
Real-Coded Memetic Algorithms with Crossover Hill-Climbing

Manuel Lozano lozano@decsai.ugr.es
Dept. of Computer Science and A.I. , University of Granada, 18071 - Granada, Spain

Francisco Herrera herrera@decsai.ugr.es
Dept. of Computer Science and A.I. , University of Granada, 18071 - Granada, Spain

Natalio Krasnogor Natalio.Krasnogor@nottingham.ac.uk
Automatic Scheduling, Optimisation and Planning Group, School of Computer Science and IT, Jubilee Campus, University of Nottingham, Nottingham, NG8 1BB, United Kingdom, <http://www.cs.nott.ac.uk/~nxk/>

Daniel Molina dmolina@programador.com
Dept. of Computer Science and A.I., University of Granada, 18071 - Granada, Spain

Abstract

This paper presents a real-coded memetic algorithm that applies a crossover hill-climbing to solutions produced by the genetic operators. On the one hand, the memetic algorithm provides global search (reliability) by means of the promotion of high levels of population diversity. On the other, the crossover hill-climbing exploits the self-adaptive capacity of real-parameter crossover operators with the aim of producing an effective local tuning on the solutions (accuracy). An important aspect of the memetic algorithm proposed is that it adaptively assigns different local search probabilities to individuals. It was observed that the algorithm adjusts the global/local search balance according to the particularities of each problem instance. Experimental results show that, for a wide range of problems, the method we propose here consistently outperforms other real-coded memetic algorithms which appeared in the literature .

Keywords

Memetic algorithms, real-coding, steady-stated genetic algorithms, crossover hill-climbing.

1 Introduction

It is now well established that pure *genetic algorithms* (GAs) are not well suited to fine tuning search in complex search spaces, and that hybridisation with other techniques can greatly improve the efficiency of search (Davis, 1991; Goldberg and Voessner, 1999). GAs that have been hybridized with local search techniques (LS) are often called *memetic algorithms* (MAs) (Moscato, 1989; Moscato, 1999). MAs are evolutionary algorithms that apply a separate LS process to refine individuals (e.g. improve their fitness by hill-climbing). An important aspect concerning MAs is the trade-off between the exploration abilities of the GA, and the exploitation abilities of the LS used (Krasnogor and Smith, 2001).

Under the initial formulation of GAs, the search space solutions are coded using the binary alphabet, however, other coding types, such as real-coding, have also been

taken into account to deal with the representation of the problem. The real-coding approach seems particularly natural when tackling optimisation problems of parameters with variables in continuous domains. A chromosome is a vector of floating point numbers whose size is kept the same as the length of the vector, which is the solution to the problem. GAs based on real number representation are called *real-coded* GAs (RCGAs) (Deb, 2001; Herrera, Lozano and Verdegay, 1998).

For function optimisation problems in continuous search spaces, an important difficulty must be addressed: solutions of high precision must be obtained by the solvers (Kita, 2001). Adapted genetic operators for RCGAs have been presented to deal with this problem, which favor the local tuning of the solutions. An example is the non-uniform mutation (Michalewicz, 1992), which decreases the strength in which real-coded genes are mutated as the RCGA's execution advances. This property causes this operator to make a uniform search in the initial space and very locally at a later stage. In addition, *real-coded* MAs (RCMAs) have been proposed, which incorporate LS mechanisms for efficiently refining solutions. Most common RCMA instances use local improvement procedures (LIPs), like gradient descent or random hill-climbing, which search efficiently only for a local optimum. One commonly used formulation of MAs applies LIPs to members of the population after recombination and mutation, with the aim of exploiting the best search regions gathered during the global sampling done by the RCGAs.

One possibility of RCMA model that has received attention concerns the use of crossover-based local search algorithms (XLS). Since the crossover operator produces children around the parents, it may be used as move operator for a LS method (Deb, Anand and Joshi, 2002; Dietzfelbinger, Naudts, Van Hoyweghen and Wegener, 2003; Satoh, Yamamura and Kobayashi 1996; Yang and Kao, 2000). This is particularly attractive for real-coding since there are real-parameter crossover operators having a *self-adaptive* nature in that they can generate offspring adaptively according to the distribution of parents without any adaptive parameter (Beyer and Deb, 2001, Kita, 2000). With the passing of generations, the RCMA loses diversity, which allows the crossover to create offspring distributed densely around the parents, inducing an effective local tuning. This kind of crossover operators show promise for building effective XLS.

In this paper, we present a RCMA model that invokes real-parameter crossover hill-climbing (XHC). XHC is a particular type of XLS that allows the self-adaptive capacity of real-parameter crossover operators to be exploited inside the proper XLS, i.e., it is a *self-adaptive* crossover local search method. The mission of XHC is to obtain the best possible accuracy levels to lead the population toward the most promising search areas, producing an effective refinement on them. On the other hand, the RCMA is designed to promote high population diversity levels. It attempts to induce *reliability* in the search process by ensuring that different promising search zones are the focus of the XHC throughout the run. In addition, the RCMA employs an adaptive mechanism that determines the probability with which every solution should receive the application of XHC. In this way, it attempts to adjust the global/local search ratio (i.e., the exploration/exploitation balance) to the particular features of the problem that is being solved.

The paper is set up as follows. In Section 2, we review some important aspects of real-parameter crossover operators and describe the one used in this work. In Section 3, we deal with RCMAs, providing a classification for the different types of algorithms that appeared in the MA literature. In addition, we put special attention on RCMAs that employ crossover local search procedures. In Section 4, we describe a proposal of

XHC model. In Section 5, we present the RCMA model based on the use of the XHC. In Section 6, we described the experiments carried out in order to determine the suitability of our approach. Finally, conclusions are presented in Section 7.

2 Crossover Operators for RCGAs

The crossover operator has always been regarded as a fundamental search operator in GAs (De Jong and Spears, 1992; Kita, 2001) since it exploits information about the search space that is currently available in the population. Much effort has been put in developing sophisticated crossover operators, and as a result, many different instances have been proposed (eg. Deb, 2001; Herrera, Lozano and Verdegay, 1998; Herrera, Lozano and Sánchez, 2003). Real-coding of solutions for numerical problems offers the possibility of defining a wide variety of special real-parameter crossover operators which can take advantage of its numerical nature. Most of these crossover operators define a probability distribution of “offsprings” 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 distribute densely around the parents. On the other hand, if the parents are located far away from each other, then the offsprings will be sparsely distributed around them. Therefore, these operators fit their action range depending on the diversity of the population using specific information held by the parents. In this way, depending on the current level of diversity in the population, they may favor the production of additional diversity (divergence) or the refinement of the solutions (convergence). This behavior is achieved without requiring an external adaptive mechanism.

In fact, in the recent past, RCGAs with some of these crossovers have been demonstrated to exhibit self-adaptive behavior similar to that observed in evolution strategies and evolutionary programming approaches (Deb and Beyer, 2001; Kita, 2001). Moreover, Beyer and Deb (2001) argue that a variation operator that harness the difference of the parents in the search space is essential for the resulting evolutionary algorithm to exhibit self-adaptive behavior on the population level.

Usually, real-parameter crossover operators are applied to pairs of chromosomes, generating two offspring for each one of them, which are introduced in the population (Herrera, Lozano and Sánchez, 2003). However, *multiparent* crossover operators have been proposed, which combine the features of more than two parents for generating the offspring (Deb, Anand and Joshi, 2002; Kita, Ono and Kobayashi, 1999; Tsutsui, Yamamura and Higuchi 1999;). Furthermore, crossover operators with *multiple descendants* have been presented (Deb, Anand and Joshi 2002; Herrera, Lozano and Verdegay, 1996; Satoh, Yamamura and Kobayashi, 1996; Walters, 1998), which produce more than two offspring for each group of parents. In this case, an offspring selection strategy limits the number of offspring that will become population members. The most common strategy selects the best offspring as elements for the next population.

In this paper, we propose a new crossover operator by extending the BLX- α crossover operator presented by Eshelman and Schaffer (1993). It is called *parent-centric* BLX- α (PBX- α) and is described as follows. Let us assume that $X = (x_1 \dots x_n)$ and $Y = (y_1 \dots y_n)$ ($x_i, y_i \in [a_i, b_i] \subset \mathfrak{R}, i = 1 \dots n$) are two real-coded chromosomes that have been selected to apply the crossover operator to them. PBX- α generates (randomly) one of these two possible offspring: $Z_1 = (z_1^1 \dots z_n^1)$ or $Z_2 = (z_1^2 \dots z_n^2)$, where z_i^1 is a randomly (uniformly) chosen number from the interval $[l_i^1, u_i^1]$ with

$$l_i^1 = \max\{a_i, x_i - I \cdot \alpha\} \text{ and } u_i^1 = \min\{b_i, x_i + I \cdot \alpha\}$$

and z_i^2 is chosen from $[l_i^2, u_i^2]$ with

$$l_i^2 = \max\{a_i, y_i - I \cdot \alpha\} \text{ and } u_i^2 = \min\{b_i, y_i + I \cdot \alpha\}$$

where $I = |x_i - y_i|$. This operator has the following features:

- It is a parent-centric crossover operator because it assigns more probability for creating offspring near parents than anywhere in the search space. Studies carried out in Deb, Anand and Joshi (2002) have shown that these operators arise as a meaningful and efficient way of solving real-parameter optimization problems.
- The degree of diversity induced by PBX- α may be easily adjusted by means of varying its associated α operator parameter. The greater the α value is, the higher the variance (diversity) introduced into the population.
- This operator assigns children solutions proportional to the spread of parent solutions, thereby making a RCGA with this operator potential to exhibit self-adaptation.

3 Real-Coded Memetic Algorithms

In this paper, the combination of RCGAs with some type of LS mechanism is denoted RCMA. They are motivated by the apparent need to employ both a global and LS strategy to provide an effective global optimisation method (Hart, 1994). RCMA instances presented in the literature may be classified into two different groups:

- *Hybrid RCGAs*. They use efficient LIPs on continuous domains, e.g., hill-climbers for nonlinear optimisation (such as Quasi-Newton, conjugate gradient, SQP, random linkage, and Solis and Wets) to efficiently refine solutions. Examples are found in Hart (1994), Hart, Rosin, Belew and Morris (2000), Joines and Kay (2002), Houck, Joines, Kay and Wilson (1997), Mühlenbein, Schomisch and Born (1991), Renders and Bersini (1994), Renders and Flasse (1996), Rosin, Halliday, Hart and Belew (1997), and Zhang and Shao (2001).

A common way to use an LIP in hybrid RCGAs is to apply it to every member of each population. The resulting solutions replace the population members, and are used to generate the next population under selection and recombination (so-called *Lamarckian* evolution). An important variation on this schema is the use of a small LS probability (Hart, 1994), i.e. the LIP is only applied to members with some (typically small) fixed probability. Moreover, there exists an alternative to Lamarckian evolution, the *Darwinian* evolution, in which the solution resulting from LIP is discarded, only its fitness influences the search, changing the fitness landscape.

A different type of hybridisation between LIPs and RCGAs concerns the construction of new classes of evolutionary algorithms, which are designed using the foundational ideas of LIPs. Two examples are the evolutionary pattern search algorithm (Hart, 2001a; Hart, 2001b) and the evolutionary gradient search procedure (Salomon, 1998).

