Real-Parameter Genetic Algorithms for Finding Multiple Optimal Solutions in Multi-modal Optimization

Pedro J. Ballester and Jonathan N. Carter

Imperial College London, Department of Earth Science and Engineering, London SW7 2AZ, UK

Abstract. The aim of this paper is to identify Genetic Algorithms (GAs) which perform well over a range of continuous and smooth multi-modal real-variable functions. In our study, we focus on testing GAs combining three classes of genetic operators: selection, crossover and replacement. The approach followed is time-constrained and thus our stopping criterion is a fixed number of generations. Results show that GAs with random selection of parents and crowding replacement are robust optimizers. By contrast, GAs with tournament selection of parents and random replacement perform poorly in comparison.

1 Introduction

Genetic Algorithms (GAs) have proven to be a useful approach to address a wide variety of optimization problems. However, different problems usually require different combinations of GA parameters (population size, number of generations, choice of genetic operators, etc.). Unfortunately, GA parameters interact in complex ways, making the task of finding a suitable parameter set not always straightforward. In addition, a GA which excels with a given class of problems might yield poor results when applied to another class (Hart and Belew [11]).

This situation has given rise to the need for studies providing guidance in setting the GA parameters for different class of problems. As taking into account all the parameters is impractical, one usually focus on two or three parameters and try to understand their interactions. For instance, Deb and Agrawal [4], using a fixed set of binary genetic operators, studied the influence of population size, crossover probability and mutation probability over several classes of functions, which are known to be difficult to optimise.

The aim of this paper is to identify GAs which performs well over a specific class of functions: continuous, smooth, multi-modal real-variable functions. We say that a GA performs well if it has a good chance of finding the global minimum (we restrict without loss of generality to the minimization case). We are also interested in GAs that are able to find alternative good minima while reaching the global minimum too. There are two main reasons to set this secondary goal. First, real-world functions do not come without errors, which distort the fitness landscape. Therefore, a secondary minimum might actually be the global

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2723, pp. 706–717, 2003.
© Springer-Verlag Berlin Heidelberg 2003

one. Second, the optimal solution represented by the global minimum might be impossible to implement from the engineering point of view. In this case, a secondary good solution can be used as an alternative.

In our study, we focus on testing the performance of GAs combining three classes of genetic operators: selection, crossover and replacement. The approach followed is time-constrained and thus our stopping criterion is a fixed number of generations.

We arrange the rest of the paper as follows. In section 2, we describe the problem of multi-modal optimisation and present a set of test functions. Section 3 describes the structure of the GAs tested in this paper. In section 4, we explain the experimental setup introducing a performance measure. Results are discussed in section 5. Lastly, we present our conclusions in section 6.

2 GAs in Multi-modal Optimisation

Most of the traditional approaches to optimisation such as gradient-based methods are local optimisers. These sort of methods find the minimum of the basin of attraction in which the initial point is situated. Multi-modal optimisation is characterized by having a high number of attractors. Therefore, it is unlikely that the initial point lies within the basin containing the global minimum.

GAs are known to be a particularly good approach for multi-modal optimisation. Unlike local optimisers, GAs are stochastic approaches and thus are not strongly influenced by the initial population. The choice of GA parameters determines the balance of exploration of search space and the exploitation of information contained in the population. It is the adequacy of this balance that leads to the successful application of a GA.

Traditionally, the success of a GA run is verified by checking if there are individuals within a tight ϵ-neighbourhood of the minimum. By contrast, we are interested in finding the basin of attraction that contains the minimum. The motivation behind this definition of success is that of reducing the number of function evaluations needed. In practice, it is normally faster to find the minimum basin and then apply a fast local optimiser. This is a useful approach when the fitness evaluation carries a high computational cost.

Our main purpose is to find a robust optimiser for continuous, smooth, multi-modal real-variable functions. A particular GA is said to be robust if performs well on a given class of functions. As we cannot possibly consider every function in a class, we restrict our tests to a set of functions which represent the class. These test functions are defined in Figs. 1 to 6. The right-side plots corresponding to their two-dimensional views have been inverted in all cases to allow a better visualisation.

