+002+	3.31e+000+	8.48e−001∼	8.80e+000+	2.41e+000+	2.65e+001+
SPC-PNX-60	1.71e+002+	4.10e−001+	7.78e−001∼	7.26e+000+	8.17e−001+	4.20e+000+
SPC-PNX-100	1.11e+002+	5.65e−003−	9.34e−001+	3.96e+000+	1.14e−001+	1.03e−001−
SPC-PNX-200	4.32e+001+	2.86e−002−	1.08e+000+	1.86e+000+	9.42e−006+	2.06e+000−
G3-PCX-1	3.07e+002+	9.62e+001+	5.07e+000+	2.20e+001+	1.53e+001+	2.42e+002+
G3-PCX-2	1.89e+002+	9.50e+001+	3.69e+000+	2.21e+001+	1.33e+001+	2.41e+002+
G3-PCX-3	1.30e+002+	9.07e+001+	2.90e+000+	2.28e+001+	1.27e+001+	2.39e+002+
G3-PCX-4	1.22e+002+	8.58e+001+	2.54e+000+	2.16e+001+	1.25e+001+	2.43e+002+
ALEP-1	3.62e+002+	4.14e+000+	6.91e−001∼	1.39e+001+	2.16e+000+	5.26e+001+
ALEP-2	1.33e+003+	5.93e+001+	1.74e+000+	2.22e+001+	1.59e+002+	1.39e+002+
LEP-1	1.73e+002+	8.07e+000+	8.63e−001∼	7.88e+000+	6.60e+000+	5.20e+001+
LEP-2	1.76e+003+	8.30e+001+	9.20e−001+	2.18e+001+	3.55e+002+	1.62e+002+
CMAES	3.23e−013−	2.02e+001+	1.13e+000+	2.24e+001+	2.65e+000+	3.24e+001+
RCMA-XHC	1.87e+002+	5.62e−001+	3.92e−001−	7.73e+000+	4.13e−002+	1.09e+001+
CHC-SW-25	8.13e+001+	1.31e−002−	9.25e−001+	1.59e+000∼	1.46e−014+	2.04e+000−
CHC-SW-50	6.98e+001+	4.43e−007−	8.80e−001+	6.63e−001∼	7.70e−015+	6.73e−002−
CHC-SW-75	8.60e+001+	3.88e−007−	8.78e−001+	2.51e−001∼	0.00e+000∼	1.30e−002−
ESA	1.52e+003+	9.06e−009−	4.23e−001−	2.46e+001+	0.00e+000∼	2.96e−004−
CLPSO	1.37e+002+	9.86e−004−	1.23e−001−	4.16e+000+	0.00e+000∼	1.62e−002−
CLPSO-30	1.90e+002+	1.86e−002−	6.06e−002−	2.58e+000+	2.76e−009+	2.36e+000−
DE	7.10e+002+	9.58e+000+	1.46e+000+	1.27e+001+	8.11e+001+	1.33e+002+
DE-60	1.26e−001−	5.62e−008−	1.58e+000+	1.52e+000∼	0.00e+000∼	3.54e−002−
DE-100	1.77e+000−	1.14e−002−	1.70e+000+	1.66e+000+	2.05e−008+	1.81e+000−
GL-25	4.70e+000	1.99e−001	7.48e−001	2.96e−001	0.00e+000	2.70e+000

$$y_0(t) = 1.0 \cdot \sin(5.0 \cdot t \cdot \theta + 1.5 \cdot \sin(4.8 \cdot t \cdot \theta + 2.0 \cdot \sin(4.9 \cdot t \cdot \theta))).$$

Each parameter is in the range [−6.4, −6.35] (the initial population was generated considering this range). This is a highly complex multi-modal problem, having strong epitasis, with minimum value $P_{\text{fms}}(x^*) = 0$.

Appendix D. Results of the FMD-U&N algorithm

See Table 11.

Appendix E. Results of the comparison

Tables 12–14 present the results of the compared metaheuristics. They are the average of the best obtained fitness value of 50 executions. A *t*-test was applied to detect significant differences between GL-25 and the other algorithms.

References

Bäck, T., 1994. Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In: Proc. of the First IEEE Conference on Evolutionary Computation, pp. 57–62.

Bäck, T., 1996. Evolutionary Algorithms in Theory and Practice. Oxford University Press.

Ballester, P.J., Carter, J.N., 2003. Real-parameter genetic algorithms for finding multiple optimal solutions in multi-modal optimization. In: Cantú-Paz, Erick et al. (Eds.), Proc. of the Genetic and Evolutionary Computation Conference, LNCS, vol. 2723. Springer, pp. 706–717.