- *RCMAs with crossover-based LS algorithms*. The crossover operator is a recombination operator that produces elements around the parents. For that reason, it may be considered to be a move operator for a LS strategy. In addition, as we

have mentioned in Section 2, there are special real-parameter crossovers having a self-adaptive nature in that they can generate offspring adaptively according to the distribution of parents without any adaptive parameter. With the passing of generations, the RCGA loses diversity due to the selective pressure. Under this circumstance, these crossovers create offspring distributed densely around the parents, favoring local tuning. Therefore, this type of operators arise as appropriate candidates for building crossover-based LS algorithms (XLS). The next section reviews instances of RCMA that work with XLS.

Different RCMA instances based on XLS have been proposed in the literature. They include the following:

- *Minimal generation gap (MGG)*. Steady-state RCGA model was originally suggested by Satoh, Yamamura and Kobayashi (1996) and later used in a number of studies (Kita, Ono and Kobayashi, 1999; Tsutsui, Yamamura and Higuchi, 1999). A generation alternation is done through applying a crossover operation λ times to a pair of parents randomly chosen from the population. From the parents and their offspring the best individual is selected and a random one using the roulette wheel technique. The original parents are replaced with them. The elite individual is selected for producing selective pressure and the random one for introducing diversity into the population. No mutation is applied under this mechanism.
- *Generalized generation gap (G3)*. Deb, Anand and Joshi (2002) modify the MGG model to make it computationally faster by replacing the roulette-wheel selection with a block selection of the best two solutions. The G3 model also preserves elite solutions from the previous iteration. In G3 the recombination and selection operators are intertwined in the following manner:
 1. From the population $P(t)$ select the best parent and $\mu - 1$ other parents randomly.
 2. Generate λ offspring from the chosen parents using a multiparent crossover operator.
 3. Choose two elements at random from the population $P(t)$.
 4. Form a combined sub-population of chosen two elements and offspring, choose the best two solutions and replace the chosen two elements with these solutions.

The justification for the design of MMG and G3 is the following. Once a standard RCGA has found fit areas of the search space, it searches over only a small fraction of the neighborhood around each search point. It must derive its power from integrating multiple single neighborhood explorations in parallel over successive generations of a population. This *many points, few neighbors strategy* is in direct contrast to a hill climber which potentially focuses effort on a greater fraction of the search neighborhood of one point but only around one point at a time. This strategy might be called *few points, many neighbors* (O'Reilly and F. Oppacher, 1995). Precisely, MGG and G3 implement this strategy by using crossover operators with multiple descendants. The idea is to induce an LS on the neighborhood of the parents involved in crossover. In this way, this type of crossover operators constitute an XLS.

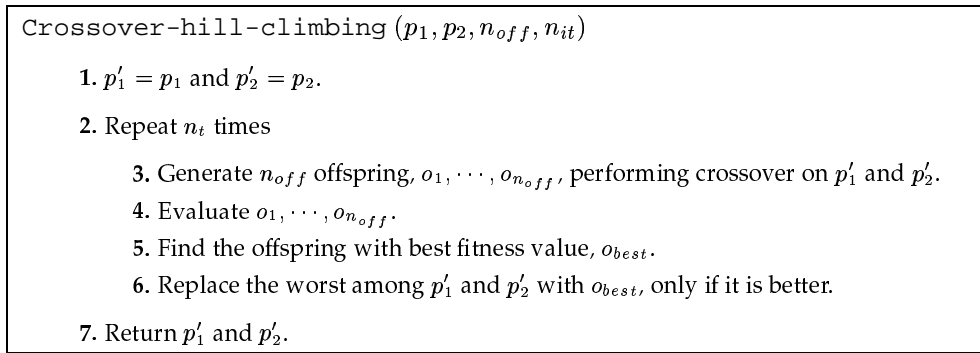


Figure 1: Pseudocode algorithm for XHC

- *Family competition* (FC). The FC model of Yang and Kao (2000) includes an XLS that explores the neighborhood of an element by applying it repeatedly crossover with different mates.

During the FC procedure, each individual I_p sequentially becomes the family father. With a probability p_c , this family father and another solution I_1 randomly chosen from the rest of the parent population are used as the parents in a crossover operation. Then the new offspring is operated by mutation to generate an offspring C_1 . For each family father, this procedure is repeated L times. Finally, L solutions (C_1, \dots, C_L) are produced but only the solution C_b with the best value of fitness function survives. Later, a replacement selection is used to select the better one from the family parent and its best individual.

The FC principle is that each individual in the population do an LS with length L and only the best offspring survives. Since L solutions are created from the same family father and perform selection, the family competition strategy is similar to $(1, \lambda)$ selection. The authors suggested that FC is a good way to avoid premature convergence but also to keep the spirit of local searches.

4 Real-Parameter Crossover Hill-Climbing

Hill-climbing is a LS algorithm that commences from a single solution point. At each step, a candidate solution is generated using a move operator of some sort. The algorithm simply moves the search from the current solution to a candidate solution if the candidate has better or equal fitness. Crossover hill-climbing (XHC) was first described by Jones (1995) and O'Reilly and F. Oppacher (1995) as a special XLS approach. Its basic idea is to use hill-climbing as the move accepting criterion of the search and use crossover as the move operator.

We present a real-parameter XHC that maintains a pair of parents and perform repeatedly crossover on this pair until some number of offspring, n_{off} , is reached. Then, the best offspring is selected and replaces the worst parent only if it is better. The process iterates n_{it} times and returns the two final current parents. This XHC model requires values for n_{off} and n_{it} , and a starting pair of parents, (p_1, p_2) . Although we use here BLX- α crossover, it must be emphasized that our model can be instantiated with any other standard real-coded crossover. Figure 1 shows the pseudocode algorithm for XHC.

An XHC instance may be obtained using the real-parameter crossover operator presented in Section 2.1, the PBX- α operator.

The XHC proposed may be conceived as a *micro selecto-recombinative* RCGA model that employs the minimal population size necessary to allow the crossover to be applicable, i.e., two chromosomes. The competition process (step 6) differentiates this mechanism from the simple application of a crossover operator with multiple descendants (Section 2). Precisely, our motivation was the definition of a XLS model that allows the self-adaptive capacity of a real-parameter crossover operator to be exploited inside the XLS itself, i.e., the design of a *self-adaptive* XLS. The competition process may modify the current pair of parents during the XHC run, changing the spread of parents. Since real-parameter crossovers generate offspring according to the distribution of parents, the convergence or divergence of XLS will be accomplished without any adaptive parameter. It has been argued that *incest* should be kept to a minimum within evolutionary algorithms to avoid premature convergence (Craighurst and Martin, 1995), (Eshelman and Schaffer, 1991), (Schaffer, Mani, Eshelman and Mathias, 1999). However, it is precisely convergence what we need to achieve with a local search (crossover based in this case) in a continuous domain. Hence, our methods can be also understood as promoting incest between the best of the current parents and the best of their offspring.

Most well-known continuous local searchers (derivative-free), such as the Solis and Wets' algorithm (Solis and Wets, 1981) and the (1+1)-evolution strategy (Rechenberg, 1973; Schwefel, 1981), make use of explicit control parameters (e.g., *step sizes*) to guide the search. In addition, they adapt the parameters, in such a way that the moves being made may be of varying sizes, depending on the success of previous steps. The rules for updating parameters capture some lawful operation of the dynamics of the algorithm over a broad range of problems. In this case, there is an explicit parameter adaptation.

The idea of employing GA models as hill-climbers is not new; Kazarlis, Papadakis, Theocharis and Petridis (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 *genetic LS*. 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.

5 Real-Coded MA with Crossover Hill-Climbing

In this section, we present our proposal of RCMA. It is a steady-state RCMA that invokes a real-parameter XHC:

- The mission of the XHC is to obtain the best possible accuracy levels to lead the population toward the most promising search areas, producing an effective refinement on them. So, its principal mission is to obtain the best possible *accuracy* levels. To accomplish that goal our approach relies on an *incest promotion* mechanism.
- The steady-state RCMA is designed to promote high population diversity levels. It attempts to induce *reliability* in the search process, ensuring that different promising search zones are focused by the XHC throughout the run. Therefore, it attempts to induce *reliability* in the search process.

1. Select two parents from the population.
2. Create an offspring using crossover and mutation.
3. Evaluate the offspring with the fitness function.
4. Select an individual in the population, which may be replaced by the offspring.
5. Decide if this individual will be replaced.

Figure 2: Pseudocode algorithm for the SSGA model

In section 5.1, we introduce the foundations of steady-state MAs. In section 5.2, we outline the different steps that constitute the steady-state RCMA. In section 5.3, we explain the resources considered to favor diversity in the population of this algorithm. Finally, in section 5.4, we present an adaptive mechanism that assigns to every chromosome a probability of being refined by XHC.

5.1 Steady-State MAs

In *steady-state* GAs (SSGAs) usually only one or two offspring are produced in each generation. Parents are selected to produce offspring and then a decision is made as to which individuals in the population to select for deletion to make room for the new offspring. SSGAs are overlapping systems, since parents and offspring compete for survival. The basic algorithm step of SSGA is shown in Figure 2.

These steps are repeated until a termination condition is achieved. In step 4, one can choose the *replacement strategy* (e.g., replacement of the worst, the oldest, or a randomly chosen individual). In step 5, one can choose the *replacement condition* (e.g., replacement if the new individual is better, or unconditional replacement). A widely used combination is to replace the worst individual only if the new individual is better. In the paper, this strategy will be called *standard replacement strategy*. In Goldberg and Deb (1991), it was suggested that the deletion of the worst individuals induced a high selective pressure, even when the parents were selected randomly.

Although SSGAs are less common than generational GAs, Land (1998) recommended their use for the design of *steady-state* MAs (SSGAs plus LS), because they may be more stable (as the best solutions do not get replaced until the newly generated solutions become superior) and allow the results of LS to be maintained in the population. The LS is applied, after Step 3, on the offspring created in Step 2. Then, Steps 4 and 5 are followed to address the inclusion of the resulting refined solution into the population. LS need not be applied to every solution being generated, because the additional function evaluations required for LS search can be very expensive. Thus, a parameter, called LS probability, p_{LS} , is introduced, which determines the probability that LS will be invoked to refine a new chromosome.

Steady-state MAs integrate global and local search more tightly than generational MAs (Land, 1998). This interleaving of the global and local search phases allows the two to influence each other, e.g., the SSGA chooses good starting points, and LS provides an accurate representation of that region of the domain. Contrarily, generational MAs proceed in alternating stages of global and local search. First, the generational GA produces a new population, then LS is performed. The specific state of LS generally is not kept from one generation to the next, though LS results do influence selection of individuals.