Each of these test functions has many local minima and one single global minimum. Functions 1, 2 and 3 are taken from previous studies [1] [8], while functions 4, 5 and 6 were constructed to be non-symmetric and composed by subfunctions which do not share minima with the whole function.

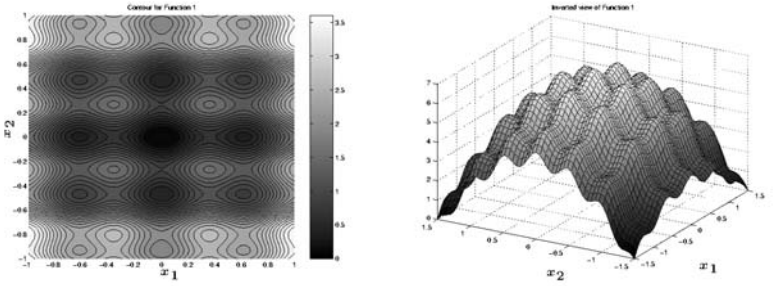


Fig. 1. $F_1(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$

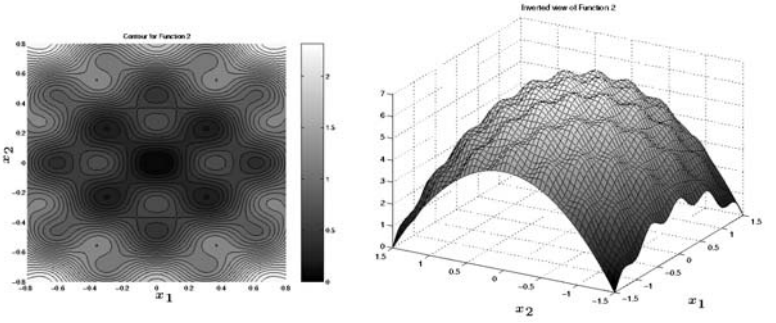


Fig. 2. $F_2(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3(\cos(3\pi x_1)\cos(4\pi x_2)) + 0.3$

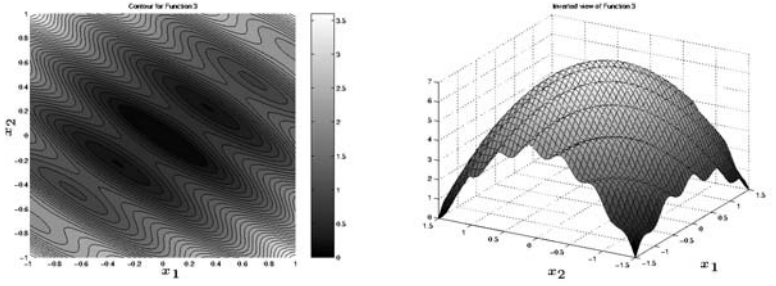


Fig. 3. $F_3(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$

3 Real-Parameter GAs

In this section, we describe the genetic operators used in our study. We aim to investigate the effect of three GA parameters (choice of selection scheme, replacement scheme and crossover operator) on the performance in multi-modal optimisation.

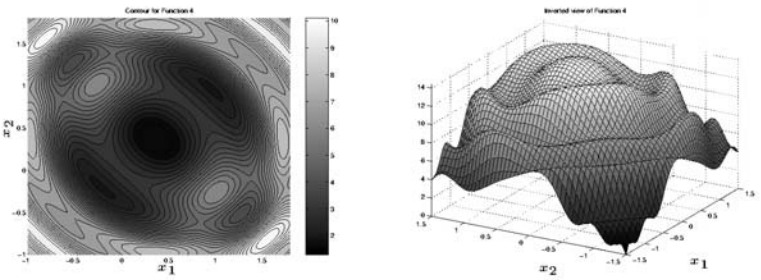


Fig. 4. $F_4(x_1, x_2) = \sum_{j=1}^{2}(x_1 - d_j)^2 + (x_2 - d_j)^2 - \cos(\pi((x_1 - d_j)^2 + (x_2 - d_j)^2)) + 1$ with $d_1 = 0$ and $d_2 = 0.75$.

