

# A Taxonomy for the Crossover Operator for Real-Coded Genetic Algorithms: An Experimental Study

F. Herrera,<sup>1,\*</sup> M. Lozano,<sup>1,†</sup> A. M. Sánchez<sup>2,‡</sup>

<sup>1</sup>*Department of Computer Science and Artificial Intelligence, University of Granada, 18071, Granada, Spain*

<sup>2</sup>*Departamento de Informática, Escuela Superior de Ingeniería Informática, University of Vigo, 32004, Orense, Spain*

The main real-coded genetic algorithm (RCGA) research effort has been spent on developing efficient crossover operators. This study presents a taxonomy for this operator that groups its instances in different categories according to the way they generate the genes of the offspring from the genes of the parents. The empirical study of representative crossovers of all the categories reveals concrete features that allow the crossover operator to have a positive influence on RCGA performance. They may be useful to design more effective crossover models. © 2003 Wiley Periodicals, Inc.

## 1. INTRODUCTION

Genetic algorithms (GAs) are adaptive methods based on natural evolution that may be used for search and optimization problems. They process a population of search space solutions with three operations: selection, crossover, and mutation.<sup>1–3</sup>

Under their initial formulation, the search space solutions are coded using the binary alphabet; however, other coding types have been taken into account for the representation issue such as real coding. The real coding approach seems particularly natural when tackling optimization problems of parameters with variables in continuous domains. A chromosome is a vector of floating point numbers in which their 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).<sup>4,5</sup>

\* Author to whom all correspondence should be addressed: e-mail: herrera@decsai.ugr.es.

† e-mail: lozano@decsai.ugr.es.

‡ e-mail: amlopez@uvigo.es.

Recently, there has been an increasing interest in solving real-world optimization problems using RCGAs. Some of their applications include chemometric problems,<sup>6</sup> neural networks,<sup>7,8</sup> aerospace design,<sup>9–12</sup> biotechnology,<sup>13</sup> control,<sup>14–16</sup> economic,<sup>17</sup> signal processing,<sup>18</sup> microware,<sup>19–21</sup> industrial electronics,<sup>22</sup> industrial engineering,<sup>23,24</sup> tomography,<sup>25</sup> water resources management,<sup>26</sup> and constrained parameter optimization problems.<sup>3</sup>

The crossover operator is a method for sharing information between chromosomes. Generally, it combines the features of two parent chromosomes to form two offspring, with the possibility that good chromosomes may generate better ones. It has always been regarded as the primary search operator in GAs<sup>27–29</sup> because it exploits the available information from the population about the search space. Moreover, it is one of the components to consider for improving the behavior of the GA.<sup>30</sup>

The main RCGA research effort has been spent on developing efficient crossover operators,<sup>4</sup> and as a result, many different instances have been proposed. At this point, a taxonomy for this operator becomes attractive because it will reveal and allow us to find properties that are needed in an effective real-parameter crossover operator. In this study, we propose a taxonomy that groups the models for this operator in different categories according to the way they generate the genes of the offspring from the genes of the parents. Furthermore, we perform an empirical study of representative instances of all the categories, which provides some clues on the key features that have a positive influence on the crossover behavior.

Section 2 introduces some issues related to real-parameter crossover operators. In Section 3, we present the taxonomy for these operators. In Section 4, we describe an experimental study aimed at determining the goodness associated with the different groups. Section 5 includes our conclusions and summarizes a few new promising studies on the topic.

## 2. CROSSOVER OPERATORS FOR RCGAs

In this section, we deal with the main aspects of the crossover operators for RCGAs. In Section 2.1, we explain the three mechanisms involved in the application of the crossover operator. This is useful to establish the particular features of the crossover operators analyzed in this study. In Section 2.2, we define different real-parameter crossover instances that appear in the GA literature. In Section 2.3, we examine the availability of these operators to adopt different exploration or exploitation degrees. Finally, in Section 2.4, we review some attempts made for specifying guidelines for the design of crossover operators for real coding.

### 2.1. The Crossover Operator

The application of the crossover operator is performed by means of three different mechanisms:

- (1) Mating selection mechanism (MSM). The MSM determines the way the chromosomes are mated for applying the crossover to them. The most common MSM pairs the parents randomly. However, other approaches have appeared.<sup>31–33</sup>
- (2) Offspring generation mechanism (OGM). The OGM produces new chromosomes from each set of parents formed by the MSM. All the OGMs proposed for the binary coding may be adapted for working under the real coding. However, this coding offers the possibility of defining a wide variety of special OGMs that take advantage of its numerical nature. Generally, they calculate the value of the genes corresponding to each position in the offspring by combining numerically the values of the genes of the parents in this position.
- (3) Offspring selection mechanisms (OSM). Between all the offspring generated for each set of parents, this mechanism chooses the ones that will be population members. One of the most used OSMs selects the best offspring as elements for the next population.<sup>34–36</sup>

Usually, the crossover operator is applied to pairs of chromosomes, generating two offspring for each one of them, which are introduced in the population.<sup>1</sup> However, multiparent crossover operators have been proposed, which combine the features of more than two parents for generating the offspring.<sup>34,37–39</sup> Furthermore, crossover operators with multiple descendents have been presented,<sup>34–36,40–42</sup> which produce more than two offspring for each group of parents. In this case, the OSM limits the number of offspring that will be population members. All of the offspring may be created using the same OGM<sup>34,42</sup> or by means of different OGMs.<sup>35</sup>

We should emphasize that this study relies on crossover operators for real coding that require only two parents.

## 2.2. Crossover Operators for Real Coding

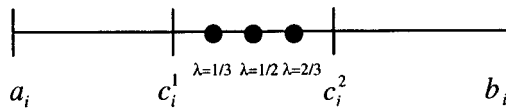
Let us assume that  $C_1 = (c_1^1, \dots, c_n^1)$  and  $C_2 = (c_1^2, \dots, c_n^2)$  are two chromosomes that have been selected to apply the crossover operator to them. In the following list we describe the operation of different crossover operators for RCGAs and show their effects:

Simple crossover.<sup>1,2</sup> A position  $i \in \{1, 2, \dots, n - 1\}$  is chosen randomly and two new chromosomes are built:

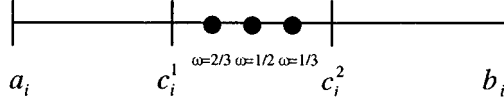
$$H_1 = (c_1^1, c_2^1, \dots, c_i^1, c_{i+1}^2, \dots, c_n^2)$$

$$H_2 = (c_1^2, c_2^2, \dots, c_i^2, c_{i+1}^1, \dots, c_n^1)$$

Two-point crossover.<sup>43</sup> Two-point crossover is a generalization of the simple crossover. Two points of crossover are selected randomly ( $i, j \in \{1, 2, \dots, n - 1\}$  with  $i < j$ ),



**Figure 1.** Arithmetical crossover with different values for  $\lambda$ .



**Figure 2.** Geometrical crossover with different values for  $\omega$ .

and the segments of the parent, defined by them, are exchanged for generating two offspring:

$$H_1 = (c_1^1, c_2^1, \dots, c_i^2, c_{i+1}^2, \dots, c_j^2, c_{j+1}^1, \dots, c_n^1)$$

$$H_2 = (c_1^2, c_2^2, \dots, c_i^1, c_{i+1}^1, \dots, c_j^1, c_{j+1}^2, \dots, c_n^2)$$

Uniform crossover.<sup>44</sup> Two offspring are created,  $H_k = (h_1^k, \dots, h_i^k, \dots, h_n^k)$ ,  $k = 1, 2$ . The value of each gene in the offspring is determined by the random uniform choice of the values of this gene in the parents:

$$h_i^k = \begin{cases} c_i^1 & \text{if } u = 0 \\ c_i^2 & \text{if } u = 1 \end{cases}$$

$u$  being a random number that can have a value of zero or one.

Arithmetical crossover.<sup>3</sup> Two offspring are produced (Figure 1),  $H_k = (h_1^k, \dots, h_i^k, \dots, h_n^k)$ ,  $k = 1, 2$ , where  $h_i^1 = \lambda \cdot c_i^1 + (1 - \lambda) \cdot c_i^2$  and  $h_i^2 = \lambda \cdot c_i^2 + (1 - \lambda) \cdot c_i^1$ , where  $\lambda \in [0, 1]$ .

Geometrical crossover.<sup>45</sup> Two offspring are built (Figure 2),  $H_k = (h_1^k, \dots, h_i^k, \dots, h_n^k)$ ,  $k = 1, 2$ , where  $h_i^1 = c_i^1 \cdot c_i^{2^{1-\omega}}$  and  $h_i^2 = c_i^2 \cdot c_i^{1-\omega}$ , with  $\omega \in [0, 1]$ .

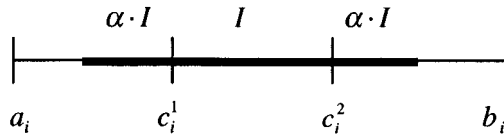
BLX- $\alpha$  (Figure 3).<sup>46,47</sup> Two offspring are generated,  $H_k = (h_1^k, \dots, h_i^k, \dots, h_n^k)$ ,  $k = 1, 2$ , where  $h_i^k$  is a randomly (uniformly) chosen number from the interval  $[C_{\min} - I\alpha, C_{\max} + I\alpha]$ , where  $C_{\max} = \max\{c_i^1, c_i^2\}$ ,  $C_{\min} = \min\{c_i^1, c_i^2\}$ , and  $I = C_{\max} - C_{\min}$ . BLX- $\alpha$ - $\beta$  (Figure 4).<sup>48</sup> Let's suppose that  $C_1$  is the parent with best fitness. Then, two offspring are produced,  $H_k = (h_1^k, \dots, h_i^k, \dots, h_n^k)$ ,  $k = 1, 2$ , where  $h_i^k$  is a randomly (uniformly) chosen number from the interval  $[c_i^1 - I \cdot \alpha, c_i^2 + I \cdot \beta]$  if  $c_i^1 \leq c_i^2$ , or from  $[c_i^2 - I \cdot \beta, c_i^1 + I \cdot \alpha]$  otherwise.

Wright's heuristic crossover (Figure 5).<sup>36</sup> Let's assume that the fitness of  $C_1$  is better than the one of  $C_2$ . Then, two offspring are generated,  $H_k = (h_1^k, \dots, h_i^k, \dots, h_n^k)$ ,  $k = 1, 2$ , where  $h_i^k = u \cdot (c_i^1 - c_i^2) + c_i^1$  and  $u$  is a random number belonging to  $[0, 1]$ .

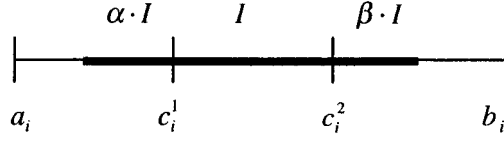
Linear BGA crossover (Figure 6).<sup>49</sup> Under the same consideration as mentioned previously,  $h_i^k = c_i^1 \pm r_i \cdot \gamma \cdot \Lambda$ , where the minus sign is chosen with a probability of 0.9,  $r_i = 0.5 \cdot (b_i - a_i)$ ,  $\gamma = \sum_{k=0}^{15} \alpha_k 2^{-k}$ , where  $\alpha_i \in \{0, 1\}$  is randomly generated with  $p(\alpha_i = 1) = 1/16$  and  $\Lambda = [(c_i^2 - c_i^1) / \|C_1 - C_2\|]$ .