Ballester, P.J., Carter, J.N., 2004. An effective real-parameter genetic algorithm with parent centric normal crossover for multimodal optimisation. In: Proc. of the Genetic and Evolutionary Computation ConferenceLNCS, vol. 3102. Springer, pp. 901–913.

Bandyopadhyay, S., Pal, S.K., Maulik, U., 1998. Incorporating chromosome differentiation in genetic algorithms. Information Sciences 104, 293–319.

Beyer, H.-G., Deb, K., 2001. On self-adaptive features in real-parameter evolutionary algorithms. IEEE Transactions on Evolutionary Computation 5 (3), 250–270.

Beyer, H.-G., Schwefel, H.-P., 2002. Evolution strategies. Natural Computing 1, 3–52.

Bonham, C.R., Parmee, I.C., 1999. An investigation of exploration and exploitation within cluster oriented genetic algorithms (COGAs). In: Proc. of the Genetic and Evolutionary Computation Conference 1999. Morgan Kaufmann, San Francisco, California, pp. 1491–1497.

Branke, J., Cutaia, M., Dold, H., 1999. Reducing genetic drift in steady state evolutionary algorithms. In: Banzhaf, W. et al. (Eds.), Proc. of the Genetic and Evolutionary Computation Conference. Morgan Kaufmann, San Francisco, California, pp. 68–74.

Cedeño, W., Vemuri, V., Slezak, T., 1995. Multi-niche crowding in genetic algorithms and its application to the assembly of

DNA restriction-fragments. Evolutionary Computation 2 (4), 321–345.

Chelouah, R., Siarry, P., 2000. A continuous genetic algorithm designed for the global optimization of multimodal functions. Journal of Heuristic 6 (2), 191–213.

Chelouah, R., Siarry, P., 2003. Genetic and Nelder–Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. European Journal of Operational Research 148, 335–348.

Craighurst, R., Martin, W., 1995. Enhancing GA performance through crossover prohibitions based on ancestry. In: Eshelman, L.J. (Ed.), Proc. 6th International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo, CA, pp. 130–137.

Davis, L., 1991. Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York.

Deb, K., 2001. Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, New York.

Deb, K., 2005. A population-based algorithm-generator for real-parameter optimization. Soft Computing 9 (4), 236–253.

Deb, K., Agrawal, R.B., 1995. Simulated binary crossover for continuous search space. Complex System 9, 115–148.

Deb, K., Beyer, H., 2001. Self-adaptive genetic algorithms with simulated binary crossover. Evolutionary Computation Journal 9 (2), 195–219.

Deb, K., Anand, A., Joshi, D., 2002. A computationally efficient evolutionary algorithm for real-parameter evolution. Evolutionary Computation Journal 10 (4), 371–395.

De Jong, K.A., 1975. An analysis of the behavior of a class of genetic adaptive systems. Doctoral dissertation, University of Michigan, Ann Arbor. Dissertation Abstracts International 36(10), 5140B (University Microfilms No. 76-9381).

De Jong, K.A., Sarma, J., 1993. Generation gaps revisited. In: Whitley, L.D. (Ed.), Foundations of Genetic Algorithms, vol. 2. Morgan Kaufmann, San Mateo, California, pp. 19–28.

De Jong, K.A., Spears, W.M., 1992. A formal analysis of the role of multi-point crossover in genetic algorithms. Annals of Mathematics and Artificial Intelligence 5 (1), 1–26.

De Jong, E.D., Watson, R.A., Pollack, J.B., 2001. Reducing bloat and promoting diversity using multi-objective methods. In: Spector, L. et al. (Eds.), Proc. of the Genetic and Evolutionary Computation Conference. Morgan Kaufmann, San Francisco, California, pp. 11–18.

Eberhart, R.C., Kennedy, J., 1995. A new optimizer using particle swarm theory. In: Proc. of the 6th International Symposium on Micromachine and Human Science, Nagoya, Japan, pp. 39–43.

Eiben, A.E., Hinterding, R., Michalewicz, Z., 1999. Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation 3 (2), 124–141.

Eshelman, L.J., 1991. The CHC adaptive search algorithm: How to have safe search when engaging in non-traditional genetic recombination. In: Rawlin, G.J.E. (Ed.), Foundations of Genetic Algorithms, vol. 1. Morgan Kaufmann, San Mateo, California, pp. 265–283.

Eshelman, L.J., Schaffer, J.D., 1991. Preventing premature convergence in genetic algorithms by preventing incest. In: Belew, R., Booker, L.B. (Eds.), Proc. of the Fourth International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo, California, pp. 115–122.