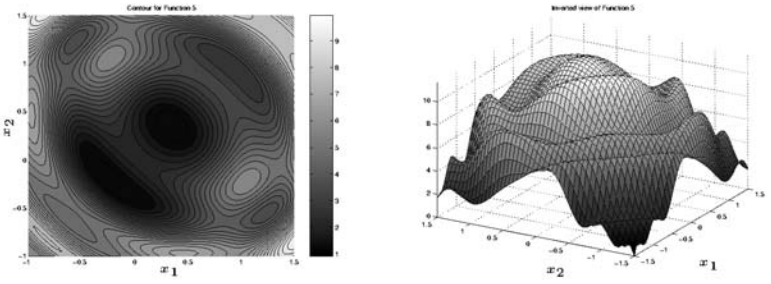


Fig. 5. $F_5(x_1, x_2) = \sum_{j=1}^{2}(x_1 - \frac{d_j}{2})^2 + (x_2 - \frac{d_j}{2})^2 - \cos(\pi((x_1 - d_j)^2 + (x_2 - d_j)^2)) + 1$ with $d_1 = 0$ and $d_2 = 0.75$.

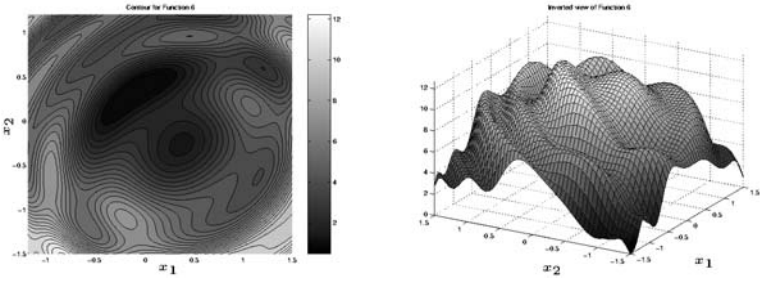


Fig. 6. $F_6(x_1, x_2) = \sum_{j=1}^{2}(x_1 - \frac{d_j}{2})^2 + (x_2 - \frac{d_j}{2})^2 - \cos(\pi((x_1 - d_j)^2 + (x_2 + d_j)^2)) + 1$ with $d_1 = 0$ and $d_2 = 0.75$.

It has been discussed elsewhere [2] that real-parameter GAs are ideally suited to handle problems in a continuous search space. The encoding scheme is that the genome of each individual is composed of two real variables (x_1, x_2), whose values represent a location on the search space. In order to avoid favouring a particular solution, the initial population is generated at random within the

range $-10 \leq x_1, x_2 \leq 10$. All the functions have their global minima within this range. An individual fitness is given by its function value.

In our GA architecture, a population of individuals enter the following loop for a fixed number of generations. First, three groups of parents are selected among the actual population. Second, each group of parents breeds two children, giving place to an offspring population of six new individuals. Lastly, offspring and parent populations are combined following a given replacement scheme to form a new population. We use a population size of 20, and terminate the process after 100 generations. This results in a total of 620 function evaluations.

We now present the different GA operators that we test in this paper:

3.1 Selection Schemes

We test two schemes: random (S_1) and tournament (S_2) selection [9]. In each scheme we generate three groups of parents. The parents within any of the groups are different from each other, although an individual may occur in more than one group. The number of parents within a group depends on the crossover operator used.

In random selection, the three groups of individuals are selected at random from the parent population. Whereas in tournament selection, two members of the parent population are picked at random and the fittest among them selected as a parent. The latter procedure is repeated until the required number of parents have been selected.

3.2 Crossover Schemes

Each of the three groups of parents produce two children, through the crossover operator, which are bred independently. This results in six new individuals per generation, which is an arbitrary choice and no claim about its optimality is made. Crossovers for real-parameter GAs have the interesting feature [2] of having tunable parameters that can be used to modify their exploration power. Thus, we use three different arbitrary parameter values for each crossover. The following four crossover operators are tested:

BLX. The Blend crossover (BLX) operator [7] randomly selects a value for each offspring gene y_i, using a uniform distribution within the range

$$[x_i^{(1)} - \alpha(x_i^{(2)} - x_i^{(1)}),\ x_i^{(2)} + \alpha(x_i^{(2)} - x_i^{(1)})]$$

where $x_i^{(1)}$ and $x_i^{(2)}$ are the parental genes, and α is the tunable parameter, we use $\alpha \in \{0.1, 0.5, 10\}$, the higher the value of α the more explorative the search.