5.2 Steady-State RCMA Model

The main features of the steady-state RCMA proposed are (Figure 3):

- High population diversity levels are favored by means of the combination of the PBX- α crossover (Section 2) with a high value for its associated parameter ($\alpha = 1$) and the negative assortative mating strategy (Fernandes and Rosa, 2001) (Section 5.3.1). The diversity is promoted as well by means of the BGA mutation operator (Mühlenbein, D. Schlierkamp-Voosen, 1993) (Section 5.3.2).
- The real-parameter XHC presented in Section 4 is invoked to refine the new chromosomes created from the application of crossover and mutation. The XHC requires two starting chromosomes to commence its operation; the first one is the new chromosome generated. In this study, we choose the current best element in the population as second parent. However, other alternatives for the selection of this parent are possible, such as a random selection or the mating strategies offered in Fernandes and Rosa (2001), Huang (2001), and Ronald (1993). After the XHC processing, two solutions are returned, the two final parents. The fittest parent may improve the current best element in the population. In this case, it will be included in the population and the old best individual will be removed. The another parent returned by XHC will be introduced in the population following the standard replacement strategy (Section 5.1).
- A fitness-based adaptive method is considered to determine the LS probability, p_{LS} , for each new chromosome. Those chromosomes that are fitter than the current worst individual in the population receive the highest p_{LS} value ($p_{LS} = 1$). In this way, they are refined using XHC. Chromosomes that do not accomplish this requirement obtain a low p_{LS} value, $p_{LS} = 0.0625$, which is considered appropriate for many practical cases (Hart, 1994; Rosin, Halliday, Hart and Belew (1997); Hart, Rosin, Belew and Morris, 2000).
- The *local/global search ratio* ($\frac{L}{G}$ ratio) shown by the RCMA (defined as the percentage of evaluations spent doing local search from the total assigned to the algorithm's run) is governed by three parameters, n_{it} , n_{off} , and p_{LS} . The $\frac{L}{G}$ ratio determines the trade-off between the exploration abilities of the RCMA, and the exploitation abilities of the XHC, and then, it has an important influence on the final performance of the algorithm on a particular problem. The higher the values for these parameters are, the near to 100% the $\frac{L}{G}$ ratio is. For complicated problems, low $\frac{L}{G}$ ratio values become more effective, because the exploration is favored by global search, whereas for non-complex problems, higher $\frac{L}{G}$ ratio values are convenient, because XHC may exploit the search space during a long time, taking advantage of its ability to refine solutions. With the employ of the adaptive p_{LS} mechanism, we attempt to adjust (as well as possible) the $\frac{L}{G}$ ratio to adequate values that allow high performance to be achieved on the particular problem to be solved.

5.3 Resources to Favor Population Diversity

Population diversity is crucial to a GA's ability to continue the fruitful exploration of the search space. If the lack of population diversity takes place too early, a premature stagnation of the search is caused. Under these circumstances, the search is likely to be trapped in a local optimum before the global optimum is found. This problem, called *premature convergence*, has long been recognized as a serious failure mode for

```

1. Initialize population.
2. While (not termination-condition) do
    3. Use Negative-assortative-mating-strategy to select two parents.
    4. Apply PBX-crossover and BGA-mutation to create an offspring,  $o_{new}$ .
    5. Evaluate  $o_{new}$ .
    6. Invoke Adaptive- $p_{LS}$ -mechanism to obtain  $p_{LS}$  for  $o_{new}$ .
    7. If  $u(0, 1) < p_{LS}$  then
        8. Find the best chromosome in the population,  $c_{best}$ .
        9. Perform Crossover-hill-climbing ( $o_{new}, c_{best}, n_{off}, n_{it}$ ).
            $c_{xhc}^1$  and  $c_{xhc}^2$  are returned ( $c_{xhc}^1$  is the best).
        10. Replace  $c_{best}$  with  $c_{xhc}^1$ , only if it is better.
        11. Utilize Standard-replacement-strategy to insert  $c_{xhc}^2$  in population.
    12. Else
        13. Employ Standard-replacement-strategy to insert  $c_{new}$  in population.

```

Figure 3: Pseudocode algorithm for the Steady-State RCMA proposed

GAs (Eshelman and Schaffer, 1991). In the MA literature, keeping population diversity while using LS together with a GA is always an issue to be addressed, either implicitly or explicitly (Krasnogor, 2002). We review some approaches:

- Mühlenbein, Schomisch and Born (1991) integrate LS procedures to distributed GAs, which keep, in parallel, several sub-populations that are processed by genetic algorithms, with each one being independent from the others. Their advantage is the preservation of diversity due to the semi-isolation of the sub-populations.
- Merz (2000) show many different combinations of LS and GA for the traveling salesman problem while defining specific purpose crossover and mutation operators. The crossover used by the authors is the DPX crossover which was specifically designed to preserve diversity by means of keeping constant the appropriately defined hamming distance between the two parent tours and the offspring generated. In addition, a restart technique is employed. During the run, the solutions contained in the population move closer together until they are concentrated on a small fraction of the search space: the search is said to have converged. The restarts perturb the population so that the points are again far away from each other. Thus, it represents an escape mechanism from suboptimal regions of the search space.
- Nagata and Kobayashi (1997) describe a powerful MA with an intelligent crossover in which the local searcher is embedded in the genetic operator. Furthermore, populations a couple of orders of magnitude bigger than those used by other authors were employed, with the expected increase in diversity.
- Krasnogor and Smith(2000) introduces a hybridization scheme for a MA based on an adaptive helper that uses statistics from the GA's population. His MA is com-

posed of two optimization processes, a GA and a helper that is a Monte Carlo method, which serves two purposes. First, when the population is diverse, it acts like a LS procedure and second, when the population converges, its goal is to diversify the search.

- Krasnogor and Smith(2001) integrates two mechanisms to promote diversity: on one hand they employ a cohort of local searchers within the MA as each one of them “sees” a different landscapes. This, in turn, allows individuals to avoid the local optima of one operator by using a different local searcher. On the other hand they employ self-adaptation for the selection of which local searcher to use at different stages of the search.
- Seront and Bersini (2000) present an MA with a clustering method that reduces the total cost of LS by avoiding the multiple rediscoveries of local optima. In addition, the clustering method supplies information that can be used to maintain the diversity in the population. Kemenade (1996) present a MA model based on evolution strategies that capture similar ideas.
- Finally, Parthasarathy, Goldberg and Burns (2001) address the issue of handling explicitly multimodal functions using MAs. They utilize of the adaptive niching method via coevolutionary sharing of Goldberg and Wang (1997) to stably maintain a diverse population throughout the search.

As we have mentioned, the steady-state RCMA proposed employs two mechanism to promote high population diversity degrees, the negative assortative mating and the BGA mutation. Next, we explain their principal characteristics.

5.3.1 Negative Assortative Mating

The mating selection mechanism determines the way the chromosomes are mated for applying the crossover to them (Step 1 in Figure 2). Mates can be selected so as to favor population diversity (Craighurst and Martin, 1995; Eshelman, 1991; Fernandes and Rosa, 2001). A way to do this is the *negative assortative mating* mechanism. 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) assume these ideas to implement a parent selection mechanism in the crossover operator. A first parent is selected by the roulette wheel method and n_{ass} chromosomes are selected with the same method (in our experiments all the parents 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, 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, the negative assortative mating mechanism increases genetic diversity in the population by mating dissimilar genomes with higher probability.

The steady-state RCMA proposed (Figure 3) combines the negative assortative mating (that favors high population diversity levels) (Step 3) with the standard replacement strategy (that induces high selective pressure, as mentioned in Section 5.1) (Steps 11 and 13). In this way, many dissimilar solutions are produced during the run and only the best ones are conserved in the population, allowing diverse and promising

solutions to be maintained. The filtering of high diversity by means of high selective pressure has been suggested by other authors as GA strategy to provide effective search. For example, in Shimodaira (1996), an algorithm is proposed employing large mutation rates and population-elitist selection, and in Eshelman (1991), a GA is proposed, which combines a disruptive crossover operator with a conservative selection strategy.

5.3.2 BGA Mutation Operator

The mutation operator serves to create random diversity in the population (Spears, 1993). In the case of working with real coding, a topic of major importance involves the control of the proportion or strength in which real-coded genes are mutated, i.e., the *step size* (Bäck, 1996). Different techniques have been suggested for the control of the step size during the RCGA's run (Herrera and Lozano, 2000a; Smith and Fogarty, 1997). An example is the non-uniform mutation, which is considered to be one of the most suitable mutation operators for RCGAs (Herrera, Lozano, Verdegay, 1998). Its main idea is to decrease the step size as the execution advances. In this way, it makes an uniform search in the initial space and very locally at a later stage, favoring local tuning. An alternative method is the one introduced in (Krasnogor and Smith, 2001) where a *discrete set* of mutation rates is made available to the algorithm which can self-adapt to use any of them. Some advantages of using this model were discussed in the reference mentioned above and analysed further in (Smith 2001).

In our case, the responsible for the local tuning of the solutions is XHC. Hence, we really require a mutation operator providing acceptable levels of diversity continuously. One of the mutation operators that behaves in this manner is the BGA mutation operator (Mühlenbein, D. Schlierkamp-Voosen, 1993).

Let us suppose $C = (c_1, \dots, c_i, \dots, c_n)$ a chromosome and $c_i \in [a_i, b_i]$ a gene to be mutated. The gene, c'_i , resulting from the application of this operator is:

$$c'_i = c_i \pm rang_i \cdot \sum_{k=0}^{15} \alpha_k 2^{-k},$$

where $rang_i$ defines the mutation range and it is normally set to $0.1 \cdot (b_i - a_i)$. The + or - sign is chosen with a probability of 0.5 and $\alpha_k \in \{0, 1\}$ is randomly generated with $p(\alpha_k = 1) = \frac{1}{16}$. Values in the interval $[c_i - rang_i, c_i + rang_i]$ are generated using this operator, with the probability of generating a neighborhood of c_i being very high. The minimum possible proximity is produced with a precision of $rang_i \cdot 2^{-15}$.

5.4 Adaptive p_{LS} Mechanism

LS typically operates over a small portion of the total visited solutions, because the additional function evaluations required for local search can be very expensive. The question naturally arises as to how best to select the solutions which will undergo LS. In (Land 1998) the author introduced the concept of "sniffs", that is, individual solutions were subject of a limited amount of local search (i.e. a sniff), moreover, those solutions that were in the proximity of a promising basin of attraction received (at a latter stage) and extended cpu budget. With that budget further iterations of local search were performed. Hart (1994) address this issue and proposes different mechanisms for adaptively calculating the LS probability with which LS is applied to each new chromosome:

- Fitness-based adaptive methods use the fitness information in the population to

bias the LS toward individuals that have better fitness. They modify the LS probability of an individual based on the relationship of its fitness to the rest of the population. These methods assume that individuals with better fitness are more likely to be in basins of attraction of good local optima.

- Distribution-based adaptive methods use redundancy in the population to avoid performing unnecessary local searches. In particular, selected solutions will be far away from each other, and ideally span as much of the search space as the population itself. This helps ensure locally optimised solutions cover the search space, and tends to prevent premature convergence.

Since the steady-state RCGA proposed attempts to maintain a diverse population, we focus our attention on fitness-based adaptive methods. In particular, we explore a simple adaptive scheme to assign an LS probability value to each chromosome generated by crossover and mutation, c_{new} :

$$p_{LS} = \begin{cases} 1 & \text{if } f(c_{new}) \text{ is better than } f(c_{worst}) \\ 0.0625 & \text{Otherwise} \end{cases}$$

where f is the fitness function and c_{worst} is the current worst element in the population.

We consider that a nascent chromosome, c_{new} , being better than the current worst element is a promising element, and thus, it deserves a local tuning. For this reason, the adaptive approach ensures that it will undergo LS by means of the XHC application. In addition, the resultant chromosomes, supposedly more promising, will form part of the population. In this way, the steady-state RCMA maintains chromosomes that provide precise information about the quality of fitting search regions. On the other side, when the above circumstance is not accomplished, then a low value for p_{LS} is assumed for c_{new} ($p_{LS} = 0.0625$). As was observed by Hart (1994), in many cases, applying LS to as little of 5% of each population results in faster convergence to good solutions.

The reader must note that many other adaptive MA approaches are to be found, e.g., in Espinoza, Minsker and Goldberg (2001), Krasnogor (2002), and Magyar, Johnson and Nevalainen (2000).