Simulated binary crossover (Figure 7).<sup>4,50</sup> Two offspring are generated,  $H_k = (h_1^k, \dots, h_i^k, \dots, h_n^k)$ ,  $k = 1, 2$ , where

$$h_i^1 = \frac{1}{2} \cdot [(1 - \beta_k) \cdot c_i^1 + (1 + \beta_k) \cdot c_i^2] \quad \text{and} \quad h_i^2 = \frac{1}{2} \cdot [(1 + \beta_k) \cdot c_i^1 + (1 - \beta_k) \cdot c_i^2],$$



**Figure 3.** BLX- $\alpha$ .


 Figure 4. BLX- $\alpha$ - $\beta$ .

$\beta_k (\geq 0)$  is a sample from a random number generator having the density

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

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

$$\beta(u) = \begin{cases} (2u)^{1/(\eta+1)} & \text{if } u(0, 1) \leq \frac{1}{2} \\ [2(1-u)]^{1/(\eta+1)} & \text{if } u(0, 1) > \frac{1}{2} \end{cases}$$

Fuzzy recombination (FR) (Figure 8).<sup>51</sup> Two offspring are produced,  $H_k = (h_1^k, \dots, h_i^k, \dots, h_n^k)$ ,  $k = 1, 2$ . The probability that the  $i$ th gene in an offspring has the value  $\nu_i$  is given by the distribution  $p(\nu_i) \in \{\phi(c_i^1), \phi(c_i^2)\}$  where  $\phi(c_i^1)$  and  $\phi(c_i^2)$  are triangular probability distributions having the features shown in Figure 8 ( $c_i^1 \leq c_i^2$  is assumed).

Linear crossover (LX) (Figure 9):<sup>36</sup> Three offspring,  $H_k = (h_1^k, \dots, h_i^k, \dots, h_n^k)$ ,  $k = 1, 2, 3$ , are calculated:

$$h_i^1 = \frac{1}{2} c_i^1 + \frac{1}{2} c_i^2, \quad h_i^2 = \frac{3}{2} c_i^1 - \frac{1}{2} c_i^2, \quad \text{and} \quad h_i^3 = -\frac{1}{2} c_i^1 + \frac{3}{2} c_i^2$$

With this type of crossover an OSM is applied, which chooses the two most promising offspring of the three to substitute their parents in the population.

Max-Min-arithmetical crossover (MMAX) (Figure 10).<sup>35</sup>  $H_k = (h_1^k, \dots, h_i^k, \dots, h_n^k)$ , with  $k = 1, 2, 3, 4$ , are generated:

$$h_i^1 = \lambda c_i^1 + (1 - \lambda) c_i^2, \quad h_i^2 = \lambda c_i^2 + (1 - \lambda) c_i^1,$$

$$h_i^3 = \min\{c_i^1, c_i^2\}, \quad \text{and} \quad h_i^4 = \max\{c_i^1, c_i^2\}$$

The two best chromosomes are selected as final descendants for the new population.

Dynamic crossover (Figure 11).<sup>52</sup> Four offspring are generated because the Dubois  $\mathcal{F}$ -crossover,  $\mathcal{G}$ -crossover,  $\mathcal{M}^+$ -crossover, and  $\mathcal{M}^-$ -crossover were applied to them (see Appendix A). An OSM is used, which chooses the two most promising offspring of the four to substitute their parents in the population.

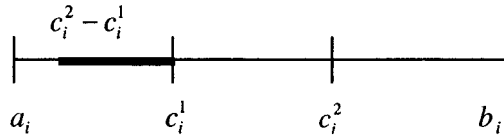
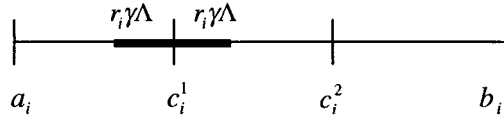


Figure 5. Wright's heuristic crossover.



**Figure 6.** Linear BGA crossover.

Dynamic heuristic crossover (DHX) (Figure 12).<sup>52</sup> Two offspring are generated: one with the dynamic dominated crossover and the other with the dynamic biased crossover (see Appendix B).

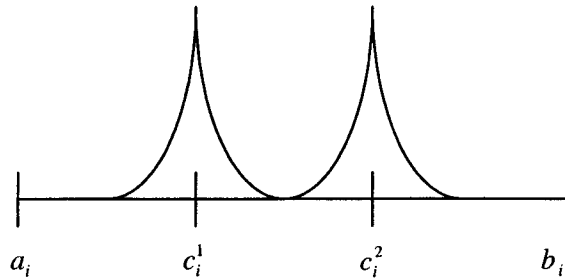
### 2.3. Exploration and Exploitation

Real-parameter crossover operators are able to produce exploration or exploitation (at different degrees) depending on the way in which they handle the current diversity of the population. They may either generate additional diversity starting from the current one (therefore exploration takes effect) or use this diversity for creating better elements (therefore exploitation comes into force). This is possible because of their self-adaptive features.<sup>28,53,54</sup>

The performance of an RCGA on a particular problem will be determined by the degrees of exploration and exploitation associated with the crossover operator being applied. Next, we introduce basic ideas about the availability of the crossover for adapting different exploration or exploitation degrees.

Let's consider  $c_i^1, c_i^2 \in [a_i, b_i]$  two genes to be combined with  $\alpha_i = \min\{c_i^1, c_i^2\}$  and  $\beta_i = \max\{c_i^1, c_i^2\}$ . The action interval  $[a_i, b_i]$  of these genes can be divided into three intervals:  $[a_i, \alpha_i]$ ,  $[\alpha_i, \beta_i]$ , and  $[\beta_i, b_i]$ . These intervals bind three regions to which the resultant genes of some combination of the former may belong. In addition, considering a region  $[\alpha'_i, \beta'_i]$  with  $\alpha'_i \leq \alpha_i$  and  $\beta'_i \geq \beta_i$  would seem reasonable (Figure 13).

These intervals may be classified as exploration or exploitation zones as is shown in Figure 13. The interval with both genes being the extremes is an exploitation zone because any gene  $g_i$  generated by a crossover in this interval fulfills  $\max\{|g_i - \alpha_i|, |g_i - \beta_i|\} \leq |\alpha_i - \beta_i|$ . The two intervals that remain on both sides are exploration zones because this property is not fulfilled. The region



**Figure 7.** Simulated binary crossover.

Prob. Dist.	Minimum Value	Modal Value	Maximum Value
$\phi(c_i^1)$	$c_i^1 - d \cdot  c_i^2 - c_i^1 $	$c_i^1$	$c_i^1 + d \cdot  c_i^2 - c_i^1 $
$\phi(c_i^2)$	$c_i^2 - d \cdot  c_i^2 - c_i^1 $	$c_i^2$	$c_i^2 + d \cdot  c_i^2 - c_i^1 $

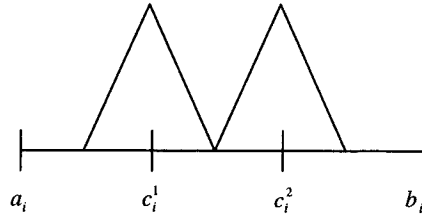


Figure 8. FR.

with extremes  $\alpha'_i$  and  $\beta'_i$  could be considered as a relaxed exploitation zone. Therefore, exploration and/or exploitation degrees may be assigned to any crossover operator for RCGAs regarding the way in which these intervals are considered for generating genes.

The arithmetical crossover with  $\lambda = 0.5$  is a clear example of an exploitative crossover operator. Contrarily, this operator will show exploration for  $\lambda > 1$  or  $\lambda < 0$ . An example of crossover showing relaxed exploitation is BLX- $\alpha$ . In the absence of selection pressure, all values of  $\alpha < 0$  will show a tendency for the population to converge toward values in the center of their ranges, producing low diversity levels in the population and inducing a possible premature convergence toward nonoptimal solutions. Only when  $\alpha < 0.5$  is a balanced relationship between the convergence (exploitation) and divergence (exploration) reached, because the probability that an offspring will lie outside its parents becomes equal to the probability that it will lie between its parents.

## 2.4. Principles for the Design of the Crossover

In Refs. 28 and 55, the following guidelines for designing crossovers for RCGAs are proposed:

Guideline 1. The crossover operator should preserve the statistics of the population such as mean vector and the variance-covariance matrix.

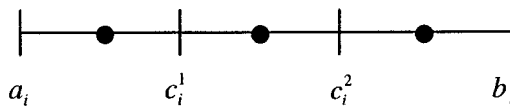


Figure 9. LX.

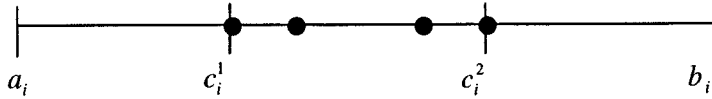


Figure 10. MMAX.

Guideline 2. Crossover operators should generate offspring having as much diversity as possible under constraint of Guideline 1.

Guideline 3. Guideline 1 is a guideline when the selection operator works ideally. However, it may fail to suggest a good region to be searched by the population. To make the search robust, children should be distributed more widely than in Guideline 1. It should be noted that the Guideline 1 gives a reference point, and there exists a trade-off between efficiency and robustness in adopting this guideline.

In Ref. 53, two properties that the crossover should have for successful applications in real-valued search spaces are postulated (which agree with Guidelines 1 and 2):

- (1) The crossover operator must produce a children population that has the same mean as that in the parent population.
- (2) The variance of the resulting children population may be larger than that of the parent population.

Guideline 2 helps to maintain a balance between the spread of solutions under selection and crossover operators. Because selection emphasizes good solutions by eliminating bad solutions in a population, it may, generally, reduce the diversity of the population. If the crossover operator also has a tendency to reduce the diversity of the population, the overall search algorithm may not have adequate power to adapt to any function landscape.<sup>54</sup> This guideline reveals the importance of considering the exploration and relaxed exploitation intervals for designing crossover operators for RCGAs. This is also shown in Ref. 5, which is an empirical study comparing several crossovers.

Finally, there are a number of theoretical studies involved with trying to understand the properties of different real-parameter crossover operators.<sup>53,56–62</sup>

### 3. TAXONOMY FOR THE CROSSOVER OPERATOR FOR RCGAs

In this section, we propose a taxonomy that classifies the crossover operators for the RCGAs presented in Section 2.2. in different groups, focusing on the

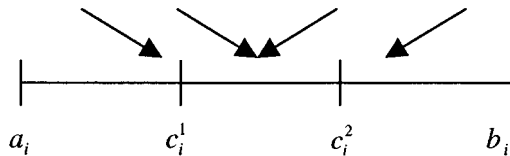


Figure 11. Dynamic crossover.



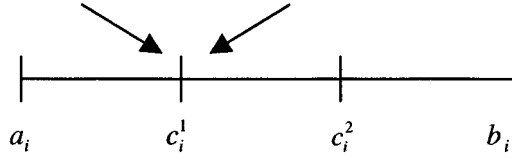


Figure 12. DHX.