Eshelman, L.J., Mathias, K.E., Schaffer, J.D., 1997. Convergence controlled variation. In: Belew, R., Vose, M. (Eds.), Foundations of Genetic Algorithms, vol. 4. Morgan Kaufmann, San Mateo, California, pp. 203–224.

Fernandes, C., Rosa, A., 2001. A Study on non-random mating and varying population size in genetic algorithms using a royal road function. In: Proc. of the 2001 Congress on Evolutionary Computation. IEEE Press, Piscataway, New Jersey, pp. 60–66.

Fogel, D.B., 1994. Evolutionary programming: An introduction and some current directions. Statistics and Computing 4, 113–129.

Fogel, D.B., 1995. Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. IEEE Press, Piscataway, New York.

Ghosh, A., Tsutsui, S., Tanaka, H., 1998. Function optimization in nonstationary environment using steady state genetic algorithms with aging of individuals. In: Proc. of the 1998 IEEE International Conference on Evolutionary Computation. IEEE Press, New York, pp. 666–671.

Goh, K.S., Lim, A., Rodrigues, B., 2003. Sexual selection for genetic algorithms. Artificial Intelligence Reviews 19, 123–152.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA.

Goldberg, D.E., Deb, K., 1991. A comparative analysis of selection schemes used in genetic algorithms. In: Rawlins, G.J.E. (Ed.), Foundations of Genetic Algorithms. Morgan Kaufmann, San Mateo, California, pp. 69–93.

Hansen, N., Ostermeier, A., 1996. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: Proc. of the 1996 IEEE Conference on Evolutionary Computation, pp. 312–317.

Hansen, N., Ostermeier, A., 2001. Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation 9 (2), 159–195.

Hansen, N., Müller, S.D., Koumoutsakos, P., 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolutionary Computation 11 (1), 1–18.

Harik, G., 1995. Finding multimodal solutions using restricted tournament selection. In: Eshelman, L.J. (Ed.), Proc. 6th International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo, California, pp. 24–31.

Herrera, F., Lozano, M., 2000. Gradual distributed real-coded genetic algorithms. IEEE Transactions on Evolutionary Computation 4 (1), 43–63.

Herrera, F., Lozano, M., Verdegay, J.L., 1998. Tackling real-coded genetic algorithms: Operators and tools for the behavioral analysis. Artificial Intelligence Reviews 12 (4), 265–319.

Herrera, F., Lozano, M., Sánchez, A.M., 2003. A taxonomy for the crossover operator for real-coded genetic algorithms. An experimental study. International Journal of Intelligent Systems 18 (3), 309–338.

Herrera, F., Lozano, M., Sánchez, A.M., 2005. Hybrid crossover operators for real-coded genetic algorithms: An experimental study. Soft Computing 9 (4), 280–298.

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

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. The University of Michigan Press, The MIT Press, London, 1992.

Hutter, M., 2002. Fitness uniform selection to preserve genetic diversity. In: Proc. of the 2002 Congress on Evolutionary Computation. IEEE Press, Piscataway, New Jersey, pp. 783–788.

Ichikawa, Y., Ishiiolland, Y., 1993. Retaining diversity of genetic algorithms for multivariable optimization and neural network learning. In: Proc. IEEE International Conference on Neural Networks, San Francisco, California, pp. 1110–1114.

Kazarlis, S.A., Papadakis, S.E., Theocharis, J.B., Petridis, V., 2001. Microgenetic algorithms as generalized hill-climbing operators for GA Optimization. IEEE Transaction on Evolutionary Computation 5 (3), 204–217.

van Kemenade, C.H.M., Kok, J.N., Eiben, A.E., 1995. Raising GA performance by simultaneous tuning of selection pressure and recombination disruptiveness. In: Proc. of the 2nd IEEE World Conference on Evolutionary Computation. IEEE Press, pp. 345–351.

Kennedy, J., Eberhart, R.C., 1995. Particle swarm optimization. In: Proc. of the IEEE International Conference on Neural Networks, Piscataway, New Jersey, pp. 1942–1948.

Kirkpatric, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220 (4598), 671–680.

Kita, H., 2001. A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms. Evolutionary Computation Journal 9 (2), 223–241.

Kuo, T., Hwang, S.Y., 1996. A genetic algorithm with disruptive selection. IEEE Transactions on Systems, Man, and Cybernetics 26 (2), 299–307.

Lee, C.-Y., 2003. Entropy-Boltzmann selection in the genetic algorithms. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 33 (1), 138–149.

Lee, C.-Y., Yao, X., 2004. Evolutionary programming using mutations based on the Levy probability distribution. IEEE Transactions on Evolutionary Computation 8 (1), 1–13.

Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S., 2004a. Particle swarm optimization algorithms with novel learning strategies. In: Proc. of the IEEE International Conference on Systems Man and Cybernetics, The Netherlands, pp. 3659–3664.

Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S., 2004b. Evaluation of comprehensive learning particle swarm optimizer. Lecture Notes in Computer Science 3316, 230–235.

Lozano, M., Herrera, F., Krasnogor, N., Molina, D., 2004. Real-coded memetic algorithms with crossover hill-climbing. Evolutionary Computation Journal 12 (3), 273–302.

Mahfoud, S.W., 1995. Niching methods for genetic algorithms. IlliGAL report 95001, University of Illinois at Urbana-Champaign, IL, Illinois Genetic Algorithms Laboratory.

Matsui, K., 1999. New selection method to improve the population diversity in genetic algorithms. Proc. IEEE International Conference on Systems, Man, and Cybernetics, pp. 625–630.

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

Mori, N., Yoshida, J., Tamaki, H., Kita, H., Nichikawa, Y.A., 1995. Thermodynamical selection rule for genetic algorithms. In: Proc. of the 2nd IEEE Conference on Evolutionary Computation (ICEC'95). IEEE Press, New York, pp. 188–192.

Moscato, P.A., 1999. Memetic algorithms: A short introduction. In: Corne, D., Dorigo, M., Glower, R. (Eds.), New Ideas in Optimization. McGraw-Hill, London, pp. 219–234.

Renders, J-M., Flasse, S.P., 1996. Hybrid methods using genetic algorithms for global optimisation. IEEE Transactions on Systems, Man, and Cybernetics 26 (2), 243–258.

Schwefel, H.-P., 1995. Evolution and Optimum Seeking. Wiley, New York.

Shimodaira, H., 1996. A new genetic algorithm using large mutation rates and population-elitist selection (GALME). In: Proc. of the International Conference on Tools with Artificial Intelligence. IEEE Computer Society, pp. 25–32.

Shimodaira, H., 1999. A diversity control oriented genetic algorithm (DCGA): Development and experimental results. In: Proc. of the Genetic and Evolutionary Computation Conference 1999. Morgan Kaufmann, San Francisco, California, pp. 603–611.

Siarry, P., Berthiau, G., Durbin, F., Haussy, J., 1997. Enhanced simulated annealing for globally minimizing functions of many-continuous variables. ACM Transactions on Mathematical Software 23 (2), 209–228.

Solis, F.J., Wets, R.J.B., 1981. Minimization by random search techniques. Mathematical Operations Research 6, 19–30.

Someya, H., Yamamura, M., 2005. A robust real-coded evolutionary algorithm with toroidal search space conversion. Soft Computing 9 (4), 254–269.

Storn, R., Price, K., 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11 (4), 341–359.

Syswerda, G., 1989. Uniform crossover in genetic algorithms. In: David Schaffer, J. (Ed.), Proc. of the Third International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo, pp. 2–9.

Takahashi, O., Kobayashi, S., 2001. An adaptive neighboring search using crossover-like mutation for multi modal function optimisation. In: IEEE International Conference on Systems, Man, and Cybernetics, pp. 261–267.

Ting, C.-K., Li, S.-T., Lee, C., 2003. On the harmonious mating strategy through tabu search. Information Sciences 156, 189–214.

Toffolo, A., Benini, E., 2003. Genetic diversity as an objective in multi-objective evolutionary algorithms. Evolutionary Computation 11 (2), 151–167.

Tsutsui, S., Fujimoto, Y., 1993. Forking genetic algorithm with blocking and shrinking modes. In: Forrest, S. (Ed.), Proc. of the Fifth International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, San Mateo San Mateo, CA, pp. 206–213.

Voigt, H.M., Mühlenbein, H., Cvetkovic, D., 1995. Fuzzy recombination for the breeder genetic algorithm. In: Eshelman, L. (Ed.), Proc. of the Sixth International Conference on Genetic Algorithms. Morgan Kaufmann, San Francisco, pp. 104–111.

Whitley, D., 1989. The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In: David Schaffer, J. (Ed.), Proc. of the Third International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo, pp. 116–121.

Winter, G., Galvan, B., Alonso, S., Gonzalez, B., Jiménez, J.I., Greiner, D., 2005. A flexible evolutionary agent: Cooperation and competition among real-coded evolutionary operators. Soft Computing 9 (4), 299–323.

Yang, J.-M., Kao, C.-Y., 2000. Integrating adaptive mutations and family competition into genetic algorithms as function optimiser. Soft Computing 4, 89–102.