6 Experiments

We have carried out different minimisation experiments, on the test suite described in Appendix A, in order to determine the performance of the RCMA with XHC and study its main features. We have planned these experiments as follow:

- First, in Section 6.1, we analyse the behavior of the RCMA varying the n_{it} parameter, with the aim of determining the more robust value for this parameter. All the posterior experiments are accomplished using this value.
- In Section 6.2, we examine the effects on the exploration/exploitation balance resulting from the combination between the two main ingredients of our proposal, the negative assortative mating strategy and the XHC.
- In Section 6.3, we compare the XHC model with a XLS based on crossover with multiple descendants. Now, our purpose is to determine if the intrinsic self-adaptation of XHC really works, inducing promising solution refinement.
- In Section 6.4, we investigate whether the adaptive p_{LS} mechanism (Section 5.4) tunes the $\frac{I}{G}$ ratio of the RCMA depending on the particular problem to be solved, allowing a robust operation to be achieved.

- In Section 6.5, we attempt to demonstrate the superiority of the proposed approach by means of an extensive comparison of the proposed approach with a number of challenging competitors chosen from MA literature.
- Finally, in Section 6.6, we examine the search bias associated with the PBX- α crossover operator, with the aim of checking if this operator has any bias towards the center of the search space.

6.1 Influence of the n_{it} Parameter

In our first empirical study, we investigate the influence of the n_{it} (number of iterations accomplished by XHC) on the performance of the RCMA proposed, since it has a significant effect upon the $\frac{L}{G}$ ratio. In particular, we analyse the behavior of the algorithm when different values for this parameter are considered. A fixed value for n_{off} was assumed (the number of offspring generated from the current pair of parents) ($n_{off} = 3$).

n_{it}	f_{sph}		f_{Ros}		f_{Sch}		f_{Ras}	
	A	B	A	B	A	B	A	B
1	1.5e-045	2.2e-049	1.1e+001	9.3e-002	8.5e+001	6.4e+000	8.9e-001	40.0 %
2	1.4e-080	2.0e-085	3.8e+000	5.6e-005	6.5e-004	2.2e-005	9.2e-001	46.0 %
3	6.5e-101	1.1e-105	2.2e+000	6.0e-004	3.8e-007	4.5e-009	1.4e+000	32.0 %
4	2.6e-110	1.1e-116	1.4e+000	8.6e-007	1.7e-008	1.1e-010	1.8e+000	22.0 %
5	4.5e-117	3.1e-123	2.8e+000	1.4e-005	2.8e-007	2.4e-012	1.1e+000	26.0 %
6	1.1e-118	3.2e-125	1.8e+000	2.2e-005	1.3e-009	2.9e-012	1.4e+000	20.0 %
7	5.2e-119	1.2e-124	1.7e+000	1.3e-004	1.1e-009	1.5e-012	1.9e+000	20.0 %
8	1.1e-115	4.1e-123	3.6e+000	6.3e-005	2.7e-004	1.0e-012	1.7e+000	18.0 %
9	3.6e-113	2.7e-119	2.5e+000	5.1e-005	1.2e-008	3.1e-013	2.7e+000	8.0 %
10	1.0e-106	7.2e-116	5.1e+000	2.4e-004	1.2e-010	1.2e-013	2.4e+000	12.0 %
15	2.8e-081	8.2e-091	6.2e+000	8.9e-003	6.0e-008	6.4e-012	3.0e+000	2.8e-014
20	3.2e-063	1.5e-070	1.4e+001	4.9e-002	2.0e-007	1.6e-009	6.9e+000	3.9e-011
50	4.8e-026	1.7e-030	2.0e+001	2.8e+000	6.9e-002	5.6e-004	2.2e+001	7.0e+000

n_{it}	f_{Gri}		P_{sle}		P_{Cheb}		P_{fms}	
	A	B	A	B	A	B	A	B
1	5.6e-003	58.0 %	5.2e+001	1.8e+000	1.3e+003	6.4e+001	4.1e+000	42.0 %
2	1.3e-002	28.0 %	2.7e+001	2.4e+000	2.1e+002	5.3e+000	7.1e+000	36.0 %
3	1.3e-002	30.0 %	5.5e+001	7.9e-001	1.4e+002	9.2e+000	7.7e+000	40.0 %
4	2.2e-002	18.0 %	9.0e+001	3.5e+000	1.7e+002	1.7e+000	1.1e+001	18.0 %
5	1.8e-002	18.0 %	1.0e+002	2.6e+000	2.4e+002	1.2e+001	1.2e+001	24.0 %
6	1.7e-002	22.0 %	1.3e+002	7.7e+000	2.7e+002	1.2e+001	1.4e+001	6.0 %
7	2.2e-002	24.0 %	1.2e+002	4.7e+000	2.3e+002	2.5e+000	1.3e+001	10.0 %
8	2.4e-002	30.0 %	1.4e+002	1.5e+001	2.5e+002	9.3e+000	1.5e+001	4.0 %
9	2.8e-002	16.0 %	1.2e+002	7.3e+000	2.7e+002	1.2e+001	1.6e+001	6.0 %
10	2.6e-002	20.0 %	1.2e+002	1.6e+001	2.6e+002	3.3e+000	1.4e+001	16.0 %
15	3.0e-002	14.0 %	1.4e+002	1.0e+001	2.5e+002	2.3e+000	1.4e+001	18.0 %
20	2.9e-002	24.0 %	1.3e+002	1.1e+000	2.6e+002	1.2e+000	1.5e+001	10.0 %
50	3.9e-002	10.0 %	1.4e+002	2.1e+001	4.1e+002	1.1e+000	1.7e+001	4.0 %

Table 1: Results with different values for n_{it}

We have implemented an instance of RCMA that applies an XHC based on the PBX- α operator ($\alpha = 1$) (Section 2). The mutation probability is $\frac{1}{n}$ and the population size is 60 chromosomes. The n_{ass} parameter associated with the negative assortative mating (Section 5.3.1) is set to a high value, $n_{ass} = 25$. We have considered high values for α and n_{ass} with the aim of favouring the production of individuals introducing high diversity levels in the population. The algorithm was executed 50 times, each one with a maximum of 100,000 evaluations. Table 1 shows the results obtained for different n_{it} values. The performance measures used are listed below. For each problem, the best values for these measures are printed in boldface.

- **A** performance: average of the best fitness function found at the end of each run.

- **B** performance: value of the fitness function of the best solution reached during all the runs. If the global optimum has been located throughout some runs, this performance measure will represent the percentage of runs in which this happens (in this case, a '%' sign appears along with the values for this performance measure).

A visual inspection of table 1 allows one to conclude (as expected) that the best A measure for each problems is reached with different n_{it} values:

- For the multimodal test functions, f_{Ras} and f_{Gri} , and all the real-world problems (which are very complex), low values for n_{it} allow the best results to be achieved. Low n_{it} values force the global search through the steady-state RCMA, favoring the generation of diversity. This conduct is essential to tackle with these type of problems.
- Higher values for n_{it} provide an elongated operation of XHC. For unimodal test functions, f_{sph} , f_{Ros} , and f_{Sch} , this allows an effective refinement of solutions to be accomplished.

In (Krasnogor and Smith, 2001) a self-adapting mechanism for the selection of the number of iterations of the local searchers was introduced. The results in table 1 are a strong indication that such a mechanism might provide additional benefits also to real coded MAs.

We have chosen a particular value for n_{it} , in order to allow the incoming study of our proposal and the comparison with other MA models to be easily understandable. We consider that with $n_{it} = 3$, an acceptable robustness is achieved, with regards to all the n_{it} values analysed (see Table 1). In many cases, the results offered with this value are similar to the best ones.

6.2 Synergy Between Negative Assortative Mating and XHC

Two important factors of the RCMA proposed are the promotion of diversity (exploration) by means of the negative assortative mating strategy (NAM) and the refinement of solutions carried out by XHC (exploitation). In this section, we attempt to examine whether the combination of these two ingredients decisively affects the performance of the RCMA, i.e., whether there exists synergy between them. The synergy will occur when the combination of NAM and XHC performs better than the sole usage of any of them (Yoon and Moon, 2002). We have used Table 2 in order to accomplish this investigation. It shows the results of the RCMA ($n_{it} = 3$) when either NAM, or XHC, or both of them are not applied by the RCMA.

A two-sided t-test (H_o : means of the two groups are equal, H_a : means of the two group are not equal) at 0.05 level of significance was applied in order to ascertain if differences in the A performance of the RCMA based on both NAM and XHC are significant when compared against the one for the other alternatives (RCMA without NAM, or without XHC, or without both of them). The direction of any significant differences is denoted either by:

- a plus sign (+) for an improvement in A performance, or
- a minus sign (-) for a reduction, or
- no sign for non significant differences.

RCMA	f_{sph}		f_{Ros}		f_{Sch}		f_{Ras}	
	A	B	A	B	A	B	A	B
without NAM and XHC	2.6e-023+	5.1e-025	2.2e+01+	1.9e+01	3.4e-01+	4.5e-02	1.0e+00	12.0 %
without XHC	2.0e-016+	3.1e-017	2.0e+01+	1.8e+01	5.7e+02+	2.3e+02	3.1e+00+	1.1e-7
without NAM	3.1e-114-	5.6e-120	1.9e+00	1.1e-04	3.3e-05	9.7e-14	2.1e+00	8.0 %
with NAM and XHC	6.5e-101	1.1e-105	2.2e+00	6.0e-04	3.8e-07	4.5e-09	1.4e+00	32.0 %

RCMA	f_{Gri}		P_{sle}		P_{Cheb}		P_{fms}	
	A	B	A	B	A	B	A	B
without NAM and XHC	4.0e-03-	74.0 %	7.6e+01	3.4e+00	3.9e+02+	2.5e+01	9.0e+00	8.0 %
without XHC	1.9e-02	3.1e-014	3.4e+02+	1.2e+02	2.6e+03+	4.6e+02	1.5e+01+	1.2e+01
without NAM	2.6e-02+	18.0 %	1.7e+02+	2.0e+00	5.0e+02+	4.3e+01	1.5e+01+	8.0 %
with NAM and XHC	1.3e-02	30.0 %	5.5e+01	7.9e-01	1.4e+02	9.2e+00	7.7e+00	40.0 %

Table 2: Results with different combinations of NAM and XHC

The proposal (RCMA with NAM and XHC) clearly improves the results of the three alternatives considered. Thus, we may conclude that there exists a profitable synergy between NAM and XHC, because their combination produces a balance between exploration and exploitation that becomes determinant for the success of this algorithm.

6.3 Analysis of the Self-Adaptation of XHC

In this paper, we have presented the XHC model as a self-adaptive XLS technique (Section 4). In addition, we have explained that the self-adaptation may be accomplished when we use a real-parameter crossover operator with self-adaptation abilities (Beyer and Deb, 2001) and the competition process integrated in XHC (Step 6 in Figure 1).

In this section, we attempt to determine whether the self-adaptation of XHC really may allow an effective refinement of the solutions to be achieved. In particular, we compare an RCMA based on XHC (denoted as RCMA-XHC) with a similar algorithm that uses an XLS that simply generates $n_{off} \times n_{it}$ offspring from the starting pair of parents, and returns the two best individuals among parents and offspring (it will be called RCMA-XLS). Since RCMA-XHC works with $n_{off} = 3$ and $n_{it} = 3$, RCMA-XLS will generate 9 chromosomes for each XLS application. In order to make an adequate comparison, we have disabled the adaptive p_{LS} mechanism and consider a fixed value for p_{LS} ($p_{LS} = 0.0625$). Table 3 contains the results. A t-test was performed to determine if there exist differences in the A performance of these two algorithms.