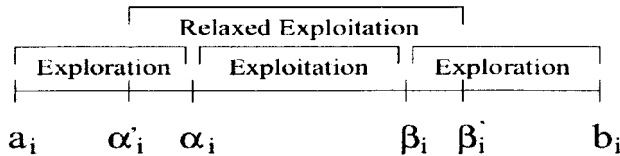
features associated with the OGMs that are applied on the parents to obtain the offspring. This taxonomy may be useful to discover the features that allow the crossover operator to have a positive influence on performance. They include whether they preserve the genes of the parents in the offspring, whether the genes of the offspring are obtained from an aggregation function in which its arguments are the genes of the parents, or whether the genes in the offspring are generated from a probability distribution defined on the neighborhoods of the genes of the parents.

The taxonomy presented is composed of the following four groups:

**Discrete crossover operators (DCOs).** This category groups all the crossover operators presented for binary coding, which are directly applicable to real coding. It includes the simple, two-point and uniform crossover operators. With these crossovers, the value of each gene in the offspring coincides with the value of this gene in one of the parents ( $h_i \in \{c_i^1, c_i^2\}$ ), i.e., the values of the genes in the parents are not transformed numerically for obtaining the values of the genes in the offspring. Geometrically, DCOs generate a corner of the hypercube defined by the component of the two parents. The effect of these operators, according to the intervals of the generation of genes, is shown in Figure 14.

**Aggregation-based crossover operators (ABCOs).** The ABCOs include operators that use an aggregation function that numerically combines the values of the genes of the parents to generate the value of the genes of the offspring. If  $[a_i, b_i]$  is the action interval for the  $i$ th gene, an aggregation function  $f_i : [a_i, b_i] \rightarrow [a'_i, b'_i]$  ( $[a'_i, b'_i] \subseteq [a_i, b_i]$ ) should be provided. Then, the value for the  $i$ th gene of the offspring is computed as  $f_i(c_i^1, c_i^2)$ . The arithmetical, geometrical, and LX operators are representatives of ABCOs. In the case of the arithmetical and LX, the aggregation functions are linear combinations of  $c_i^1$  and  $c_i^2$ . The ABCOs act as shown in Figure 15, which indicates that the ABCOs may generate genes in the exploitation interval or in the exploration interval.

**Neighborhood-based crossover operators (NBCOs).** This group includes crossovers that determine the genes of the offspring extracting values from intervals defined by neighborhoods associated with the genes of the parents throughout probability distributions. Examples of NBCOs are BLX- $\alpha$ , simulated binary crossover, and FR, which are based on uniform, exponential, and triangular probability distributions, respectively. Figure 16

Figure 13. Action interval for  $c_i^1$  and  $c_i^2$ .



**Figure 14.** Effects of the DCOs.

shows the neighborhoods considered by NBCOs. In general, this type of operator generates genes in relaxed exploitation intervals.

Hybrid crossover operators (HCOs). These crossover operators produce offspring by applying OGMs that belong to the different categories. The MMAX is an HCO, because it creates two offspring by means of DCOs (the maximum and minimum) and two by an ABCO (the arithmetical). This operator allows different levels of exploitation and exploration to be introduced simultaneously in the search process.

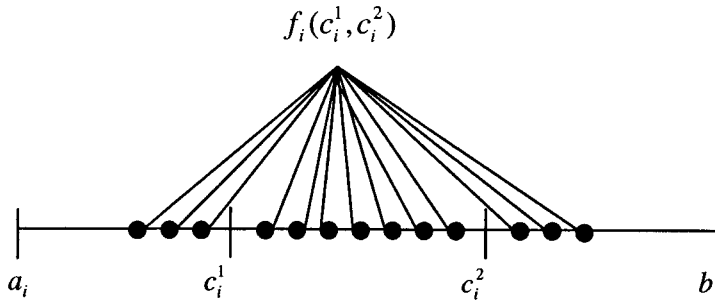
Table I shows the classification of the crossover operators described in Section 2.2, according to the taxonomy presented.

Finally, both ABCOs and NBCOs dispose of heuristic crossover operators, which take into account the goodness of the parents for generating the offspring, with the objective of leading the search process toward the most promising zones, e.g., DHX (ABCOs), Wright's heuristic crossover, linear BGA crossover, and BLX- $\alpha$ - $\beta$  (NBCOs).

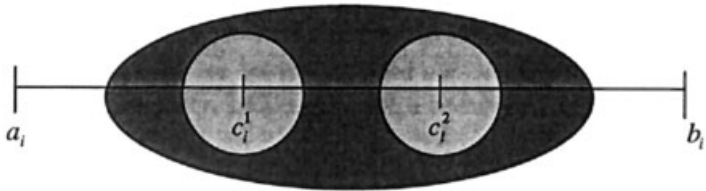
#### 4. EXPERIMENTS

Minimization experiments on the test suite (described in Appendix C) were performed with the aim of determining if differences on performance exist between crossover operators that belong to the categories of the taxonomy proposed. If this occurs, then, we will be able to discover specific features that allow the crossover operator to have a positive influence on RCGA behavior.

We have considered a generational RCGA model that applies the nonuniform mutation operator.<sup>3</sup> This operator has been widely used, reporting good results.<sup>5</sup> The selection probability calculation follows linear ranking<sup>63</sup> ( $\eta_{\min} = 0.75$ ) and the sampling algorithm is the stochastic universal sampling.<sup>64</sup> The elitist strategy<sup>65</sup> is



**Figure 15.** Possible gene values computed by ABCOs from  $c_i^1$  and  $c_i^2$ .



**Figure 16.** Neighborhoods taken into account by NBCOs.

considered as well, which involves making sure that the best-performing chromosome always survives intact from one generation to the next.

We have implemented different RCGAs that are distinguished regarding the crossover operator applied. They are shown in Table II. The population size is 61 individuals, the probability of updating a chromosome by mutation is 0.125, and the crossover probability is 0.6. We executed all the algorithms 30 times, each one with a maximum of 100,000 evaluations.

4.1. Results

Tables DI–DVII in Appendix D show our results. The performance measures used are the following:

- *A* performance: average of the best-fitness function found at the end of each run.
- *SD* performance: standard deviation.
- *B* performance: best of the fitness values averaged as *A* performance.

The *T1* and *T2* columns in these tables show the result of the *t*-test (at 0.05 level of significance) as follows:

**Table I.** Classification of the crossover operators for RCGAs.

Crossover	DCOs	ABCOs	NBCOs	HCOs
Simple	X			
Two-point	X			
Uniform	X			
Arithmetical		X		
Geometric		X		
Linear		X		
Wright’s heuristic		X		
Linear BGA		X		
Dynamic		X		
Dynamic heuristic		X		
BLX- $\alpha$			X	
BLX- $\alpha$ - $\beta$			X	
Simulated binary			X	
FR			X	
MMAX				X

**Table II.** RCGA instances implemented.

RCGAs	Crossover operator
SX	Simple crossover
AX	Arithmetical crossover ( $\lambda = 0.25$ )
GX	Geometrical crossover ( $\omega = 0.25$ )
BLX-0	BLX- $\alpha$ ( $\alpha = 0.0$ )
BLX-0.3	BLX- $\alpha$ ( $\alpha = 0.3$ )
BLX-0.5	BLX- $\alpha$ ( $\alpha = 0.5$ )
SBX-2	Simulated binary crossover ( $\eta = 2$ )
SBX-5	Simulated binary crossover ( $\eta = 5$ )
FR	Fuzzy recombination ( $d = 0.5$ )
BLX-0.5-0	BLX- $\alpha$ - $\beta$ ( $\alpha = 0.5$ and $\beta = 0$ )
WHX	Wright's heuristic crossover
2PX	Two-point crossover
UX	Uniform crossover
BGAX	Linear BGA crossover
DHX	Dynamic heuristic
MMAX	Max-min-arithmetical crossover ( $\lambda = 0.25$ )
DX	Dynamic crossover
LX	Linear crossover

- T1 is introduced to ascertain if differences in the  $A$  performance for the best crossover operator are significant when compared with the one for the other crossovers in the respective table. Only crossovers that generate two offspring are considered. Therefore, this column is empty for MMAX, DX, and LX.
- In T2, the same comparison is made, but taking into account all the crossover operators.

In these columns, the crossover with the best  $A$  performance value is marked with \*\*, and the direction of any significant differences is denoted either by a plus sign (+) for an improvement in  $A$  performance or an approximate sign ( $\cong$ ) for nonsignificant differences.

## 4.2. Analysis of the Results

We performed the analysis of the results from a double perspective. First, in Section 4.2.1, we examine the results for the crossover operators that produce two offspring. Then, in Section 4.2.2, we study the crossover operators with multiple descendents, which generate more than two offspring and select the two best offspring to replace the parents in the population (MMAX, DX, and LX). Finally, in Section 4.2.3, we make conclusions from the results for all the groups of the taxonomy.

### 4.2.1. Crossover Operators with Two Offspring

Table III shows the percentages in which each crossover operator with two offspring has obtained the best  $A$  performance on all test functions. Their columns have the following information:

**Table III.** Analysis for the crossover operators with two offspring.

Crossover	Best average/ best $t$ -test	Best average/ similar $t$ -test	Total best	Similar $t$ -test/ no best average	Total best/ similar
DHX	23.07%	15.38%	38.45%	15.38%	53.83%
FR	0%	23.07%	23.07%	30.76%	53.83%
BLX-0.3	0%	7.69%	7.69%	38.45%	46.14%
SBX2	0%	7.69%	7.69%	30.76%	38.45%
SBX5	0%	7.69%	7.69%	23.07%	30.76%
AX	0%	7.69%	7.69%	15.38%	23.07%
UX	0%	7.69%	7.69%	7.69%	15.38%
BLX-0	0%	0%	0%	23.07%	23.07%
BLX-0.5	0%	0%	0%	23.07%	23.07%
2PX	0%	0%	0%	15.38%	15.38%
GX	0%	0%	0%	15.38%	15.38%
SX	0%	0%	0%	15.38%	15.38%
BGAX	0%	0%	0%	7.69%	7.69%
WHX	0%	0%	0%	0%	0%

- Best average/best  $t$ -test. Percentage of test functions in which the crossover has obtained the best  $A$  performance (\*\* in T1 column) and the application of the  $t$ -test confirms that it is the best [plus sign (+) in the T1 column associated with the other crossovers].
- Best average/similar  $t$ -test. This column shows the same information as the previous column, but there is a different crossover with no differences in  $A$  performance, according to the  $t$ -test (there is another crossover with a  $\equiv$  sign in the T1 column).
- Total best. Percentage of test functions in which the crossover operator achieves the best  $A$  performance, without considering the  $t$ -test. This percentage is calculated as the sum of the previous two columns.
- Similar  $t$ -test/no best average. Percentage of test functions in which the crossover operator shows, after the application of the  $t$ -test, nonsignificant differences in  $A$  performance regarding the best value ( $\equiv$  in the T1 column).
- Total best/similar. Percentage of test functions in which the crossover operator obtains either the best  $A$  behavior or one similar to the best. This percentage is the result of the sum of the two previous columns.

The best crossovers are the DHX, FR, BLX- $\alpha$ , and SBX:

- DHX produces two offspring around the best parent, one using exploration intervals and the another by means of the exploitation interval. In addition, it provides diversity levels that decrease with time, introducing a heuristic local tuning that becomes effective for RCGA performance.<sup>52</sup>
- The common feature of FR, BLX- $\alpha$ , and SBX is that they are NBCOs that build the offspring through relaxed exploitation intervals. This confirms the importance of considering this type of interval for designing crossover operators for RCGAs<sup>5,47,66</sup> (Section 2.4).