SBX. We use the Simulated Binary crossover (SBX) operator [3] [5] with independently bred children. Therefore, for each children, we choose at random between the equally likely expressions $y_i = 0.5((1+\beta_i)x_i^{(1)} + (1-\beta_i)x_i^{(2)})$ and

$y_i = 0.5((1-\beta_i)x_i^{(1)} + (1+\beta_i)x_i^{(2)})$, where y_i is the offspring gene, $x_i^{(1)}$ and $x_i^{(2)}$ are the parental genes, and β_i a parameter. Then, we generate, for each coordinate, a second random number $u_i \in [0,1]$ with β_i given by

$$\beta_i = \begin{cases} (2u_i)^{\frac{1}{\eta+1}} & 0 < u_i \leq 0.5 \\ (1/2(1-u_i))^{\frac{1}{\eta+1}} & 0.5 < u_i \leq 1 \end{cases}$$

where η is the tunable parameter. A restricted search is achieved with a large value of η (we use $\eta \in \{0.1, 10, 150\}$). A plot of the distribution of children is given by Fig. 7.

vSBX. In SBX, children have zero probability of appearing in some regions of the parameter space, as shown in Fig. 7. A version of the latter crossover, called vSBX, is proposed that does not exclude any regions of the parameter space, while preserving the good SBX properties. This may allow a better exploration of the search space. In vSBX, you generate a random number $u \in [0,1]$ to make the two following choices (only one value of u is needed per child). First, you choose between

$$y_i^{(1)} = \begin{cases} 0.5((1+\beta_{1i})x_i^{(1)} + (1-\beta_{1i})x_i^{(2)}) & 0 < u \leq 0.5 \\ 0.5((1-\beta_{1i})x_i^{(1)} + (1+\beta_{1i})x_i^{(2)}) & 0.5 < u \leq 1 \end{cases}$$

with $\beta_{1i} = (1/2u_i)^{\frac{1}{\eta+1}}$. Second, you choose between

$$y_i^{(2)} = \begin{cases} 0.5((3-\beta_{2i})x_i^{(1)} - (1-\beta_{2i})x_i^{(2)}) & 0 < u \leq 0.5 \\ 0.5(-(1-\beta_{2i})x_i^{(1)} + (3-\beta_{2i})x_i^{(2)}) & 0.5 < u \leq 1 \end{cases}$$

with $\beta_{2i} = (1/2(1-u_i))^{\frac{1}{\eta+1}}$. u_i is a second random number that is used to choose between $y_i^{(1)}$ and $y_i^{(2)}$. The latter provides the offspring gene value y_i. A restricted search is achieved with a large value of η (we use $\eta \in \{0.1, 10, 150\}$).

UNDX. The Unimodal Normally Distributed crossover (UNDX) operator [13] changes every gene simultaneously by making use of a multivariate normal distribution. UNDX is unusual in that it needs three non-identical parents. The first two parents are used to define a dominant axis through the search space of real numbers. The crossover is carried out on a co-ordinate system that is rotated to coincide with this dominant axis. The distance between the two parents is used to define the standard deviation of a normal distribution along this dominant axis. The first parameter, a, is introduced as a non-negative real number multiplying the latter standard deviation. The distance of the third parent from this dominant axis is used to define the standard deviation used in that direction. The second parameter, b, is a non-negative real number multiplying the latter standard deviation. The reader is referred to the original paper (Ono and Kobayashi [13]) for details of the implementation.

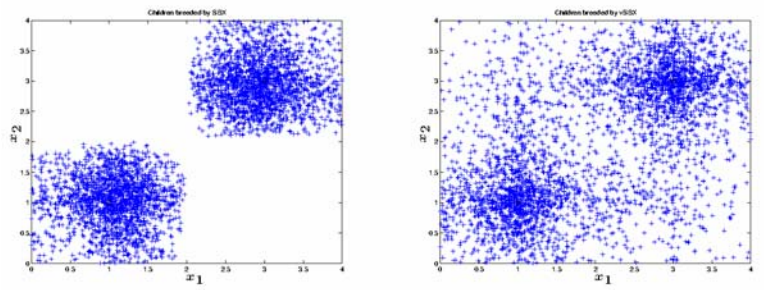


Fig. 7. Children bred from parents $\boldsymbol{x}^{(1)} = (1,1)$ and $\boldsymbol{x}^{(2)} = (3,3)$ for a) SBX ($\eta = 1$) and b) vSBX ($\eta = 1$)