Algorithm	f_{sph}		f_{Ros}		f_{Sch}		f_{Ras}	
	A	B	A	B	A	B	A	B
RCMA-XLS	8.9e-020+	2.6e-021	1.5e+001+	7.9e+000	2.4e+001+	4.4e-001	1.6e+000	3.0e-011
RCMA-XHC	6.0e-040	3.0e-042	7.7e+000	2.0e-002	1.3e-003	1.3e-005	1.1e+000	28.0 %

Algorithm	f_{Gri}		P_{sle}		P_{Cheb}		P_{fms}	
	A	B	A	B	A	B	A	B
RCMA-XLS	1.3e-002	22.0 %	1.3e+001-	1.3e+000	9.5e+002+	5.6e+001	3.4e+000-	60.0 %
RCMA-XHC	1.5e-002	34.0 %	3.3e+001	3.5e+000	1.8e+002	4.2e+000	6.0e+000	46.0 %

Table 3: Comparison between RCMA-XHC and RCMA-XLS

RCMA-XHC provides a better A performance than RCMA-XLS on the unimodal functions, f_{sph} , f_{Ros} , and f_{Sch} . For the multimodal f_{Gri} and f_{Ras} the t-test indicates non differences between the algorithms. However, RCMA-XHC achieves a better B performance than RCMA-XLS on these functions. For P_{sle} and P_{fms} (which are very complex), RCMA-XHC is outperformed by RCMA-XLS.

In XHC, the use of the competition process introduces a high selective pressure

and limits the area where the XHC acts, which is then exploited by the crossover. In general, this induces a profitable behavior on many problems. Nevertheless, for the complex ones, the selective pressure may have negative effects, because it eliminates the capacity of XHC to produce appropriate jumps to locate more promising areas in the search space regions being refined.

6.4 Study of the Adaptive p_{LS} Mechanism

There are, at least, two ways to study the operation of an adaptive mechanism for GAs (Spears, 1995). The first is from the point of view of performance (test functions are commonly used to evaluate performance improvement). The second view is quite different in that it ignores performance and concentrates more on the adaptive mechanism itself, i.e., its ability to adjust the GA configuration according to the particularities of the problem to be solved. Once given these two points of view, it is natural to investigate the way in which adaptive behavior is responsible for the performance improvement.

In this section, we tackle the study of the adaptive p_{LS} mechanism from the point of view of the adaptation itself. In particular, we are interested in determining whether it adjusts the $\frac{L}{G}$ ratio of the RCMA proposed according to the particularities of the problem to be solved, allowing performance improvement to be achieved. Results are shown in Figure 4. For every test problem, it outlines the average of the $\frac{L}{G}$ ratio found throughout the 50 runs of the RCMA proposed ($n_{it} = 3$).

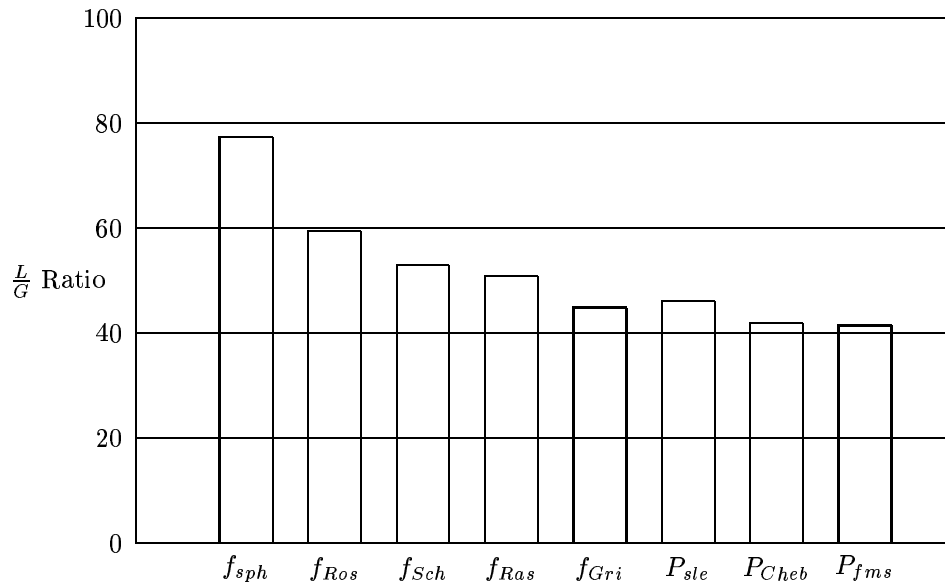


Figure 4: $\frac{L}{G}$ ratio produced by the RCMA

There are differences between the $\frac{L}{G}$ ratio values for the different problems. The highest values are more fruitful for the unimodal f_{sph} , f_{Ros} , and f_{Sch} , whereas lower values becomes more effective for the most complex problems, P_{Cheb} and P_{fms} . In fact, the ratio value for f_{sph} duplicates the one for P_{fms} . These differences in the $\frac{L}{G}$ ratio arise as a sign of the adaptation ability (from the point of view of the adaptation itself) of the adaptive p_{LS} mechanism. They confirm that this method induces $\frac{L}{G}$ ratios adjusted to the particular problem to be solved.

Although this adaptive mechanism shows signs of adaptation, we have to check whether this adaptation is indeed beneficial. In order to do so, we have executed three RCMA instances with the same features as the proposal, but they use a fixed p_{LS} value (0.0625, 0.25, and 1, respectively). Table 4 shows their results, which may be compared with the ones for the proposal (denoted as *Adaptive* in the table) by means of the t-test.

p_{LS}	f_{sph}		f_{Ros}		f_{Sch}		f_{Ras}	
	A	B	A	B	A	B	A	B
0.0625	6.0e-040+	3.0e-042	7.7e+000+	2.0e-002	1.3e-003	1.3e-005	1.1e+000	28.0 %
0.25	6.8e-057+	6.3e-061	3.6e+000+	1.4e-004	1.1e-006	3.7e-009	1.3e+000	40.0 %
1	7.4e-065+	1.1e-068	2.4e+000	9.5e-004	6.6e-008-	1.0e-010	1.4e+000	22.0 %
<i>Adaptive</i>	6.5e-101	1.1e-105	2.2e+000	6.0e-004	3.8e-007	4.5e-009	1.4e+000	32.0 %

p_{LS}	f_{Gri}		P_{sle}		P_{Cheb}		P_{fms}	
	A	B	A	B	A	B	A	B
0.0625	1.5e-002	34.0 %	3.3e+001-	3.5e+000	1.8e+002	4.2e+000	6.0e+000	46.0 %
0.25	1.5e-002	28.0 %	8.1e+001+	2.4e+000	1.9e+002+	1.1e+001	1.0e+001	22.0 %
1	1.7e-002	16.0 %	1.1e+002+	3.2e+000	3.5e+002+	4.8e+000	1.2e+001+	20.0 %
<i>Adaptive</i>	1.3e-002	30.0 %	5.5e+001	7.9e-001	1.4e+002	9.2e+000	7.7e+000	40.0 %

Table 4: Comparison with different fixed values for p_{LS}

To favor a high $\frac{L}{G}$ ratio by using $p_{LS} = 1$ allows suitable results to be obtained for the unimodal f_{sph} , f_{Ros} , and f_{Sch} , whereas the production of a low $\frac{L}{G}$ ratio considering $p_{LS} = 0.0625$ achieves the best performance for the multimodal f_{Gri} and for all the complex real-world problems. This means that global search is well-suited for complex problems and local search is useful for unimodal test functions, which is very reasonable.

Precisely, Figure 4 indicates that, for each problem, the proposal might induce $\frac{L}{G}$ ratio with this tendency. This may explain that, in general, it might return results that are similar to the ones for the most successful instance with fixed p_{LS} values (the case of f_{Ros} , f_{Ras} , f_{Gri} , P_{Cheb} , and P_{fms}), or even better than all them (the case of f_{sph}).

To sum up, this study shows that the adaptation ability of the adaptive p_{LS} mechanism allows the $\frac{L}{G}$ ratios to be adjusted according to the particularities of the search space, allowing significant performance to be achieved for problems with different difficulties.

6.5 Comparison with Other RCMA

In this subsection, we compare the performance of the RCMA with XHC ($n_{off} = 3$ and $n_{it} = 3$) with the one of other RCMA proposed in the literature. They include: hybrid steady-state RCMA, the G3 model (Section 3), the family competition algorithm (Section 3) and an RCMA based on the CHC algorithm (Eshelman, 1991).

- *Hybrid steady-state RCMA*. It is a simple steady-state RCMA (Figure 3), where parents are selected at random and the standard replacement is considered. Every new chromosome generated by PBX- α ($\alpha = 1$) and BGA mutation undergoes the Solis and Wets's LS procedure (Solis and Wets, 1981) with $p_{LS} = 0.0625$. Three instances were run with different number of iterations for the LS (100, 1000, and 5000). They are called SW-100, SW-1000, and SW-5000, respectively.
- *G3 Model*. We have implemented different G3 instances that consider $\mu = 2$ and use PBX- α ($\alpha = 1$). They are distinguished by the value for λ (1,10, 15, 20, and 50). We will denote these algorithms as G3- λ .