Now, we make some conclusions about the performance of the groups of the taxonomy that include crossovers with two offspring. To do this, we have introduced Table IV, which refers to the categories of the taxonomy.

The following most profitable groups are ABCOs and NBCOs:

**Table IV.** Analysis for the groups of the taxonomy that include crossovers with two offspring.

Taxonomy groups	Best average/ best $t$ -test	Best average/ similar $t$ -test	Total best	Similar $t$ -test/ no best average	Total best/ similar
ABCOs	23.07%	23.07%	46.14%	30.76%	76.9%
NBCOs	23.07%	23.07%	46.14%	30.76%	76.9%
DCOs	0%	7.69%	7.69%	7.69%	15.38%

- ABCOs achieve the best  $A$  performance for six test functions: Sphere (Table DI), Rastrigin (Table DII), Rosenbrock (Table DIV), polynomial fitting problem (Table DIV), Ackley (Table DV), and Bohachevsky (Table DVI). In addition, after the application of the  $t$ -test, three ABCO instances are the best, whereas the other three are similar to crossover operators of the NBCO category.
- For six test functions, the best  $A$  results are returned by NBCOs: Schwefel (Table DI), Griewangk (Table DII), E-F10 (Table DIII), system of linear equations (Table DIII), FMS parameter identification (Table DV), and Colville (Table DVII). Again, the  $t$ -test confirms that three NBCO instances are the best and that the other three present similarities with ABCOs.

DCOs outperform the other categories only on the Watson's function (Table DVI); however, the  $t$ -test reveals that their results are similar to the ones for crossover operators in other categories.

#### 4.1.2. Crossovers with Two Offspring vs. Crossovers with Multiple Descendents

This section compares the performance of the crossover operators with two offspring with one of the crossover operators with multiple descendents (MMAX, DX, and LX). Table V contains the same information as Table III for the crossover operators that have obtained the best  $A$  performance for each one of the 13 test functions, considering all the crossovers (Tables DI–DVII, column T2).

Two important observations are shown in Table V:

**Table V.** Results analysis of the best crossover operators.

Crossover	Best average/ best $t$ -test	Best average/ similar $t$ -test	Total best	Similar $t$ -test/ no best average	Total best/ similar
DHX	23.07%	15.38%	38.45%	7.69%	46.14%
FR	15.38%	0%	15.38%	7.69%	23.07%
SBX-2	0%	0%	0%	7.69%	7.69%
SBX-5	7.69%	0%	7.69%	15.38%	23.07%
MMAX	0%	7.69%	7.69%	15.38%	23.07%
DX	0%	0%	0%	7.69%	7.69%
LX	23.07%	7.69%	30.76%	7.69%	38.45%

**Table VI.** Analysis for the groups of the taxonomy.

Taxonomy groups	Best average/ best $t$ -test	Best average/ similar $t$ -test	Total best	Similar $t$ -test/ no best average	Total best/ similar
DCOs	0%	0%	0%	7.69%	7.69%
ABCOs	46.14%	23.07%	69.21%	15.38%	84.59%
NBCOs	15.38%	7.69%	23.07%	30.76%	53.83%
HCOs	0%	7.69%	7.69%	15.38%	23.07%

- Again, the DHX operator is the best operator.
- Two crossovers with multiple descendents, LX and MMAX achieved a significant performance. Therefore, the generation of more than two offspring and the selection of the two best seem to be a promising way to improve the behavior of the crossover operator for RCGAs.

Finally, we show that all of these operators are ABCOs and generate offspring with both exploration and exploitation features.

#### 4.1.3. *Essential Points of the Taxonomy*

Table VI analyzes the performance associated with the four groups of the taxonomy, considering the results for all the crossover operators:

- The ABCO group embraces the majority of crossover operators that provide solutions with high quality. In particular, for 84.59% of the test functions, they allow the best A performance to be reached.
- The crossover operators that exploit the numerical nature of the real coding (ABCOs and NBCOs) consistently outperform the classical DCOs. The poor behavior shown by DCOs may be explained by the claim made in Ref. 4 about the application of this type of operator to RCGAs in which Deb et al. state, “they do not have an adequate search power and thus the search within a decision variable has to mainly rely on the mutation operator.”
- Finally, we may underline the remarkable results given by the HCO group, which is composed of one crossover only, the MMAX. In this way, the hybridization of crossovers that belong to different categories of the taxonomy arises as a promising technique to design powerful crossover operators for RCGAs.

## 5. CONCLUSIONS

This study presented a taxonomy for the crossover operator for RCGAs, which is an effective tool to reveal the features of this operator that allows RCGA performance to be enhanced.

An empirical study of different crossover instances presented in the RCGA literature has been made. The following principal conclusions were reached from an analysis of the results:

- The category of the taxonomy that groups the majority of crossovers with the best performance is the ABCOs. NBCOs present a significant behavior as well. These types

of operators exploit the numerical nature of real coding. However, the classical DCOs appear useless for RCGAs.

- The most promising crossover operator is the DHX. It combines both the exploration and the exploitation of the neighborhood associated with the best parent. Furthermore, their degrees vary during the run, with the goal of producing a heuristic local tuning.
- Three mechanisms are essential to design powerful crossover operators:
  - (1) Use of probability distributions defined on relaxed exploitation intervals to generate the genes of the offspring (NBCOs).
  - (2) Generation of more than two offspring and the selection of the two best (crossover operators with multiple descendents).
  - (3) Creation of offspring using crossovers that belong to different categories of the taxonomy (hybrid crossovers).

In conclusion, additional research is necessary in some areas of this study, including the design of hybrid crossovers that combine ABCOs and NBCOs and the study of NBCOs with multiple descendents based on relaxed exploitation intervals (initial work is found in Ref. 67). Furthermore, the taxonomy may be extended with the aim of classifying crossover operators that are applicable to more than two parents: UNDX,<sup>68</sup> BNDX,<sup>69</sup> TMX,<sup>70</sup> UNDX-m,<sup>37</sup> SPX,<sup>39</sup> EDX,<sup>71</sup> PCX<sup>34</sup> and crossover based on confidence intervals.<sup>38</sup> Finally, another open issue concerns the use of different aggregation functions or probability distributions for generating the genes of an offspring.<sup>72</sup>

## References

1. [Goldberg DE. Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley; 1989.](#)
2. [Holland JH. Adaptation in Natural and Artificial Systems. London: The MIT Press; 1992.](#)
3. [Michalewicz Z. Genetic Algorithms + Data Structures = Evolutionary Programs. Berlin: Springer-Verlag; 1992.](#)
4. [Deb K. Multi-objective optimization using evolutionary algorithms. Chichester: Wiley; 2001.](#)
5. [Herrera F, Lozano M, Verdegay JL. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. Artif Intell Rev 1998;12\(4\):265–319.](#)
6. [Lucasius CB, Kateman G. Applications of genetic algorithms in chemometrics. In: Schaffer JD, editor. Proc of the 3rd Int Conf on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann Publishers; 1989. pp 170–176.](#)
7. [Blanco A, Delgado M, Pegalajar MC. A real-coded genetic algorithm for training recurrent neural networks. Neural Netw 2001;14\(1\):93–105.](#)
8. [Zamparelli M. Genetically trained cellular neural networks. Neural Networks 1997;10\(6\):1143–1151.](#)
9. [Hajela P. Soft computing in multidisciplinary aerospace design—new directions for research. Prog Aerospace Sci 2002;38\(1\):1–21.](#)
10. [Lin C-L, Lai R-M. Parameter design for a guidance and control system using genetic approach. Aerospace Sci Technol 2001;5\(6\):425–434.](#)
11. [Oyama A, Obayashi S, Nakamura T. Real-coded adaptive range genetic algorithm applied to transonic wing optimization. Appl Soft Comput 2001;1\(3\):179–187.](#)
12. [Périaux J, Sefioui M, Stoufflet B, Mantel B, Laporte E. Robust genetic algorithms for optimization problems in aerodynamic design. In: Genetic Algorithms in Engineering and Computer Science. Chichester: John Wiley and Sons; 1995. pp 370–396.](#)
13. [Roubos JA, van Straten G, van Boxtel AJB. An evolutionary strategy for fed-batch bioreactor optimization; concepts and performance. J Biotechnol 1999;67\(2–3\):173–187.](#)



14. Arfiadi Y, Hadi MNS. Optimal direct (static) output feedback controller using real coded genetic algorithms. *Comp Struct* 2001;79(17):1625–1634.
15. Huang P-Y, Chen Y-Y. Design of PID controller for precision positioning table using genetic algorithms. In: *Proc of the 36th IEEE Conf on Decision and Control*. 1997. Piscataway, NJ: IEEE Press; pp 2513–2514.
16. Kawabe T, Tagami T. A real coded genetic algorithm for matrix inequality design approach of robust PID controller with two degrees of freedom. In: *Proc of the 1997 IEEE Int Symp on Intelligent Control*. 1997. Piscataway, NJ: IEEE Press; pp 119–124.
17. Duffy J, McNelis PD. Approximating and simulating the stochastic growth model: Parameterized expectations, neural networks, and the genetic algorithm. *J Econ Dyn Control* 2001;25(9):1273–1303.
18. Harris SP, Ifeachor EC. Automatic design of frequency sampling filters by hybrid genetic algorithm techniques. *IEEE Trans Sig Process* 1998;46(12):3304–3314.
19. Caorsi S, Massa A, Pastorino M. A computational technique based on a real-coded genetic algorithm for microwave imaging purposes. *IEEE Trans Geosci Remote Sens* 2000;38(4):1697–1708.
20. Lu Y, Cai X, Gao Z. Optimal design of special corner reflector antennas by the real-coded genetic algorithm. In: *Proc Microwave Conf 2000 Asia-Pacific*. 2000. pp 1457–1460.
21. Qing A, Lee CK. Microwave imaging of a perfectly conducting cylinder using a real-coded genetic algorithm. *Microwaves, Antennas Propagation, IEE Proc* 1999;146(6):421–425.
22. Shi KL, Chan TF, Wong YK, Ho SL. Speed estimation of an induction motor drive using an optimized extended Kalman filter. *IEEE Trans Indust Electron* 2002;49(1):124–133.
23. Azariadis PN, Nearchou AC, Aspragathos NA. An evolutionary algorithm for generating planar developments of arbitrarily curved surfaces. *Comp Industry* 2002;47(3):357–368.
24. Thomas GM, Gerth R, Velasco T, Rabelo LC. Using real-coded genetic algorithms for Weibull parameter estimation. *Comput Indust Eng* 1995;29(1–4):377–381.
25. Turcanu C-O, Craciunescu T. A genetic approach to limited data tomographic reconstruction of time-resolved energy spectrum of short-pulsed neutron sources. *Pattern Recognition Lett* 2002;23(8):967–976.
26. Chang F-J. Real-coded genetic algorithm for rule-based flood control reservoir management. *Water Resource Manag* 1998;12(3):185–198.
27. De Jong KA, Spears WM. A formal analysis of the role of multi-point crossover in genetic algorithms. *Ann Math Artif Intell* 1992;5(1):1–26.
28. Kita H. A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms. *Evol Comput J* 2001;9(2):223–241.
29. Lin G, Yao X. Analysing crossover operators by search step size. In: *Proc of 1997 Int Conf on Evolutionary Computation*. Piscataway, NJ: IEEE Press; 1997. pp 13–16.
30. Liepins GE, Vose MD. Characterizing crossover in genetic algorithms. *Ann Math Artif Intell* 1992;5(1):27–34.
31. Fernandes C, Rosa A. 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*. Piscataway, NJ: IEEE Press; 2001. pp 60–66.
32. Huang C-F. An analysis of mate selection in genetic algorithms. In: *Proc of the Genetic and Evolutionary Computation Conf (GECCO-2001)*. San Mateo, CA: Morgan Kaufmann; 2001. p 766.
33. Ronald E. When selection meets seduction. In: Forrest S, editor. *Proc 5th Int Conf on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann; 1993. pp 167–173.
34. Deb K, Joshi D, Anand A. Real coded evolutionary algorithms with parent-centric recombination. In: *Proc of the 2002 Congress on Evolutionary Computation*. Piscataway, NJ: IEEE Press; 2002. pp 61–66.