The authors recommend $a = 1$ and $b = 0.5$. We keep the latter ratio between these two parameters, while changing the value of a. We test UNDX with parameter duples (a, b) equal to $(1, 0.5)$, $(0.5, 0.25)$ and $(0.25, 0.125)$, with the first $(1, 0.5)$ carrying the higher exploration and $(0.25, 0.125)$ the lower one.

3.3 Replacement Schemes

Five replacement schemes are tested: random (R_1), tournament (R_2), modified probabilistic crowding (R_3), classic probabilistic crowding (R_4) and deterministic crowding (R_5). In each of them, the offspring population is combined with the parent population to form a new one for the next generation. This new population is constructed to have the same size as the original population.

In Random replacement, six individuals from the current population are selected at random and completely replaced by the offspring population.

Tournament and crowding schemes operate in two steps. First, a part of the parent population is preselected at random. The size of this preselected group is taken as two in this work. Thereafter, the chosen scheme is followed to find out whether the offspring replaces one of the preselected individuals or not. The described operation is repeated for every offspring.

In the tournament scheme [10], the least fit of the preselected group competes with the offspring. Then, the fitter among them remains in the population. In the deterministic crowding [6], the preselected individual with the lower euclidean distance to the offspring competes with it. The fitter of them remains in the population.

The probabilistic crowding scheme [12] identifies the closest preselected individual ($\boldsymbol{x}^{cst}$) to enter a probabilistic tournament with the offspring ($\boldsymbol{x}^{ofp}$), with survival likelihoods given by $p(\boldsymbol{x}^{ofp}) = f(\boldsymbol{x}^{ofp})/(f(\boldsymbol{x}^{ofp}) + f(\boldsymbol{x}^{cst}))$ and $p(\boldsymbol{x}^{cst}) = f(\boldsymbol{x}^{cst})/(f(\boldsymbol{x}^{ofp}) + f(\boldsymbol{x}^{cst}))$.

Note that if the global minimum has a very high value with respect to the differences in fitness across the population, these likelihoods would be very similar in all cases. The modified probabilistic crowding is introduced to avoid this

situation. It operates with survival likelihoods $p(\boldsymbol{x}^{ofp}) = f(\boldsymbol{x}^{ofp})/(f(\boldsymbol{x}^{ofp}) + f(\boldsymbol{x}^{cst}) - 2f_{worst})$ and $p(\boldsymbol{x}^{cst}) = f(\boldsymbol{x}^{cst})/(f(\boldsymbol{x}^{ofp}) + f(\boldsymbol{x}^{cst}) - 2f_{worst})$, where f_{worst} is the fitness of the worst individual in the offspring and preselected group.

4 Experimental Setup

We define a 'GA experiment' by the ith selection scheme S_i (see section 3.1), the jth replacement scheme R_j (see section 3.3), the kth crossover operator C_k (see section 3.2) and the lth objective function F_l (see figures 1, 2, 3, 4, 5 and 6). As an experiment has some dependence on the initial conditions, we repeat each possible experiment with a different initial population (50 different sets of initial population are used).

We need to set boundaries to the main basins of attraction in order to evaluate the success of a run. This is done by adjusting an ellipse to the projection of the basin onto the search space. Thus, an individual lies within a basin defined by (c_1, c_2, θ, a, b) if

$$\frac{((x_1 - c_1)\cos\theta + (x_2 - c_2)\sin\theta)^2}{a^2} + \frac{((x_2 - c_2)\cos\theta - (x_1 - c_1)\sin\theta)^2}{b^2} \leq 1 \tag{1}$$

where a and b are the ellipse semiaxis, (c_1, c_2) is the ellipse center and θ is the angle between the ellipse and search space coordinate systems. Figure 8 shows the contour of function 1 with the global basin (represented by the central ellipse) and a set of basins containing the best alternative minima. In table 1 are the basins used for all of the test functions.