Algorithm	f_{sph}		f_{Ros}		f_{Sch}		f_{Ras}	
	A	B	A	B	A	B	A	B
CHC	5.8e-031 +	3.1e-031	1.9e+001+	1.7e+001	2.0e-002+	1.5e-003	1.6e+001+	7.0e+000
CHC-SW-100	2.1e-014 +	6.7e-015	1.8e+001+	1.6e+001	2.4e+002+	9.4e+001	4.5e+001+	2.9e+001
CHC-SW-1000	9.6e-025 +	4.6e-026	1.5e+001+	7.3e+000	1.4e+001+	2.5e+000	6.2e+001+	4.0e+001
CHC-SW-5000	8.5e-063 +	3.1e-065	1.5e+001+	7.4e+000	1.2e-001+	1.6e-002	9.4e+001+	5.0e+001
G3-1	9.0e-017	7.9e-103	2.8e+001+	4.2e+000	8.3e+002+	5.3e+001	7.4e+001+	3.2e+001
G3-2	1.0e-099 +	7.6e-111	1.7e+001+	6.6e-002	1.3e+002+	6.8e-003	6.9e+001+	3.0e+001
G3-3	6.5e-095 +	1.5e-104	1.1e+001+	4.0e-005	9.3e+001+	4.3e-005	6.9e+001+	3.5e+001
G3-4	2.2e-089 +	5.6e-098	8.2e+000+	3.4e-005	2.9e+001+	1.8e-005	6.5e+001+	2.0e+001
G3-5	4.8e-083 +	1.0e-089	1.1e+001+	3.0e-004	5.0e+000+	6.9e-007	6.3e+001+	4.0e+001
G3-6	4.1e-078 +	1.9e-084	8.4e+000+	1.6e-003	9.2e+000	1.6e-007	6.5e+001+	3.1e+001
G3-7	2.1e-073	2.7e-078	9.0e+000+	1.1e-005	4.0e+000+	2.4e-006	6.0e+001+	2.5e+001
G3-8	4.0e-068	1.1e-072	1.3e+001+	1.4e-004	3.1e+000	7.2e-007	6.0e+001+	3.3e+001
G3-9	2.7e-064	2.3e-069	8.5e+000+	5.9e-003	2.2e+000	1.0e-006	5.0e+001+	1.4e+001
G3-10	9.9e-062 +	1.3e-065	8.4e+000+	9.1e-005	3.8e-001	3.4e-006	6.0e+001+	3.3e+001
G3-15	3.3e-049 +	6.4e-053	1.2e+001+	2.4e-003	8.5e-001	9.0e-007	5.1e+001+	1.9e+001
G3-20	2.8e-041 +	7.1e-044	1.7e+001+	5.0e-003	4.6e-002	1.3e-005	5.2e+001+	2.5e+001
G3-50	1.2e-021 +	4.6e-024	2.1e+001+	1.7e-002	3.5e-001+	4.0e-003	4.4e+001+	2.0e+001
SW-100	3.8e-020 +	5.6e-021	1.0e+001+	1.2e-001	2.9e-007	1.1e-017	7.6e+000+	40.0 %
SW-1000	6.9e-078	1.6e-175	4.5e+000+	1.7e-012	5.0e-008-	6.0e-029	6.8e+001+	2.3e+001
SW-5000	2.9e-120 -	1.4e-322	4.3e+000+	2.4e-003	4.1e-009-	1.4e-026	1.1e+002+	5.9e+001
FC	1.5e-006+	3.7e-007	2.3e+001+	2.1e+001	1.1e+002+	4.5e+001	5.5e+000+	2.2e+000
RCMA-XHC	6.5e-101	1.1e-105	2.2e+000	6.0e-004	3.8e-007	4.5e-009	1.4e+000	32.0 %
Algorithm	f_{Gri}		P_{sle}		P_{Cheb}		P_{fms}	
	A	B	A	B	A	B	A	B
CHC	6.5e-003 -	42.0 %	3.9e+001	7.7e-001	3.3e+002+	3.2e+000	1.7e-018-	9.1e-021
CHC-SW-100	3.4e-003 -	6.0 %	1.4e+001-	3.7e+000	1.5e+002	5.3e+000	5.0e+000-	2.1e-015
CHC-SW-1000	2.0e-002 +	4.4e-016	1.5e+002+	3.6e+001	6.6e+002+	3.4e+001	1.6e+001+	5.2e-003
CHC-SW-5000	4.4e-002 +	7.4e-003	3.6e+002+	1.7e+002	1.4e+003+	4.4e+002	2.0e+001+	1.2e+001
G3-1	5.1e-001 +	7.8e-014	3.8e+002+	3.2e+001	1.9e+003+	4.7e+001	2.1e+001+	1.1e+001
G3-2	2.7e-001	2.0 %	2.2e+002+	3.1e+001	7.8e+002+	4.3e+001	1.8e+001+	4.0 %
G3-3	2.3e-001	1.1e-016	2.1e+002+	3.2e+001	7.1e+002+	5.6e+001	1.7e+001+	4.0 %
G3-4	4.5e-002 +	8.0 %	1.5e+002+	1.1e+001	8.3e+002+	1.0e+001	1.9e+001+	5.0e-028
G3-5	3.1e-002 +	8.0 %	1.9e+002+	7.4e+000	8.2e+002+	1.4e+001	1.7e+001+	6.0 %
G3-6	3.4e-002 +	4.0 %	1.4e+002+	1.3e+001	4.9e+002+	4.3e+000	1.7e+001+	4.0 %
G3-7	3.6e-002 +	16.0 %	1.6e+002+	4.6e+000	5.6e+002+	6.6e+000	1.6e+001+	8.0 %
G3-8	2.6e-002 +	16.0 %	1.3e+002+	8.5e+000	4.3e+002+	4.9e+000	1.6e+001+	10.0 %
G3-9	2.9e-002 +	4.0 %	1.4e+002+	6.0e+000	5.4e+002+	3.2e+001	1.7e+001+	2.0 %
G3-10	2.5e-002 +	8.0 %	1.4e+002+	7.2e+000	6.3e+002+	1.9e+001	1.6e+001+	8.0 %
G3-15	1.7e-002	20.0 %	1.2e+002+	9.5e+000	4.2e+002+	2.6e+001	1.5e+001+	8.0 %
G3-20	2.1e-002 +	12.0 %	1.0e+002+	3.4e+000	3.3e+002+	1.4e+001	1.5e+001+	10.0 %
G3-50	2.0e-002	26.0 %	6.9e+001	3.0e+000	2.9e+002+	3.3e+000	1.2e+001+	18.0 %
SW-100	2.7e-002	14.0 %	9.1e+000-	6.9e-001	1.0e+002	3.0e+000	1.2e+001+	9.1e+000
SW-1000	4.9e-004 -	5.6e-016	1.1e+002+	4.5e+000	3.6e+002+	8.4e+000	1.5e+001+	8.4e+000
SW-5000	2.9e-003 -	2.6e-015	3.3e+002+	5.4e+001	1.1e+003+	2.2e+002	2.1e+001+	1.4e+001
FC	3.5e-004 -	2.2e-005	2.6e+001-	7.0e+000	3.9e+002+	3.9e+001	1.1e+001+	9.7e-003
RCMA-XHC	1.3e-002	30.0 %	5.5e+001	7.9e-001	1.4e+002	9.2e+000	7.7e+000	40.0 %

Table 5: Comparison with other RCMA models

- *Family competition.* An FC instance has been built considering the original genetic operators and values for the control parameters provided in Yang and Kao (2000).
- *Hybrid CHC Algorithm.* CHC has arisen as reference point in the GA literature (Herrera and Lozano, 2000b; Whitley, Rana, Dzuber and Mathias, 1996). Here, it is considered as alternative to steady-state RCGAs, because it is based on a $(\lambda + \lambda)$ selection strategy. Furthermore, it becomes very adequate for the design of RCMA since incorporates different techniques to promote high population diversity.

We have used a real-coded implementation of CHC (a full description is found in Herrera and Lozano, 2000b) that applies PBX- α ($\alpha = 1$). In addition, we have combined CHC with the Solis and Wets' LS procedure, producing an RCMA called CHC-SW. Each time CHC generates a chromosome by crossover, the LS is applied with $p_{LS} = 0.0625$. Three instances were run varying the number of iterations

assigned to the LS procedure. They will be called, CHC-SW-100, CHC-SW-1000, and CHC-SW-5000.

Table 5 shows the results. We have included the results for the CHC algorithm (without LS). The proposal will be referred as RCMA-XHC. A t-test was performed to ascertain if differences in the A performance for RCMA-XHC are significant when compared against the ones for the other algorithms. The direction of any differences will be denoted as in Section 6.2.

We may remark that, in general, RCMA-XHC outperforms all the other algorithms. Only CHC-SW-100 and SW-5000 significantly improve the results of RCMA-XHC on three problems and CHC and SW-1000 on two problems. Therefore, we may summarize that the proposal is very competitive with state-of-the-art RCMA.

6.6 Analysis of the Sampling Bias of PBX- α

Real-parameter crossover operators use the variation of the population to constrain the search and bias the distribution of offspring (i.e., *sampling bias*). Some real-parameter crossover operators, such as BLX- α (Eshelman and Schaffer, 1993), have a sampling bias that favour the generation of offspring being gathered towards the center of the region where the parental population covers. In other words, they tend to search the interpolative region intensively.

Fogel and Beyer (1995) and Deb, Anand and Joshi (2002) have shown that the typical initialisation used to compare evolutionary algorithms can give false impressions of relative performance when applying crossover operators with this type of biases. If the global optimum is located in the centre of the search region covered by initialising the population uniformly at random, these crossover operators generate offspring which are essentially unbiased estimates of the global optimum. In particular, they may recombine parents from opposite sides of the origin, placing the offspring close to the center of the initialisation region. In other words, the uniform initialisation technique is assumed to introduce a bias that favours a successful identification of the global optimum, such that the success of a strategy employing these crossover operators might be just an artefact of a useful combination of initialisation and global optimum location.

In this section, we analyze the PBX- α crossover operator (Section 2) from the point of view of sampling bias. In particular, we attempt to check if this operator has any bias towards the center of the search space. This is accomplished by following Angeline (1998), Eiben and Bäck (1997), and Gehlhaar and Fogel (1996), which recommend an initialization in regions that expressly do not include the optima during testing to verify results obtained for symmetric initialisation schemes. Thus, in order to study the bias of PBX- α , we have carried out two additional experiments on the test functions in which the optimum lies at $x_i = 0 \ i = 1 \dots n$, i.e., f_{sph} , f_{sch} , f_{ras} , and f_{Gri} . First, using the typical symmetric about the origin initialization, and second, starting from a *skewed initialisation*, where the initial population is located in a subset of the search space far apart from the global optimum. We assumed the initialisation intervals shown in Table 6.

If the performance of an evolutionary algorithm based on PBX- α varied extremely little under these two initialization methods, then we may believe that this operator does not show an inherent affinity towards the center of the search space. We have run three instances of our RCMA (Figure 3) that use fixed values for p_{LS} (0.0625, 0.25, and 1, respectively). The adaptive p_{LS} mechanism was disabled with the aim of ensuring that the comparison between the two initialisation methods is made under an equitable number of calls for the XHC procedure. The results are included in Table 7.

Test Function	Range
f_{sph}	[4, 5]
f_{Sch}	[60, 65]
f_{Ras}	[4, 5]
f_{Gri}	[580, 600]

Table 6: Ranges for each test function

Initialization method	p_{LS} value	f_{sph}		f_{Ros}		f_{Ras}		f_{Gri}	
		A	B	A	B	A	B	A	B
symmetric	0.0625	6.0e-40	3.0e-42	1.3e-03	1.3e-05	1.1e+00 -	28 %	1.5e-02	34 %
skewed	0.0625	1.7e-41	1.7e-41	4.1e-04	4.1e-04	4.3e+00	30 %	1.7e-02	28 %
symmetric	0.25	6.8e-57	6.3e-61	1.1e-06	3.7e-09	1.3e+00 -	40 %	1.5e-02	28 %
skewed	0.25	1.4e-56	1.3e-59	7.7e-06	1.0e-08	6.0e+00	24 %	2.2e-02	18 %
symmetric	0.5	7.4e-65	1.1e-68	6.6e-08	1.0e-10	1.4e+00 -	22 %	1.7e-02	16 %
skewed	0.5	3.8e-64	1.1e-68	3.8e-03	2.5e-10	1.3e+01	10 %	2.4e-02	26 %

Table 7: Results for the study of the search bias of PBX- α

We have analysed these results by means of a t-test and we have observed that no significant impact on the final solution accuracy is observed for all but one objective function, namely f_{Ras} , where the symmetric initialisation allow the best results to be reached for all the p_{LS} values considered. Thus, in general, these additional experiments clarify that PBX- α does not cause a search bias towards the origin of the coordinate system in case of domains of variables which are symmetric around zero.

In case of the f_{Ras} function, the skewed initialisation adds new challenges to the RCMA instances because they must overcome a number of local minima to reach the global basin. This feature causes a certain percentage of runs to stagnate in local optima, deteriorating the results. This does not occurs with the symmetric initialisation, since this method helps the algorithms to locate the global basin more easily.

7 Conclusions

This paper presented an RCMA model that applies a XHC to the solutions being generated by the genetic operators. The XHC attempts to obtain the best possible accuracy levels by iteratively promoting incest, whereas the RCMA incorporates mechanisms aimed to induce reliability in the search process. In addition, an adaptive mechanism was employed that determines the probability with which solutions are refined with XHC. The principal conclusions derived from the results of experiments carried out are the following:

- The proposal improves the performance of other RCMA approaches appeared in the MA literature (on the test suite considered in this paper).
- This success is partly possible thanks to the combination of the exploration properties of the negative assortative mating strategy and the refinement of solutions carried out by XHC.
- The adaptive p_{LS} mechanism tunes the $\frac{L}{G}$ ratio to produce a robust operation for test functions with different characteristics.
- The self-adaptive behavior of XHC works adequately on many cases, nevertheless some difficulties appeared on some of the more complex problems.