35. [Herrera F, Lozano M, Verdegay JL. Fuzzy connectives based crossover operators to model genetic algorithms population diversity. \*Fuzzy Set Syst\* 1997;92\(1\):21–30.](#)
36. [Wright A. Genetic algorithms for real parameter optimization. In: Rawlin GJE, editor. \*Foundations of Genetic Algorithms 1\*. San Mateo, CA: Morgan Kaufmann; 1991. pp 205–218.](#)
37. [Kita H, Kobayashi S. Multi-parental extension of the unimodal normal distribution crossover for real-coded genetic algorithms. In: \*Proc of the Int Conf on Evolutionary Computation'99\*. Piscataway, NJ: IEEE Press; 1999. pp 646–651.](#)
38. [Ortiz D, Hervás C, Muñoz J. Genetic algorithm with crossover based on confidence intervals as an alternative to least squares estimation for nonlinear models. In: \*4th Metaheuristics Int Conf\* 2001. pp 343–347.](#)
39. [Tsutsui S, Yamamura M, Higuchi T. Multi-parent recombination with simplex crossover in real-coded genetic algorithm. In: \*Proc of the Genetic and Evolutionary Computation Conf \(GECCO-99\)\*. San Mateo, CA: Morgan Kaufmann; 1999. pp 657–664.](#)
40. [Esquivel S, Leiva A, Gallard R. Multiple crossover per couple in genetic algorithms. In: \*Proc of the 4th IEEE Int Conf on Evolutionary Computation \(ICEC'97\)\*. Piscataway, NJ: IEEE Press; 1997. pp 103–106.](#)
41. [Esquivel S, Ferrero S, Gallard R, Salto C, Alfonso H, Schütz M. Enhanced evolutionary algorithms for single and multiobjective optimization in the job shop scheduling problem. \*J Knowledge Based Systems\* 2002;15\(1–2\):12–25.](#)
42. [Satoh H, Yamamura M, Kobayashi S. Minimal generation gap model for GAs considering both exploration and exploitation. In: \*Proc Methodologies for the Conception, Design, and Application of Intelligent Systems \(IIZUKA'96\)\*. 1996. pp 494–497.](#)
43. [Eshelman LJ, Caruana A, Schaffer JD. Biases in the crossover landscape. In: Schaffer JD, editor. \*Proc of the Third Int Conf on Genetic Algorithms\*. San Mateo, CA: Morgan Kaufmann Publishers; 1989. pp 86–91.](#)
44. [Syswerda G. Uniform crossover in genetic algorithms. In: Schaffer JD, editor. \*Proc 3rd Int Conf on Genetic Algorithm\*. San Mateo, CA: Morgan Kaufmann Publishers; 1989. pp 2–9.](#)
45. [Michalewicz Z, Nazhiyath G, Michalewicz M. A note on usefulness of geometrical crossover for numerical optimization problems. In: \*Proc of the 5th Ann Conf on Evolutionary Programming\*. Cambridge, MA: MIT Press; 1996. pp 305–312.](#)
46. [Bremermann HJ, Rogson M, Salaff S. Global properties of evolution processes. In: Pattee HH, Edlsack EA, Frin L, Callahan AB, editors. \*Natural Automata and Useful Simulations\*. Washington, DC: Spartan; 1966. pp 3–41.](#)
47. [Eshelman LJ, Schaffer JD. Real-coded genetic algorithms and interval-schemata. In: Whitley LD, editor. \*Foundations of Genetic Algorithms 2\*. San Mateo, CA: Morgan Kaufmann Publishers; 1993. pp 187–202.](#)
48. [Eshelman LJ, Mathias KE, Schaffer JD. Crossover operator biases: Exploiting the population distribution. In: Bäck T, editor. \*Proc of the 7th Int Conf on Genetic Algorithms\*. San Mateo, CA: Morgan Kaufmann; 1997. pp 354–361.](#)
49. [Schlierkamp-Voosen D. Strategy adaptation by competition. In: \*Proc 2nd Eur Congress on Intelligent Techniques and Soft Computing\*. 1994. pp 1270–1274.](#)
50. [Deb K, Agrawal RB. Simulated binary crossover for continuous search space. \*Complex Syst\* 1995;9:115–148.](#)
51. [Voigt HM, Mühlenbein H, Cvetkovic D. Fuzzy recombination for the breeder genetic algorithm. In: Eshelman L, editor. \*Proc 6th Int Conf on Genetic Algorithms\*. San Mateo, CA: Morgan Kaufmann Publishers; 1995. pp 104–111.](#)
52. [Herrera F, Lozano M, Verdegay JL. Dynamic and heuristic fuzzy connectives based crossover operators for controlling the diversity and convergence of real-coded genetic algorithms. \*Int J Intell Syst\* 1996;11:1013–1041.](#)

53. Beyer H-G, Deb K. On self-adaptive features in real parameter evolutionary algorithms. [IEEE Trans Evolut Comp 2001;5\(3\):250–270.](#)
54. Deb K, Beyer H. Self-adaptive genetic algorithms with simulated binary crossover. [Evol Comput J 2001;9\(2\):195–219.](#)
55. Kita H, Yamamura M. A functional specialization hypothesis for designing genetic algorithms. In: [Proc of the 1999 IEEE Int Conf on Systems, Man, and Cybernetics. 1999. Piscataway, NJ: IEEE Press; pp 579–584.](#)
56. Fogel DB, Beyer H-G. A note on empirical evaluation of intermediate recombination. [Evol Comput J 1995;3\(4\):491–495.](#)
57. Higuchi T, Tsutsui S, Yamamura M. Theoretic analysis of simplex crossover for real-coded genetic algorithms. In: [Parallel Problem Solving from Nature VI. Berlin: Springer; 2000. pp 365–374.](#)
58. Kita H, Ono I, Kobayashi S. Theoretical analysis of the unimodal normal distribution crossover for real-coded genetic algorithms. In: [Proc of the Int Conf on Evolutionary Computation'98. Piscataway, NJ: IEEE Press; 1998. pp 529–534.](#)
59. Nomura T. An analysis on linear crossover for real number chromosomes in an infinite population size. In: [Proc of the Int Conf on Evolutionary Computation \(ICEC'97\). Piscataway, NJ: IEEE Press; 1997. pp 111–114.](#)
60. Nomura T, Shimohara K. An analysis of two-parent recombinations for real-valued chromosomes in an infinite population. [Evol Comput J 2001;9\(3\):283–308.](#)
61. Nomura T. Behavior analysis of real-valued evolutionary algorithms for independent loci in an infinite population. In: [Proc of the Genetic and Evolutionary Computation Conf \(GECCO-2001\). San Mateo, CA: Morgan Kaufmann; 2001. p 774.](#)
62. Qi X, Palmieri F. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space. Part II: Analysis of the diversification role of crossover. [IEEE Trans Neural Netw 1994;5\(1\):120–128.](#)
63. Baker JE. Adaptive selection methods for genetic algorithms. In: [Grefenstette JJ, editors. Proc of the 1st Int Conf on Genetic Algorithms and their Applications. Hillsdale, MA: L. Erlbaum Associates; 1987. pp 14–21.](#)
64. Baker JE. Reducing bias and inefficiency in the selection algorithm. In: [Grefenstette JJ, editor. Proc of the Second Int Conf on Genetic Algorithms and their Applications. Hillsdale, NJ: Lawrence Erlbaum; 1987. pp 14–21.](#)
65. De Jong KA. An analysis of the behavior of a class of genetic adaptive systems. PhD dissertation, University of Michigan, 1975.
66. Tsutsui S, Goldberg DE. Search space boundary extension method in real-coded genetic algorithms. [Inform Sci 2001;133:229–247.](#)
67. Herrera F, Lozano M, Pérez E, Sánchez AM, Villar P. Multiple crossover per couple with selection of the two best offspring: An experimental study with the BLX- $\alpha$  crossover operator for real-coded genetic algorithms. In: [Jarifo FJ, Riquelme JC, Toro M, editors. Advances in Artificial Intelligence—IBERAMIA 2002, LNAI 2527. Berlin: Springer; 2002. pp 392–401.](#)
68. Ono I, Kobayashi S. A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover. In: [Bäck T, editor. Proc of the 7th Int Conf on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann; 1997. pp 246–253.](#)
69. Ono I, Yamamura M, Kobayashi S. A genetic algorithm with characteristic preservation for function optimization. In: [Proc Methodologies for the Conception, Design and Application of Intelligent Systems \(IIZUKA'96\). 1996. pp 511–514.](#)
70. Someya H, Yamamura M. Genetic algorithm with search area adaptation for the function optimization and its experimental analysis. In: [Proc 2001 Congress on Evolutionary Computation. Piscataway, NJ: IEEE Press; 2001. pp 933–940.](#)

71. Sakuma J, Kobayashi S. Extrapolation-directed crossover for real-coded GA: Overcoming deceptive phenomena by extrapolative search. In: *Proc 2001 Congress on Evolutionary Computation*. Piscataway, NJ: IEEE Press; 2001. pp 655–662.
72. Nomura T, Miyoshi T. Numerical coding and unfair average crossover in GA for fuzzy rule extraction in dynamic environments. In: Furuhashi T, Uchikawa Y, editors. *Fuzzy Logic, Neural Networks, and Evolutionary Computation, Lecture Notes in Artificial Intelligence 1152*. Berlin: Springer; 1996. pp 55–72.
73. Schwefel H-P. *Numerical Optimization of Computer Models*. Chichester: Wiley; 1981.
74. Törn A, Antanas Z. *Global Optimization. Lecture Notes in Computer Science 350*. Berlin: Springer; 1989.
75. Griewangk AO. Generalized descent of global optimization. *J Optimiz Theor Applicat* 1981;34:11–39.
76. Whitley D, Beveridge R, Graves C, Mathias K. Test driving three 1995 genetic algorithms: New test functions and geometric matching. *J Heuristic* 1995;1:77–104.
77. Eshelman LJ, Mathias KE, Schaffer JD. Convergence controlled variation. In: Belew R, Vose M, editors. *Foundations of genetic algorithms 4*. San Mateo, CA: Morgan Kaufmann Publishers; 1997. pp 203–224.
78. Tsutsui S, Fujimoto Y. Forking genetic algorithm with blocking and shrinking modes. In: Forrest S, editor. *Proc 5th Int Conf on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann Publishers; 1993. pp 206–213.
79. Storn R, Price K. Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012. Berkeley, CA: International Computer Science Institute; 1995.
80. Ackley DH. *A connectionist machine for genetic hillclimbing*. Boston: Kluwer Academic Publishers; 1987.
81. Reynolds RC, Chung C. Knowledge-based self-adaptation in evolutionary programming using cultural algorithms. In: *Proc of 1997 Int Conf on Evolutionary Computation*. Piscataway, NJ: IEEE Press; 1997. pp 71–76.