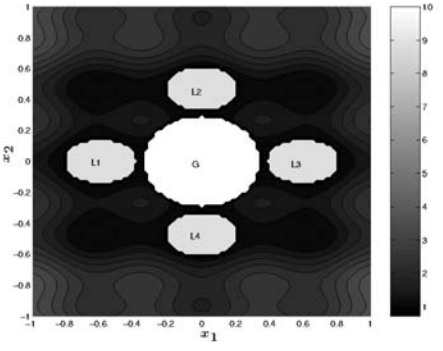


Fig. 8. Basins containing the best minima of function 1.

Finally, we define some performance indicators to study the success of each experiment. First, we define the primary success ratio (psr) for an experiment (S_i, R_j, C_k, F_l) as one, if at least one individual in the final population is within

Table 1. Defining values of each of the marked basins. The name of a basin has a G if includes the global minimum or L if includes a local minimum. θ is in radians.

	F_1					F_2					F_3			F_4			F_5		F_6	
	G	L1	L2	L3	L4	G	L1	L2	L3	L4	G	L1	L2	G	L1	L2	G	L1	G	L1
c_1	0	-.6	0	.6	0	0	-.3	-.3	.3	.3	0	-.35	.35	.37	-.15	.9	-.12	.35	-.03	.41
c_2	0	0	.47	0	-.47	0	-.2	.2	.2	-.2	0	-.25	.25	.37	-.15	.9	-.12	.35	.22	-.32
θ	0	0	0	0	0	0	0	0	0	0	.9	.9	.9	$\pi/4$	$\pi/4$	$\pi/4$	$\pi/4$	$\pi/4$	-.89	-.89
a^2	.35	.22	.22	.22	.22	.2	.15	.15	.15	.15	.2	.2	.2	.4	.3	.3	.3	.35	.3	.35
b^2	.3	.15	.15	.15	.15	.25	.1	.1	.1	.1	.9	.9	.9	.5	.8	.8	.8	.45	.85	.45

the basin containing the global minimum and zero otherwise. As we are also interested in multiple solutions, we introduce the secondary success ratio (ssr) for an experiment (S_i, R_j, C_k, F_l) as one, if $psr = 1$ and, in addition, at least one individual in the final population is within any of the marked alternative basins and zero otherwise.

5 Results

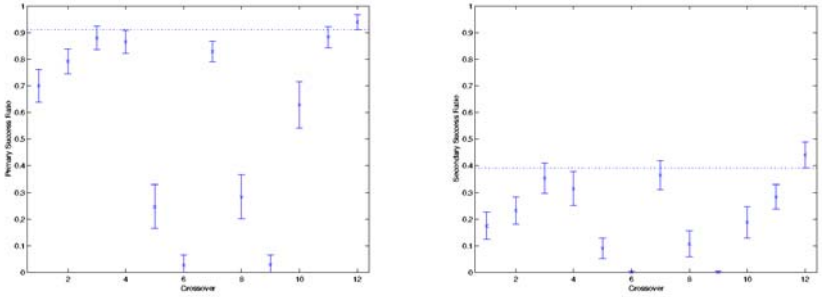
In this section, we present performance results of the several GA simulations carried out.

In order to investigate the performance of each crossover, let us take the case of the indicator primary success ratio (psr) as example and proceed as follows. First, for each of the 12 crossover operators C_k considered we sum over all possible S_i, R_j and F_l to dissipate the influence of selection, replacement and function over the crossover operator performance. Thereafter, we divide by the number of available schemes to normalise the resulting number (we do not include the case of random selection combined with random replacement). This procedure is contained in the following expression

$$psr(C_k) = \frac{1}{(N_S N_R - 1)N_F} \sum_i \sum_j \sum_l psr(S_i, R_j, C_k, F_l) \tag{2}$$

As each experiment was run 50 times, we can assess the uncertainty in equation 2 by providing an average and an standard deviation. Figure 9 shows the average of both psr and ssr (represented by 'x') and their corresponding estimation of uncertainty as twice the standard deviation (represented as a vertical interval). We also trace an horizontal line in each plot with the value of the highest average minus its standard deviation in order to allow an easier comparison. From Fig. 9, we see that those operators with the higher exploration power in their respective groups (ie. BLX-10, SBX-0.1, vSBX-0.1 and UNDX-(1,0.5)) outperforms the rest both in terms of psr and ssr. This result suggests that multi-modal optimisation requires extensive searches in order to find good minima.