In essence, RCMA based on crossover-based local searchers are very promising and indeed worth of further studies. We are currently extending our investigation to different test-suites and real-world problems. Also we intend to:

- Test alternative XHC designs, which may be based on different replacement strategies (Step 6 in Figure 1) or other real-parameter crossover operators (e.g the multi-parent crossovers (Section 2).
- Incorporate the adaptive p_{LS} mechanism in other types of RCMA and benchmark their performance,
- Build RCMA that apply different types of LS procedures along with XHC.
- Adapt the n_{off} and n_{it} parameters during the run employing a mechanism similar to the one described in (Krasnogor and Smith, 2001).
- Investigate the so called “crossover-aware local search and “mutation-aware local search” (Krasnogor, 2002).
- Study the sensitivity of the BGA mutation operator and α and n_{ass} parameters used by our RCMA.

Appendix A. Test Suite

The test suite that we have used for the experiments consists of six test functions and three real-world problems. They are described in Subsections A.1 and A.2, respectively.

Test Functions

We have considered six frequently used test functions: *Sphere* model (f_{Sph}) (De Jong, 1975; Schwefel, 1981), *Generalized Rosenbrock's* function (f_{Ros}) (De Jong, 1975), *Schwefel's Problem 1.2* (f_{Sch}) (Schwefel, 1981), *Generalized Rastrigin's* function (f_{Ras}) (Bäck, 1992; Törn and Antanas, 1989), *Griewangk's* function (f_{Gri}) (Griewangk, 1981). Figure 5 shows their formulation. The dimension of the search space is 25.

$\begin{aligned} & \mathbf{f}_{Sph} \\ f_{Sph}(\vec{x}) &= \sum_{i=1}^n x_i^2 \\ f_{Sph}^* &= f_{Sph}(0, \dots, 0) = 0 \end{aligned}$	$\begin{aligned} & \mathbf{f}_{Ros} \\ f_{Ros}(\vec{x}) &= \sum_{i=1}^{n-1} (100 \cdot (x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \\ f_{Ros}^* &= f_{Ros}(1, \dots, 1) = 0 \end{aligned}$
$\begin{aligned} & \mathbf{f}_{Sch} \\ f_{Sch}(\vec{x}) &= \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2 \\ f_{Sch}^* &= f_{Sch}(0, \dots, 0) = 0 \end{aligned}$	$\begin{aligned} & \mathbf{f}_{Ras} \\ f_{Ras}(\vec{x}) &= a \cdot n + \sum_{i=1}^n x_i^2 - a \cdot \cos(\omega \cdot x_i) \\ & a = 10, \omega = 2\pi \\ f_{Ras}^* &= f_{Ras}(0, \dots, 0) = 0 \end{aligned}$
$\begin{aligned} & \mathbf{f}_{Gri} \\ f_{Gri}(\vec{x}) &= \frac{1}{d} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \\ & d = 4000 \\ f_{Gri}^* &= f_{Gri}(0, \dots, 0) = 0 \end{aligned}$	

Figure 5: Test functions

- f_{sph} is a continuous, strictly convex, and unimodal function.

- f_{Ros} is a continuous and unimodal function, with the optimum located in a steep parabolic valley with a flat bottom. This feature will probably cause slow progress in many algorithms since they must continually change their search direction to reach the optimum. This function has been considered by some authors to be a real challenge for any continuous function optimization program (Schlierkamp-Voosen and Mühlenbein, 1994). A great part of its difficulty lies in the fact that there are nonlinear interactions between the variables, i.e., it is *nonseparable* (Whitley, Rana, Dzuberka and Mathias).
- f_{Sch} is a continuous and unimodal function. Its difficulty concerns the fact that searching along the coordinate axes only gives a poor rate of convergence, since the gradient of f_{Sch} is not oriented along the axes. It presents similar difficulties to f_{Ros} , but its valley is much narrower.
- f_{Ras} is a scalable, continuous, and multimodal function, which is made from f_{sph} by modulating it with $a \cdot \cos(\omega \cdot x_i)$.
- f_{Gri} is a continuous and multimodal function. This function is difficult to optimize because it is non-separable and the search algorithm has to climb a hill to reach the next valley. Nevertheless, one undesirable property exhibited is that it becomes easier as the dimensionality is increased (Whitley, Rana, Dzuberka and Mathias).

A GA does not need too much diversity to reach the global optimum of f_{sph} since there is only one optimum which could be easily accessed. On the other hand, for multimodal functions (f_{Ras} and f_{Gri}), the diversity is fundamental for finding a way to lead towards the global optimum. Also, in the case of f_{Ros} and f_{Sch} , diversity can help to find solutions close to the parabolic valley, and so avoid slow progress. v

Real-World Problems

We have chosen the following three real-world problems, which, in order to be solved, are translated to optimization problems of parameters with variables on continuous domains: *Systems of Linear Equations* (Eshelman, Mathias and Schaeffer, 1997), *Frequency Modulation Sounds Parameter Identification Problem* (Tsutsui and Fujimoto, 1993), and *Polynomial Fitting Problem* (Storn and Price, 1995). They are described below.

Systems of Linear Equations The problem may be stated as solving for 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:

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

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

We have considered a 10-parameter problem instance. Its matrices are the following:

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

Frequency Modulation Sounds Parameter Identification Problem 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 $\theta = \frac{2 \cdot \pi}{100}$. The fitness function is defined as the summation of square errors between the evolved data and the model data as follows:

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

where the model data are given by the following equation:

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

Each parameter is in the range -6.4 to 6.35. This problem is a highly complex multimodal one having strong epistasis, with minimum value $P_{fms}(x^*) = 0$.

Polynomial Fitting Problem This problem lies in finding the coefficients of the following polynomial in z :

$$P(z) = \sum_{j=0}^{2k} c_j \times z^j, \quad k > 0 \text{ is integer,}$$

such that

$$P(z) \in [-1, 1], \text{ for } z \in [-1, 1], \text{ and}$$

$$P(1.2) \geq T_{2k}(1.2) \text{ and } P(-1.2) \geq T_{2k}(-1.2),$$

where $T_{2k}(z)$ is a Chebychev polynomial of degree $2k$.

The solution to the polynomial fitting problem consists of the coefficients of $T_{2k}(z)$. This polynomial oscillates between -1 and 1 when its argument z is between -1 and 1 . Outside this region the polynomial rises steeply in direction of high positive ordinate values. This problem has its roots in electronic filter design and challenges an optimization procedure by forcing it to find parameter values with grossly different magnitudes, something very common in technical systems. The Chebychev polynomial employed here is:

$$T_8(z) = 1 - 32 \cdot z^2 + 160 \cdot z^4 - 256 \cdot z^6 + 128 \cdot z^8.$$

So, it is a nine-parameter problem. The pseudocode algorithm shown below was used in order to transform the constraints of this problem into an objective function to be minimized, called P_{Chev} . We consider that $C = (c_0, \dots, c_8)$ is the solution to be evaluated and $P_C(z) = \sum_{j=0}^8 c_j \times z^j$.

```

Choose  $p_0, p_2, \dots, p_{100}$  from  $[-1, 1]$ ;
 $R = 0$ ;
For  $i = 0, \dots, 100$  do
  If  $(-1 > P_C(p_i)$  or  $P_C(p_i) > 1)$  then  $R \leftarrow R + (1 - P_C(p_i))^2$ ;
  If  $(P_C(1.2) - T_8(1.2) < 0)$  then  $R \leftarrow R + (P_C(1.2) - T_8(1.2))^2$ ;
  If  $(P_C(-1.2) - T_8(-1.2) < 0)$  then  $R \leftarrow R + (P_C(-1.2) - T_8(-1.2))^2$ ;
Return  $R$ ;

```

Each parameter (coefficient) is in the range -512 to 512. The objective function value of the optimum is $P_{Chev}(C^*) = 0$.

References

- Angeline, P.J. (1998). Using selection to improve particle swarm optimization. In *Proc. of the International Congress on Evolutionary Computation 1998*, pages 84–89.
- Bäck, T. (1992). Self-adaptation in genetic algorithms. In Varela, F. J. and Bourgine, P., editors, *Proc. of the First European Conference on Artificial Life*, pages 263–271, MIT Press, Cambridge, MA.
- Bäck, T., Schütz, M. and Khuri, S. (1996). Evolution strategies: an alternative evolution computation method. In Alliot, J. M., Lutton, E., Ronald, E., Schoenhauer, M., Snyers, D., Editors, *Artificial Evolution*, pages 3–20, Springer, Berlin.
- Beyer, H.-G. and Deb, K. (2001). On self-adaptive features in real-parameter evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 5(3): 250–270.
- Craighurst, R. and Martin, W. (1995). Enhancing GA performance through crossover prohibitions based on ancestry. In Eshelman, L. J., editor, *Proc. 6th Int. Conf. Genetic Algorithms*, pages 130–137, Morgan Kaufmann, San Mateo, California.
- 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. and Beyer, H. (2001). Self-adaptive genetic algorithms with simulated binary crossover. *Evolutionary Computation Journal*, 9(2): 195–219.
- Deb, K., Anand, A. and 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. and 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.
- Dietzfelbinger, M., Naudts, B., Van Hoyweghen, C. and Wegener, I. (2003). The Analysis of a Recombinative Hill-Climber on H-IFF. *IEEE Transactions on Evolutionary Computation*, 7(5): 417–423.
- Eiben, A. E. and Bäck, T. (1997). Empirical investigation of multiparent recombination operators in evolution strategies. *Evolutionary Computation*, 5(3): 347–365.

- Eshelman, L. J. (1991). The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination. In Rawlin, G. J. E., editor, *Foundations of Genetic Algorithms 1*, pages 265–283, Morgan Kaufmann, San Mateo, California.
- Eshelman, L. J. and Schaffer, J. D. (1991). Preventing premature convergence in genetic algorithms by preventing incest. In Belew, R. and Booker, L.B., editors, *Proc. of the Fourth Int. Conf. on Genetic Algorithms*, pages 115–122, Morgan Kaufmann, San Mateo, California.
- Eshelman, L. J. and Schaffer, J. D. (1993). Real-coded genetic algorithms and interval-schemata. In Whitley, L. D., editor, *Foundations of Genetic Algorithms 2*, pages 187–202, Morgan Kaufmann, San Mateo, California.
- Eshelman, L. J. Mathias, K. E. and Schaffer, J. D. (1997). Convergence controlled variation. In Belew, R. and Vose, M., editors, *Foundations of Genetic Algorithms 4*, pages 203–224, Morgan Kaufmann, San Mateo, California.
- Espinoza, F. P., Minsker, B. S. and Goldberg, D. E. (2001). A self adaptive hybrid genetic algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference 2001*, pages 759, Morgan Kaufmann, San Mateo, California.
- Fernandes, C. and 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*, pages 60–66, IEEE Press, Piscataway, New Jersey.
- Fogel, D. B. and Beyer, H.-G. (1995). A note on the empirical evaluation of intermediate recombination. *Evolutionary Computation*, 3(4): 491–495.
- Gehlhaar, D. and Fogel, D. (1996). Tuning evolutionary programming for conformationally flexible molecular docking. In Fogel, L., Angeline, P. and Bäck, T., editors, *Proc. Of the Fifth Annual Conference on Evolutionary Programming*, pages 419–429, MIT Press, Cambridge, MA.
- Goldberg, D. E. and Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In Rawlins, G. J. E., editor, *Foundations of Genetic Algorithms*, pages 69–93, Morgan Kaufmann, San Mateo, California.
- Goldberg, D. E. and Wang, L. (1997). Adaptive niching via coevolutionary sharing. In Quagliarella et al., editors, *Genetic Algorithms in Engineering and Computer Science*, pages 21–38, John Wiley and Sons, New York.
- Goldberg, D. E. and Voessner, S. (1999). Optimizing global-local search hybrids. In Banzhaf, W., et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference'99*, pages 220–228, Morgan Kaufmann, San Mateo, California.
- Griewangk, A. O. (1981). Generalized descent of global optimization. *Journal of Optimization Theory and Applications*, 34: 11–39.
- Hart, W. E. (1994). *Adaptive Global Optimization with Local Search*. Ph.D. Thesis, University of California, San Diego, California.
- Hart, W. E., Rosin, C. R., Belew, R. K. and Morris, G. M. (2000). Improved evolutionary hybrids for flexible ligand docking in autodock. In *Proc Intl Conf on Optimization in Computational Chem and Molec Bio.*, pages 209–230.
- Hart, W. E. (2001a). A convergence analysis of unconstrained and bound constrained evolutionary pattern search. *Evolutionary Computation*, 9(1): 1–23.
- Hart, W. E. (2001b). Evolutionary pattern search algorithms for unconstrained and linearly constrained optimization. *IEEE trans. Evolutionary Computation*, 5(4): 388–397.
- Herrera, F., Lozano, M. and Verdegay, J. L. (1996). Dynamic and Heuristic Fuzzy Connectives-Based Crossover Operators for Controlling the Diversity and Convergence of Real Coded Genetic Algorithms. *Int. Journal of Intelligent Systems*, 11: 1013–1041.