## APPENDIX A: DYNAMIC CROSSOVER OPERATORS

In Ref. 52, dynamic crossover operators for RCGAs were proposed based on the use of parameterized fuzzy connectives. These operators keep a suitable sequence between the exploration and the exploitation along the GA run: “to protect the exploration in the initial stages and the exploitation later.”

To describe these operators, two steps are followed: in Section A1, we define function families for the combination of genes and in Section A2, we use these families to design dynamic crossover operators.

### A1. Function Families for the Combination of Genes

Regarding the exploration and exploitation intervals shown in Figure 13, in Ref. 35, three monotone and nondecreasing functions are proposed:  $F$ ,  $S$ , and  $M$ , defined from  $[a, b] \times [a, b]$  into  $[a, b]$ ,  $a, b \in \mathfrak{R}$ , which fulfill

$$\forall c, c' \in [a, b], \quad F(c, c') \leq \min\{c, c'\}, \quad S(c, c') \geq \max\{c, c'\},$$

$$\text{and } \min\{c, c'\} \leq M(c, c') \leq \max\{c, c'\}$$

Each one of these functions allows us to combine two genes giving results belonging to each one of the aforementioned intervals. Therefore, each function will have different exploration or exploitation properties depending on the range being covered by it.

For an RCGA with a maximum number of generations  $g_{\max}$ , in Ref. 52, three families of functions were proposed: a family of  $F$  functions  $\mathcal{F} = (F^1, \dots, F^{g_{\max}})$ , a family of  $S$  functions  $\mathcal{S} = (S^1, \dots, S^{g_{\max}})$ , and a family of  $M$  functions  $\mathcal{M} = (M^1, \dots, M^{g_{\max}})$ , which for  $1 \leq t \leq g_{\max} - 1$  fulfill

$$\forall c, c' \in [a, b] \quad F^t(c, c') \leq F^{t+1}(c, c') \quad \text{and} \quad F^{g_{\max}}(c, c') \approx \min\{c, c'\}$$

$$\forall c, c' \in [a, b] \quad S^t(c, c') \geq S^{t+1}(c, c') \quad \text{and} \quad S^{g_{\max}}(c, c') \approx \max\{c, c'\}$$

$$\forall c, c' \in [a, b] \quad M^t(c, c') \geq M^{t+1}(c, c') \quad \text{or} \quad M^t(c, c') \leq M^{t+1}(c, c') \quad \forall t$$

and

$$M^{g_{\max}}(c, c') \approx M_{\lim}(c, c')$$

where  $M_{\lim}$  is an  $M$  function called  $M$  limit function. We shall denote  $\mathcal{M}^+$  or  $\mathcal{M}^-$  an  $\mathcal{M}$  function family fulfilling the first and the second part of the last property, respectively.

$\mathcal{F}$  and  $\mathcal{S}$  function families may be built using a parameterized  $t$ -norm  $T^q$  converging on the minimum and a parameterized  $t$ -conorm  $G^q$  converging on the maximum, respectively. In this study, we use the Dubois parameterized  $t$ -norm and  $t$ -conorm:

$$x, y \in [0, 1], \quad T^q(x, y) = \frac{x \cdot y}{x \vee y \vee q} \quad \text{and}$$

$$G^q(x, y) = 1 - \frac{(1-x) \cdot (1-y)}{(1-x) \vee (1-y) \vee q}, \quad 0 \leq q \leq 1$$

To do so, a transformation of the genes to be combined is needed from the interval  $[a, b]$  into  $[0, 1]$  and, later, the result into  $[a, b]$ . Furthermore, we need a function  $\partial(\cdot)$  for transforming the values of  $\{1, \dots, g_{\max}\}$  into the range of  $q$ . All this may be summed up in the following equations. Given the Dubois parameterized  $t$ -norm  $T^q$  and  $t$ -conorm  $G^q$ , we build two function families  $\mathcal{F} = (F^1, \dots, F^{g_{\max}})$  and  $\mathcal{S} = (S^1, \dots, S^{g_{\max}})$  as

$$\forall c, c' \in [a, b] 1 \leq t \leq g_{\max} \quad F^t(c, c') = a + (b-a) \cdot T^{\delta_F(t)}(s, s')$$

$$\forall c, c' \in [a, b] 1 \leq t \leq g_{\max} \quad S^t(c, c') = a + (b-a) \cdot G^{\delta_S(t)}(s, s')$$

where  $s = [(c-a)/(b-a)]$ ,  $s' = [(c'-a)/(b-a)]$ , and  $\delta_F(t) = \delta_S(t) = 1/t$  are the transformation functions.

We may obtain  $\mathcal{M}$  function families using a parameterized averaging function, e.g.,

$$\forall x, y \in [0, 1], \quad P^q(x, y) = \sqrt[q]{\frac{x^q + y^q}{2}}, \quad -\infty \leq q \leq \infty$$

This may be accomplished as follows:

$$\forall c, c' \in [a, b], \quad 1 \leq t \leq g_{\max} \quad M'(c, c') = a + (b - a) \cdot P^{\delta_M(t)}(s, s')$$

In particular, we could obtain an  $\mathcal{M}^+$  and an  $\mathcal{M}^-$  function family using  $\delta_M = \delta_{M^+} = 1 + \ln(g_{\max}/t)$  and  $\delta_M = \delta_{M^-} = 1 + \ln(t/g_{\max})$ , respectively.

## A2. $\mathcal{F}$ -Crossover, $\mathcal{S}$ -Crossover, and $\mathcal{M}$ -Crossover

Let's consider  $C_1^t = (c_1^{1t}, \dots, c_n^{1t})$  and  $C_2^t = (c_1^{2t}, \dots, c_n^{2t})$ , two chromosomes that were selected in the generation  $t$ , and apply the crossover operator to them. If  $O^t$  is the  $t$ th function belonging to an  $\mathcal{F}$ ,  $\mathcal{S}$ , and  $\mathcal{M}$  function family, then we could generate an offspring  $H^t = (h_1^t, \dots, h_n^t)$  as

$$h_i^t = O^t(c_i^{1t}, c_i^{2t}), \quad i = 1, \dots, n$$

If, during the GA run, we use the functions  $Q^s$  for  $s = 1, \dots, g_{\max}$  belonging to an  $\mathcal{F}$ ,  $\mathcal{S}$  or  $\mathcal{M}$  family, as described previously, then we shall call this type of crossover operator  $\mathcal{F}$ -crossover,  $\mathcal{S}$ -crossover, or  $\mathcal{M}$ -crossover, respectively. Note that the  $\mathcal{F}$ -crossovers and the  $\mathcal{S}$ -crossovers show exploration properties, whereas  $\mathcal{M}$ -crossovers show exploitation properties.

The Dubois  $\mathcal{F}$ ,  $\mathcal{S}$ , and  $\mathcal{M}$  function families designed in Section A1 may be used for building an  $\mathcal{F}$ -crossover,  $\mathcal{S}$ -crossover, and  $\mathcal{M}$ -crossover, respectively. In this study, they will be called the Dubois DHXs.

## APPENDIX B: DHX OPERATORS

The DHXs<sup>52</sup> put together the features of the heuristic crossovers (offspring close to the best parent) and the features of the dynamic crossovers (Appendix A). They allow the level of heuristic effects to be dependent on the current generation in which they are applied. At the beginning, this level is low and diversity is high (offspring are distant from parents); later on, the heuristic effects gradually increase.

Let's suppose  $C_1^t = (c_1^{1t}, \dots, c_n^{1t})$  and  $C_2^t = (c_1^{2t}, \dots, c_n^{2t})$ , two chromosomes selected to apply the crossover to them in a generation  $t$ . Let's also suppose that  $C_1^t$  is the one with the best fitness. Then, we may generate  $H^t = (h_1^t, \dots, h_n^t)$  using one of the following DHXs crossovers<sup>52</sup>:

Dynamic dominated crossover

$$h_i^t = \begin{cases} F^t(c_i^{1t}, c_i^{2t}) & \text{if } c_i^{1t} \leq c_i^{2t} \\ S^t(c_i^{1t}, c_i^{2t}) & \text{otherwise} \end{cases} \quad i = 1, \dots, n$$

where  $F^t$  and  $S^t$  belong to an  $\mathcal{F}$  and  $\mathcal{S}$  function family, respectively. We may use the Dubois  $\mathcal{F}$  and an  $\mathcal{S}$  families (Section A2) to obtain the Dubois dynamic dominated crossover operators. Dynamic dominated crossovers have heuristic exploration properties, which allow useful diversity to be introduced into the RCGA population.

Dynamic biased crossover. The  $h_i^t = M_i^t(c_i^{1t}, c_i^{2t})$ ,  $i = 1, \dots, n$ , where  $M_i^t$  belongs to a function family  $\mathcal{M}^+$  if  $c_i^{1t} \leq c_i^{2t}$  or to  $\mathcal{M}^-$ , otherwise. Both have the same  $M$  limit function, which fulfills

$$|c_i^{1t} - M_{\text{lim}}(c_i^{1t}, c_i^{2t})| \leq |c_i^{2t} - M_{\text{lim}}(c_i^{1t}, c_i^{2t})|$$

The following parameterized averaging operator was used to build such an operator:

$$\forall x, y \in [0, 1], \quad 0 \leq q \leq 1, \quad P^q(x, y) = q \cdot x + (1 - q) \cdot y$$

Also, we consider the following initial conditions:

- (1)  $M^1(c_i^1, c_i^2) = [(c_i^1 + c_i^2)/2]$
- (2)  $M_{\text{lim}}(c_i^1, c_i^2) = (1 - \lambda) \cdot c_i^1 + \lambda \cdot c_i^2$

where  $\lambda$  is computed as

$$\lambda = 1 - \frac{f(C_1)}{f(C_1) + f(C_2)}$$

$f(\cdot)$  being the fitness function.

Dynamic biased crossover shows heuristic exploitation properties, which induce a biased convergence toward the best elements.

## APPENDIX C: TEST FUNCTIONS

### Sphere model.<sup>65</sup>

$$f_{\text{Sph}}(x) = \sum_{i=1}^n x_i^2$$

$-5.12 \leq x_i \leq 5.12$ ,  $n = 25$ , and  $f_{\text{Sph}}(x^*) = 0$ .

### Schwefel's function 1.2.<sup>73</sup>

$$f_{\text{Sch}}(x) = \sum_{i=1}^n \sum_{j=1}^i x_j^2$$

$-65.536 \leq x_i \leq 65.536$ ,  $n = 25$ , and  $f_{\text{Sch}}(x^*) = 0$ .

### Generalized Rastrigin's function.<sup>74</sup>

$$f_{\text{Ras}}(x) = a \cdot n + \sum_{i=1}^n x_i^2 - a \cdot \cos(\omega \cdot x_i)$$

$a = 10$ ,  $\omega = 2 \cdot \pi$ ,  $-5.12 \leq x_i \leq 5.12$ ,  $n = 25$ , and  $f_{\text{Ras}}(x^*) = 0$ .