Let us now compare the available combinations of selection and replacement schemes stated at the legend of Fig. 10. For each of these combinations (except



C_k	par (k)	par (k)	par (k)
BLX	0.1 (1)	0.5 (2)	10 (3)
SBX	0.1 (4)	10 (5)	150 (6)

C_k	par (k)	par (k)	par (k)
vSBX	0.1 (7)	10 (8)	150 (9)
UNDX	0.1 (10)	0.5 (11)	1 (12)

Fig. 9. Average and standard deviation of both performance indicators for each crossover.

again for combination one), we now dissipate the influence of crossover and function as follows

$$psr(S_i, R_j) = \frac{1}{N_C N_F} \sum_k \sum_l psr(S_i, R_j, C_k, F_l) \tag{3}$$

Figure 10 shows the average and standard deviation over 50 different realisations of each available combination. It is observed that the GA with tournament selection and random replacement (combination 6) has the lowest probability of finding the global minimum. In addition, it does not perform well in finding more than one good minima, as we see from its ssr value. By contrast, random selection of parents combined with crowding replacement strategies (combinations 3, 4 and 5) seems to be more advisable approach for the class of problems we are dealing with. On the other hand, our proposed modified probabilistic crowding has a similar performance to the classical probabilistic crowding. Lastly, combinations 3, 4 and 5 perform better than 8, 9 and 10, which might be caused by too much exploitation in the latter group. Next, we want to study the performance of each GA implementation individually. Hence, for each combination of selection, crossover and replacement, we sum the indicator value over all possible functions as described in equation 4

$$psr(S_i, R_j, C_k) = \frac{1}{N_F} \sum_l psr(S_i, R_j, C_k, F_l) \tag{4}$$

Thereafter, we calculate the indicator average and standard deviation for each of these tripartite combinations. This operation leads to two remarkable findings. For all of the crossover operators, the combination tournament selection (S_2) and random replacement (R_1) gives poor results. On the other hand, the highest psr (0.997 ± 0.024) and ssr (0.847 ± 0.138) is achieved by S_1, R_3 and C_{12}. Followed by S_1, R_5 and C_{12}, with psr=0.993 ± 0.033 and ssr=0.810 ± 0.151.

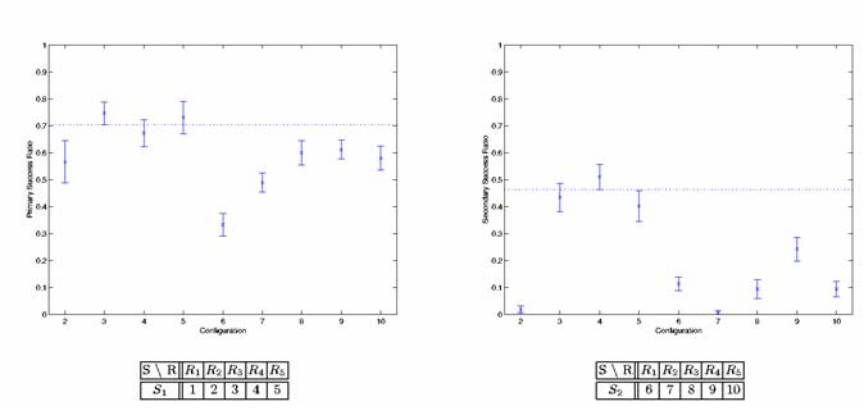


Fig. 10. Average and standard deviation of performance indicators for each combination of selection and replacement.

It is worth noting that all these results might depend on the number of generations. For instance, the GA with the best performance after 100 generations might be outperformed by another as the number of generations increase. This situation would be due to distinct convergence rates. Also, none of the implemented GAs is expected to indefinitely maintain subpopulations. Therefore, ssr would eventually reach the zero value, again at different rates for each GA.