- Herrera, F., Lozano, M. and 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. (2000a). Two-loop real-coded genetic algorithms with adaptive control of mutation step sizes. *Applied Intelligence*, 13: 187–204.
- Herrera, F. and Lozano, M. (2000b). Gradual distributed real-coded genetic algorithms. *IEEE Transactions on Evolutionary Computation* 4(1): 43–63.
- Herrera, F., Lozano, M., Pérez, E., Sánchez, A. M. and Villar, P. (2002). Multiple crossover per couple with selection of the two best offspring: an experimental study with the BLX- α crossover operator for real-coded genetic algorithms. In Garijo, F.J., Riquelme, and J.C., Toro, M., editors, *IBERAMIA 2002, Lecture Notes in Artificial Intelligence 2527*, pages 392–401, Springer-Verlag, Berlin.
- Herrera, F., Lozano, M. and 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.
- Houck, C. R., Joines, J. A., Kay, M. G. and Wilson, J. R. (1997). Empirical investigation of the benefits of partial lamarckianism. *Evolutionary Computation Journal*, 5(1): 31–60.
- Huang, C.-F. (2001). An analysis of mate selection in genetic algorithms. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 766, Morgan Kaufmann, San Mateo, California.
- Joines, J. A. and Kay, G. M. (2002). Hybrid genetic algorithms and random linkage. In *Proceedings of the 2002 Congress on Evolutionary Computation*, pages 1733–1738, IEEE Press, Piscataway, New Jersey.
- Jones, T. (1995). Crossover, macromutation, and population-based search. In Eshelman, L., editor, *Proc. of the Sixth Int. Conf. on Genetic Algorithms*, pages 73–80, Morgan Kaufmann, San Francisco, California.
- Kazarlis, S. A., Papadakis, S. E., Theocharis, J. B. and Petridis, V. (2001). Microgenetic algorithms as generalized hill-climbing operators for GA Optimization. *IEEE Transaction on Evolutionary Computation*, 5(3): 204–217.
- Kemenade, C. H. M. (1996). Cluster evolution strategies, enhancing the sampling density function using representatives. In *Proceedings of the Third IEEE conference on Evolutionary Computation*, pages 637–642. IEEE Press, Piscataway, New Jersey.
- Kita, H., Ono, I. and Kobayashi, S. (1999). Multi-parental extension of the unimodal normal distribution crossover for real-coded genetic algorithms. In *Proc. of the International Conference on Evolutionary Computation'99*, pages 646–651, IEEE Press, Piscataway, New Jersey.
- Kita, H. (2001). A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms. *Evolutionary Computation Journal*, 9(2): 223–241.
- Krasnogor, N. (2002). *Studies on the Theory and Design Space of Memetic Algorithms*. Ph. D. Thesis, University of the West of England, Bristol, United Kingdom.
- Krasnogor, N. and Smith, J. E (2000). A Memetic Algorithm with Self-Adapting Local Search: TSP as a case study. *Proceedings of the 2000 International Conference on Genetic and Evolutionary Computation*, pages 987–994, Morgan Kaufmann, San Mateo, California.
- Krasnogor, N. and Smith J. E. (2001). Emergence of Profitable Search Strategies Based on a Simple Inheritance Mechanism. In *Proceedings of the 2001 International Conference on Genetic and Evolutionary Computation*, pages p. 432–439, Morgan Kaufmann, San Mateo, California.
- Land, M. W. S. (1998). *Evolutionary Algorithms with Local Search for Combinatorial Optimisation*. Ph.D. Thesis, University of California, San Diego.

- Magyar, G., Johnsson, M. and Nevalainen, O. (2000). An adaptive hybrid genetic algorithm for the three-matching problem. *IEEE Transactions on Evolutionary Computation*, 4(2): 135–146.
- Merz, P. (2000). *Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies*. Ph. D. Thesis, University of Siegen, Germany.
- Moscato, P. A. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical Report Caltech Concurrent Computation Program Report 826, Caltech, Pasadena, California.
- Moscato, P. A. (1999). Memetic algorithms: a short introduction. In Corne, D., Dorigo, M., and Glower, F., editors, *New Ideas in Optimization*, pages 219–234, McGraw-Hill, London.
- Mühlenbein, H., Schomisch, M. and Born, J. (1991). The parallel genetic algorithm as function optimizer. In Belew, R., Booker, L. B., editors, *Fourth Int. Conf. on Genetic Algorithms*, pages 271–278, Morgan Kaufmann, San Mateo, California.
- Mühlenbein, H. and Schlierkamp-Voosen, D. (1993). Predictive models for the breeder genetic algorithm I. continuous parameter optimization. *Evolutionary Computation*, 1: 25–49.
- Nagata, Y. and Kobayashi, S. (1997). Edge assembly crossover: a high-power genetic algorithm for the traveling salesman problem. In Bäck, T., editor, *Proc. of the Seventh Int. Conf. on Genetic Algorithms*, pages 450–457, Morgan Kaufmann, San Mateo, California.
- O'Reilly, U.-M. and Oppacher F. (1995). Hybridized crossover-based search techniques for program discovery. In *IEEE International Conference on Evolutionary Computation 1995*, pages 573–578, IEEE Press, Piscataway, New Jersey.
- Parthasarathy, P. V., Goldberg, D. E. and Burns, S. A. (2001). Tackling multimodal problems in hybrid genetic algorithms. In Whitley, D., editor, *Proc. of the Genetic and Evolutionary Computation Conference 2001*, pages 775, Morgan Kaufmann, San Francisco, California.
- Rechenberg, I. (1975). *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog.
- Renders, J. M. and Bersini, H. (1994). Hybridizing genetic algorithms with hill-climbing methods for global optimization: two possible ways. In *Proc. of The First IEEE Conference on Evolutionary Computation*, pages 312–317, IEEE Press, Piscataway, New Jersey.
- Renders J.-M. and Flasse, S. P. (1996). Hybrid methods using genetic algorithms for global optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 26(2): 243–258.
- Ronald, E. (1993). When selection meets seduction. In Forrest, S., editor, *Proc. of the Fifth Int. Conf. on Genetic Algorithms*, pages 167–173, Morgan Kaufmann, San Mateo, California.
- Rosin, C. D., Halliday, R. S., Hart, W. E. and Belew, R. K. (1997). A comparison of global and local search methods in drug docking. In Bäck, T., editor, *Proc. of the Seventh Int. Conf. on Genetic Algorithms*, pages 221–228, Morgan Kaufmann, San Mateo, California.
- Salomon, R. (1998). Evolutionary algorithms and gradient search: Similarities and differences. *IEEE Transactions on Evolutionary Computation*, 2(2): 45–55.
- Satoh, H., Yamamura, M. and Kobayashi, S. (1996). Minimal generation gap model for GAs considering both exploration and exploitation. In *IIZUKA'96*, pages 494–497.
- Schaffer D., Mani M., Eshelman L. and Mathias K. (1999). The Effect of Incest Prevention on Genetic Drift. In Banzhaf, W. and Reeves, C., editors, *Foundations of Genetic Algorithms 5*, pages 235–243, Morgan Kaufmann, San Mateo, California.
- Schlierkamp-Voosen, D. and Mühlenbein, H. (1994). Strategy adaptation by competing subpopulations. In Davidor, Y., Schwefel, H.-P., Männer, and R., editors, *Parallel Problem Solving from Nature 3*, pages 199–208, Springer-Verlag, Berlin, Germany.

- Schwefel, H.-P. (1981). *Numerical Optimization of Computer Models*, Wiley, Chichester.
- Seront, G. and Bersini, H. (2000). A new GA-local search hybrid for continuous optimization based on multi level single linkage clustering. In Whitley, D., editor, *Proc. of the Genetic and Evolutionary Computation Conference 2000*, pages 90-95, Morgan Kaufmann, San Francisco, California.
- 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*, pages 25-32, IEEE Computer Society.
- Smith, J. E. and Fogarty, T. C. (1997). Operator and parameter adaptation in genetic algorithms. *Soft Computing*, 1(2): 81-87.
- Smith, J. E. (2001). Modelling GAs with Self-Adapting Mutation Rates. *Proc. of the Genetic and Evolutionary Computation Conference 2001*.
- Solis, F. J. and Wets, R. J.-B. (1981). Minimization by random search techniques. *Mathematical Operations Research*, 6: 19-30.
- Spears, W. M. (1993). Crossover or mutation?. In Whitley, L. D., editor, *Foundations of Genetic Algorithms 2*, pages 221-238, Morgan Kaufmann, San Mateo, California.
- Spears, W. M. (1995). Adapting crossover in evolutionary algorithms. In *Proc. of the Evolutionary Programming Conference 1995*, pages 367-384.
- Storn, R. and Price, K. (1995). Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley, California.
- Törn, A. and Antanas, Ž. (1989). *Global Optimization*, Lecture Notes in Computer Science, 350, Springer, Berlin, Germany.
- Tsutsui, S. and Fujimoto, Y. (1993). Forking genetic algorithm with blocking and shrinking modes. In Forrest, S., editor, *Proc. of the Fifth Int. Conf. on Genetic Algorithms*, pages 206-213, Morgan Kaufmann, San Mateo, California.
- Tsutsui, S., Yamamura, M. and Higuchi, T. (1999). Multi-parent recombination with simplex crossover in real-coded genetic algorithms. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO-99)*, 657-664, Morgan Kaufmann, San Mateo, California.
- Walters, T. (1998). Repair and brood selection in the traveling salesman problem. In Eiben, A. E., Bäck, T., Schoenauer, M. and Schwefel H.-P., editors, *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature*, pages 813-822, Springer-Verlag, Berlin, Germany.
- Whitley, D., Rana, S., Dzubera, J. and Mathias, E. (1996). Evaluating evolutionary algorithms. *Artificial Intelligence*, 85: 245-276.
- Yang, J.-M. and Kao, C.-Y. (2000). Integrating adaptive mutations and family competition into genetic algorithms as function optimiser. *Soft Computing*, 4: 89-102.
- Yoon, H.-S. and Moon, B.-R. (2002). An empirical study on the synergy of multiple crossover operators. *IEEE Transactions on Evolutionary Computation*, 6(2): 212-223.
- Zhang, C.-K. and Shao, H.-H. (2001). A hybrid strategy: real-coded genetic algorithm and chaotic search. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics 2001*, pages 2361-2364.