**Griewangk's function.**<sup>75</sup>

$$f_{\text{Gri}}(x) = \frac{1}{d} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$d = 4000$ ,  $-600 \leq x_i \leq 600$ ,  $n = 25$ , and  $f_{\text{Gri}}(x^*) = 0$ .

**Expansion of F10.**<sup>76</sup>

$$e_{\text{F10}}(x) = f_{10}(x_1, x_2) + \cdots f_{10}(x_{i-1}, x_i) \cdots + f(x_n, x_1)$$

$$f_{10}(x, y) = (x^2 + y^2)^{0.25} \cdot [\sin^2(50 \cdot (x^2 + y^2)^{0.1}) + 1]$$

$x, y \in (-100, 100]$ ,  $e_{\text{F10}}(x^*) = 0$ .

**Generalized Rosenbrock's function.**<sup>65</sup>

$$f_{\text{Ros}}(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$$

$-5.12 \leq x_i \leq 5.12$ ,  $n = 25$ , and  $f_{\text{Ros}}(x^*) = 0$ .

**Systems of linear equations.**<sup>77</sup> The problem may be stated as solving for the elements of a vector  $\mathbf{X}$ , given the matrix  $\mathbf{A}$  and vector  $\mathbf{B}$  in the expression:  $\mathbf{A} \cdot \mathbf{X} = \mathbf{B}$ . The evaluation function used for these experiments is

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

Clearly, the best value for this objective function is  $P_{\text{sle}}(x^*) = 0$ . Interparameter linkage (i.e., nonlinearity) is controlled easily 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.**<sup>78</sup> The problem is to specify six parameters  $a_1, w_1, a_2, w_2, a_3,$  and  $w_3$  of the frequency modulation sound model represented by

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

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

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

where the model data are given by the following equation:

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

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

**Polynomial fitting problem.**<sup>79</sup> This problem lies in finding the coefficients of the following polynomial in  $z$ :

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

such that  $P(z) \in [-1, 1]$  for  $z \in [-1, 1]$ , and  $P(1.2) \geq T_{2k}(1.2)$  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 the 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 used here is:

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

It is a nine-parameter problem. The pseudocode algorithm shown in the following was used in order to transform the constraints of this problem into an objective function to be minimized, called  $P_{\text{Chev}}$ . 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  $p_0, \dots, p_{100}$  from  $[-1, 1]$ ;
 $R = 0$ ;
For  $i = 0, \dots, 100$  do
  If  $(-1 > P_C(p_i) \text{ or } P_C(p_i) > 1)$  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 of  $-512$ – $512$ . The objective function value of the optimum is  $P_{\text{Chev}}(\mathbf{C}^*) = 0$ .

**Ackley's function.**<sup>80</sup>

$$f_{\text{Ack}}(x) = -a \cdot \exp\left(-b \cdot \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos \omega \cdot x_i\right) + a + e$$

$a = 20$ ,  $b = 0.2$ ,  $\omega = 2 \cdot \pi$ ,  $-32.768 \leq x_i \leq 32.768$ ,  $n = 25$ , and  $f_{\text{Ack}}(x^*) = 0$ .

**Bohachevsky's function.**<sup>81</sup>

$$f_{\text{Boh}}(x) = x_1^2 + 2x_1^2 - 0.3 \cos(3\pi x_1) \cos(4\pi x_2) + 0.3$$

$-6 \leq x_i \leq 6$  and  $f_{\text{Boh}}(x^*) = 0$ .

**Watson's function.**<sup>81</sup>

$$f_{\text{Wat}}(x) = \sum_{i=1}^{30} \left( \sum_{j=1}^5 (j a_i^{j-1} x_{j+1}) - \left[ \sum_{j=1}^6 a_i^{j-1} x_j \right]^2 - 1 \right)^2 + x_1^2$$

$a_i = [(i - 1)/29]$ ,  $-2 \leq x_i \leq 2$ , and  $f_{\text{Wat}}(x^*) = 2.28800e - 3$ .

**Colville's function.**<sup>81</sup>

$$f_{\text{Col}}(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 + 90(x_3^2 - x_4) + (1 - x_3)^2 \\ + 0.1((1 - x_2)^2 + (1 - x_4)) + 19.8(x_2 - 1)(x_4 - 1)$$

$-10 \leq x_i \leq 10$  and  $f_{\text{Col}}(x^*) = 0$ .

**APPENDIX D: RESULTS OF THE EXPERIMENTS****Table DI.** Results for the sphere model and the Schwefel's function 1.2.

Sphere	B	A	T1	T2	SD	Schwefel	B	A	T1	T2	SD
SX	1.84E-09	5.75E-09	+	+	3.90E-09	SX	7.75E+01	6.10E+02	+	+	3.36E+02
AX	1.74E-09	1.09E-08	+	+	6.53E-09	AX	1.15E+01	4.03E+01	+	+	2.68E+01
GX	2.09E-09	1.35E-08	+	+	1.21E-08	GX	1.30E+02	5.94E+02	+	+	5.10E+02
BLX-0	2.54E-09	1.28E-08	+	+	1.09E-08	BLX-0	9.86E+00	4.00E+01	+	+	1.84E+01
BLX-0.3	1.27E-11	7.51E-11	+	+	5.35E-11	BLX-0.3	8.04E+00	3.37E+01	+	+	1.56E+01
BLX-0.5	6.12E-07	6.31E-06	+	+	8.11E-06	BLX-0.5	5.76E+02	1.36E+03	+	+	2.60E+02
SBX-2	4.38E-10	1.97E-09	+	+	1.17E-09	SBX-2	7.09E-01	7.56E+00	**	+	4.28E+00
SBX-5	6.00E-11	2.76E-10	+	+	2.08E-10	SBX-5	1.14E+01	9.54E+01	+	+	7.97E+01
FR	4.62E-12	1.30E-11	+	+	6.52E-12	FR	1.10E+00	8.97E+00	≈	+	7.08E+00
BLX-0.5-0	1.15E+01	3.19E+01	+	+	9.55E+00	BLX-0.5-0	8.14E+03	1.46E+04	+	+	4.90E+03
WHX	3.93E-03	4.54E-02	+	+	4.58E-02	WHX	7.43E+01	2.09E+03	+	+	2.49E+03
2PX	8.82E-10	3.77E-09	+	+	3.61E-09	2PX	1.24E+02	4.78E+02	+	+	2.56E+02
UX	1.73E-09	1.06E-08	+	+	8.13E-09	UX	2.64E+02	7.21E+02	+	+	3.44E+02
BGAX	2.15E-03	4.98E-03	+	+	1.95E-03	BGAX	4.03E+01	1.11E+02	+	+	5.67E+01
DHX	1.35E-15	1.37E-14	**	**	9.63E-15	DHX	1.91E+01	6.04E+01	+	+	2.99E+01
MMAX	1.80E-12	3.17E-11	+	+	3.75E-11	MMAX	4.46E+01	1.77E+02	+	+	8.32E+01
DX	1.29E-09	5.75E-09	+	+	3.73E-09	DX	1.18E+02	4.39E+02	+	+	1.71E+02
LX	6.94E-11	3.19E-10	+	+	1.70E-10	LX	4.11E-02	3.86E-01	**	+	2.66E-01

**Table DII.** Results for the Rastrigin and Griewank's functions.

Rastrigin	B	A	T1	T2	SD	Griewangk	B	A	T1	T2	SD
SX	2.08E-07	6.63E-01	+	+	5.93E-01	SX	1.00E-06	2.91E+02	+	+	2.75E-02
AX	9.94E-01	3.97E+00	+	+	2.59E+00	AX	3.59E-06	1.78E≈02	+	+	1.87E-02
GX	1.39E+01	1.94E+01	+	+	3.80E+00	GX	7.42E-06	1.38E≈02	+	+	1.15E-02
BLX-0	9.94E-01	4.47E+00	+	+	2.01E+00	BLX-0	2.26E-06	1.55E≈02	+	+	1.82E-02
BLX-0.3	4.97E+00	7.86E+00	+	+	1.80E+00	BLX-0.3	1.10E-08	1.54E≈02	+	+	1.56E-02
BLX-0.5	6.08E+01	8.72E+01	+	+	1.25E+01	BLX-0.5	5.06E-03	5.29E+01	+	+	2.16E-01
SBX-2	6.96E+00	1.36E+01	+	+	4.56E+00	SBX-2	3.22E-07	1.91E≈02	≈	+	2.28E-02
SBX-5	2.98E+00	7.13E+00	+	+	2.15E+00	SBX-5	8.69E-08	2.32E+02	≈	+	2.51E-02
FR	1.19E+01	1.96E+01	+	+	4.84E+00	FR	3.21E-09	7.71E≈03	≈	+	9.60E-03
BLX-0.5-0	1.01E+02	1.48E+02	+	+	2.90E+01	BLX-0.5-0	4.07E+01	1.10E+02	+	+	3.27E+01
WHX	5.28E+01	9.98E+01	+	+	3.09E+01	WHX	1.01E+00	1.14E+00	+	+	1.25E-01
2PX	1.52E-07	6.96E-01	+	+	7.77E-01	2PX	7.07E-07	2.65E+02	+	+	2.45E-02
UX	6.03E-07	6.96E-01	+	+	7.33E-01	UX	2.15E-06	2.22E+02	+	+	1.93E-02
BGAX	5.33E+00	1.04E+01	+	+	2.32E+00	BGAX	8.66E-01	9.77E+01	+	+	5.21E-02
DHX	8.52E-13	1.13E-11	**	**	1.09E-11	DHX	1.76E-12	9.67E≈03	≈	+	1.32E-02
MMAX	2.62E-09	9.28E-01	+	+	9.23E-01	MMAX	8.07E-09	1.31E-02	+	+	1.60E-02
DX	2.85E-06	1.85E+00	+	+	1.30E+00	DX	8.96E-07	1.35E-02	+	+	1.27E-02
LX	5.56E+00	3.06E+01	+	+	2.97E+01	LX	5.81E-08	2.30E-03	**	+	5.04E-03

**Table DIII.** Results for the expansion of F10 and the system of linear equations.

EF10	B	A	T1	T2	SD	SLE	B	A	T1	T2	SD
SX	9.61E-01	1.91E+00	+	+	7.05E-01	SX	6.35E+01	3.68E+02	+	+	2.32E+02
AX	1.66E+00	3.37E+00	+	+	1.79E+00	AX	2.84E+00	2.50E+01	≈	+	1.89E+01
GX	1.36E+01	5.83E+01	+	+	2.64E+01	GX	5.55E+00	5.31E+01	+	+	5.27E+01
BLX-0	1.12E+00	2.15E+00	+	+	8.62E-01	BLX-0	7.44E+00	2.74E+01	≈	+	1.67E+01
BLX-0.3	1.58E-01	3.18E-01	≈	≈	1.21E-01	BLX-0.3	1.44E+00	2.03E+01	**	+	2.16E+01
BLX-0.5	6.74E+00	1.47E+01	+	+	4.54E+00	BLX-0.5	1.42E+00	2.62E+01	≈	+	2.69E+01
SBX-2	3.21E+00	1.35E+01	+	+	8.26E+00	SBX-2	5.40E-01	3.54E+01	≈	+	3.82E+01
SBX-5	2.54E+00	1.99E+01	+	+	1.46E+01	SBX-5	8.03E+00	1.14E+02	+	+	8.52E+01
FR	1.54E-01	2.45E-01	**	**	7.29E-02	FR	3.53E+00	2.66E+01	≈	+	1.72E+01
BLX-0.5-0	1.41E+02	1.66E+02	+	+	1.50E+01	BLX-0.5-0	5.59E+02	1.26E+03	+	+	3.63E+02
WHX	1.20E+02	1.58E+02	+	+	2.29E+01	WHX	4.43E+00	3.78E+02	+	+	3.79E+02
2PX	5.45E-01	1.60E+00	+	+	8.98E-01	2PX	8.36E+01	2.82E+02	+	+	1.55E+02
UX	1.14E+00	2.70E+00	+	+	1.16E+00	UX	6.76E+01	3.68E+02	+	+	2.00E+02
BGAX	1.61E+01	2.36E+01	+	+	4.72E+00	BGAX	3.25E+01	2.00E+02	+	+	1.37E+02
DHX	1.74E-01	1.31E+00	+	+	8.92E-01	DHX	5.62E+01	1.27E+02	+	+	5.19E+01
MMAX	4.84E-01	3.15E+00		+	2.20E+00	MMAX	4.39E+01	1.12E+02		+	5.95E+01
DX	1.36E+00	5.71E+00		+	3.84E+00	DX	2.78E+01	1.98E+02		+	1.00E+02
LX	4.35E-01	8.89E-01		+	3.14E-01	LX	2.81E-01	2.69E+00		**	1.90E+00