In order to examine this question, we repeated our experiments with 300 generations (1820 function evaluations). Our main results are very similar to those with 100 generations. In general, all psr are marginally improved while ssr undergoes a large drop. Specifically, S_2 and R_1 has again the worst psr for most crossovers (all except C_{11} and C_{12}). The highest performance (psr=1 and ssr=0.150 ± 0.118) corresponds still to S_1, R_3 and C_{12}, but now is shared with S_1, R_4 and C_{12} (psr=1 and ssr=0.567 ± 0.165). These two followed by S_1, R_3 and C_{12} (psr=0.997 ± 0.024 and ssr=0.233 ± 0.154), which previously obtained the second position.

6 Conclusions

Based on the set of experiments carried out, we arrive to the following conclusions for the multi-modal function optimisation problem:

- GAs using crossover operators with the highest search power obtain the best performance.
- GAs with tournament selection of parents and random replacement are not recommended because they have a poor performance.
- GAs with random selection of parents and crowding replacement are robust optimisers for this class of functions. In addition, they have a good chance of finding alternative good minima.

In this work, the number of new individuals per generation is fixed. We consider that future work analysing the influence of this variable is worthwhile. In addition, we plan to reproduce this benchmark for multi-modal functions in higher dimensions.

References

1. Bohachevsky, I.O., Johnson, M.E., and Stein, M.L. (1986): *Generalized Simulated Annealing for Function Optimization.* Technometrics 28(3), pp. 209–218.
2. Deb, K. (2001): *Multi-objective optimization using evolutionary algorithms.* 1st ed. (John Wiley & Sons, Ltd.).
3. Deb, K. and Agrawal, S. (1995): *Simulated binary crossover for continuous search space.* Complex Systems 9(2), pp. 115–148.
4. Deb, K. and Agrawal, S. (1999): *Understanding Interactions Among Genetic Algorithm Parameters.* In: Foundations of Genetic Algorithms 5, Ed. Wolfgang Banzhaf and Colin Reeves (Morgan Kaufmann), pp. 265–286.
5. Deb, K. and Kumar, A. (1995): *Real-coded genetic algorithms with simulated binary crossover: Studies on multi-modal and multi-objective problems.* Complex Systems 9(6), pp. 431–454.
6. DeJong, K.A. (1975): *An Analysis of the Behavior of a Class of Genetic Adaptative Systems.* Ph. D. Thesis (University of Michigan, Ann Arbor).
7. Eshelman, L.J. and Schaffer, J.D. (1993): *Real coded genetic algorithms and interval schemata.* In: Foundations of Genetic Algorithms 2, Ed. L. Darrell Whitley (Morgan Kaufmann), pp. 187–202.
8. Fogel, D.B. (1995): *Evolutionary computation: toward a new philosophy of machine intelligence* (Institute of Electrical and Electronics Engineers, Inc.).
9. Goldberg, D.E. and Deb, K. (1991): *A comparative analysis of selection schemes used in genetic algorithms.* In: Foundations of Genetic Algorithms, Ed. Gregory J.E. Rawlings (Morgan Kaufmann), pp. 69–93.
10. Harik, G.R. (1995): *Finding multi-modal solutions using restricted tournament selection.* In: Proceedings of the Sixth International Conference on Genetic Algorithms, Ed. L.J. Eshelman (Morgan Kaufmann), pp. 24–31
11. Hart, W.E. and Belew, R.K. (1991): *Optimizing an Arbitrary Function is Hard for the Genetic Algorithm.* In: Proceedings of the Fourth International Conference on Genetic Algorithms, Ed. Richard K. Belew and Lashon B. Booker (Morgan Kaufmann), pp. 190–195.
12. Mengshoel, O.J. and Goldberg, D.E. (1999): *Probabilistic Crowding: Deterministic Crowding with Probabilistic Replacement.* In: Proceedings of the Genetic and Evolutionary Computation Conference, Ed. W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela and R.E. Smith (Morgan Kauffmann), pp. 409–416.
13. Ono, I. and Kobayashi, S. (1997): *A real-coded genetic algorithm for function optimization using unimodal normal distribution.* In: Proceedings of the Seventh International Conference on Genetic Algorithms, Ed. Thomas Bäck (Morgan Kauffmann), pp. 246–253.