**Table DIV.** Results for the Rosenbrock’s function and the polynomial fitting problem.

Rosenbrock	B	A	T1	T2	SD	PFP	B	A	T1	T2	SD
SX	2.33E-01	4.48E+01	+	+	3.03E+01	SX	6.39E+02	5.24E+03	+	+	4.05E+03
AX	2.13E+01	2.25E+01	+	+	3.98E-01	AX	2.01E+01	1.97E+02	**	+	1.24E+02
GX	2.21E+01	2.27E+01	+	+	1.80E-01	GX	2.65E+01	3.45E+02	≈	+	2.84E+02
BLX-0	2.05E+01	2.22E+01	+	+	5.88E-01	BLX-0	4.54E+01	2.89E+02	≈	+	2.16E+02
BLX-0.3	1.92E+01	2.18E+01	≈	≈	7.35E-01	BLX-0.3	3.53E+01	2.19E+02	≈	+	1.55E+02
BLX-0.5	2.09E+01	2.61E+01	≈	≈	1.42E+01	BLX-0.5	1.95E+01	3.16E+02	≈	+	2.58E+02
SBX-2	1.74E+01	2.99E+01	≈	≈	1.98E+01	SBX-2	3.99E+01	4.18E+02	+	+	2.85E+02
SBX-5	1.64E+00	3.90E+01	+	+	2.71E+01	SBX-5	4.58E+01	8.03E+02	+	+	8.99E+02
FR	1.56E+01	2.54E+01	≈	≈	1.53E+01	FR	6.06E+00	4.51E+02	+	+	3.38E+02
BLX-0.5-0	4.40E+03	1.81E+04	+	+	1.08E+04	BLX-0.5-0	2.19E+04	1.59E+05	+	+	1.36E+05
WHX	2.77E+01	1.39E+02	+	+	9.91E+01	WHX	3.39E+02	3.94E+03	+	+	5.15E+03
2PX	1.31E-01	4.70E+01	+	+	3.18E+01	2PX	6.21E+02	4.77E+03	+	+	3.22E+03
UX	1.60E+00	5.10E+01	+	+	2.96E+01	UX	3.60E+02	4.56E+03	+	+	5.50E+03
BGAX	1.26E+01	7.85E+01	+	+	6.01E+01	BGAX	2.83E+02	2.05E+03	+	+	1.62E+03
DHX	1.99E+01	2.17E+01	**	**	5.70E-01	DHX	1.14E+02	7.40E+02	+	+	4.65E+02
MMAX	1.59E+01	2.67E+01		≈	1.53E+01	MMAX	2.57E+02	1.28E+03		+	9.59E+02
DX	7.10E-02	3.25E+01		≈	2.25E+01	DX	3.12E+02	2.07E+03		+	1.69E+03
LX	2.13E+01	2.20E+01		≈	3.03E-01	LX	9.64E-03	6.35E-01		**	9.03E-01

**Table DV.** Results for the FMS parameter identification and the Ackley's function.

FMSPi	B	A	T1	T2	SD	Ackley	B	A	T1	T2	SD
SX	6.05E-08	1.15E+01	≈	≈	6.17E+00	SX	1.86E-04	4.05E-04	+	+	1.32E-04
AX	1.17E+01	2.14E+01	+	+	3.28E+00	AX	2.43E-04	5.13E-04	+	+	1.93E-04
GX	3.66E-07	1.85E+01	+	+	6.46E+00	GX	2.77E-04	5.55E-04	+	+	1.69E-04
BLX-0	1.14E+01	2.05E+01	+	+	4.05E+00	BLX-0	2.19E-04	5.40E-04	+	+	2.45E-04
BLX-0.3	9.65E-13	1.39E+01	+	+	6.75E+00	BLX-0.3	1.45E-05	3.92E-05	+	+	1.52E-05
BLX-0.5	3.40E-12	1.50E+01	+	+	4.56E+00	BLX-0.5	2.42E-03	1.02E-02	+	+	6.38E-03
SBX-2	1.15E+01	1.79E+01	+	+	4.05E+00	SBX-2	1.17E-04	2.27E-04	+	+	8.51E-05
SBX-5	4.36E-15	1.08E+01	≈	≈	4.98E+00	SBX-5	2.95E-05	9.26E-05	+	+	4.54E-05
FR	2.84E-15	7.30E+00	**	**	6.67E+00	FR	7.18E-06	1.81E-05	+	+	6.44E-06
BLX-0.5-0	1.15E+01	2.30E+01	+	+	3.58E+00	BLX-0.5-0	1.33E+01	1.65E+01	+	+	1.09E+00
WHX	1.80E+01	2.37E+01	+	+	2.08E+00	WHX	2.61E+00	4.28E+00	+	+	9.25E-01
2PX	8.71E-09	1.01E+01	≈	≈	7.79E+00	2PX	1.14E-04	2.54E-04	+	+	8.31E-05
UX	1.15E-07	1.14E+01	≈	≈	6.63E+00	UX	1.88E-04	4.19E-04	+	+	1.52E-04
BGAX	1.12E-04	1.20E+01	≈	≈	6.12E+00	BGAX	4.67E-01	8.73E-01	+	+	3.34E-01
DHX	1.03E-11	1.64E+01	+	+	7.91E+00	DHX	1.52E-07	3.81E-07	**	**	1.67E-07
MMAX	1.05E-06	1.57E+01	+	+	7.62E+00	MMAX	1.21E-05	2.87E-05	+	+	1.41E-05
DX	2.19E-05	1.42E+01	+	+	7.58E+00	DX	1.14E-04	3.78E-04	+	+	1.61E-04
LX	1.16E+01	2.06E+01	+	+	3.31E+00	LX	3.62E-05	8.29E-05	+	+	2.75E-05

**Table DVI.** Results for the Watson and Bohachevsky's functions.

Watson	B	A	T1	T2	SD	Bohachevsky	B	A	T1	T2	SD
SX	1.10E+00	1.12E+00	≈	≈	2.57E-02	SX	7.65E-13	1.15E-11	+	+	1.19E-11
AX	1.11E+00	1.12E+00	+	+	8.41E-03	AX	1.96E-12	2.29E-11	+	+	1.67E-11
GX	1.11E+00	1.12E+00	+	+	1.55E-02	GX	1.17E-12	2.19E-11	+	+	3.26E-11
BLX-0	1.11E+00	1.11E+00	+	+	6.04E-03	BLX-0	1.35E-13	2.29E-11	+	+	3.60E-11
BLX-0.3	1.11E+00	1.11E+00	≈	+	2.79E-03	BLX-0.3	2.22E-16	7.52E-14	+	+	1.23E-13
BLX-0.5	1.11E+00	1.16E+00	+	+	3.50E-02	BLX-0.5	7.09E-13	7.84E-12	+	+	7.70E-12
SBX-2	1.11E+00	1.37E+00	+	+	2.74E-01	SBX-2	4.48E-14	1.74E-12	+	+	2.08E-12
SBX-5	1.11E+00	1.13E+00	≈	≈	4.73E-02	SBX-5	1.99E-15	1.91E-13	≈	≈	4.30E-13
FR	1.11E+00	1.11E+00	≈	+	1.09E-02	FR	1.07E-14	7.33E-14	+	+	6.92E-14
BLX-0.5-0	2.87E+00	1.16E+01	+	+	5.20E+00	BLX-0.5-0	1.69E+00	8.02E+00	+	+	3.73E+00
WHX	2.34E+00	8.54E+00	+	+	2.68E+00	WHX	2.21E-05	7.32E-01	+	+	7.73E-01
2PX	1.11E+00	1.11E+00	≈	+	1.22E-02	2PX	7.15E-13	4.44E-12	+	+	4.71E-12
UX	1.11E+00	1.11E+00	**	+	3.11E-03	UX	4.44E-13	2.22E-11	+	+	2.44E-11
BGAX	1.17E+00	1.31E+00	+	+	1.04E-01	BGAX	1.33E-04	6.55E-04	+	+	3.72E-04
DHX	1.11E+00	1.11E+00	≈	+	3.69E-03	DHX	0.00E+00	0.00E+00	**	**	0.00E+00
MMAX	1.10E+00	1.10E+00	**	**	4.62E-05	MMAX	0.00E+00	4.07E-16	≈	≈	1.03E-15
DX	1.10E+00	1.10E+00	+	+	1.41E-04	DX	8.68E-13	1.12E-11	+	+	7.75E-12
LX	1.11E+00	1.16E+00	+	+	2.65E-02	LX	1.55E-15	4.39E-14	+	+	5.51E-14

**Table DVII.** Results for the Colville’s function.

Coville	B	A	T1	T2	SD
SX	−1.64E+02	−1.63E+02	+	+	3.42E+00
AX	−1.63E+02	−1.50E+02	+	+	1.39E+01
GX	−1.61E+02	−1.44E+02	+	+	1.39E+01
BLX-0	−1.62E+02	−1.54E+02	+	+	6.78E+00
BLX-0.3	−1.64E+02	−1.62E+02	+	+	4.29E+00
BLX-0.5	−1.64E+02	−1.62E+02	+	+	4.71E+00
SBX-2	−9.00E+02	−8.81E+02	≈	≈	1.94E+01
SBX-5	−9.00E+02	−8.86E+02	**	**	1.82E+01
FR	−1.64E+02	−1.63E+02	+	+	3.15E+00
BLX-0.5-0	−1.25E+02	−4.04E+01	+	+	4.02E+01
WHX	−1.64E+02	−1.62E+02	+	+	4.71E+00
2PX	−1.64E+02	−1.63E+02	+	+	2.00E+00
UX	−1.64E+02	−1.61E+02	+	+	5.13E+00
BGAX	−1.64E+02	−1.64E+02	+	+	5.68E−14
DHX	−1.64E+02	−1.62E+02	+	+	2.46E+00
MMAX	−1.64E+02	−1.59E+02		+	7.27E+00
DX	−1.64E+02	−1.58E+02		+	8.80E+00
LX	−1.64E+02	−1.61E+02		+	5.05E+00