Real-Parameter Crossover Operators with Multiple Descendents: An Experimental Study

A. M. Sánchez,[1,†] M. Lozano,[2,‡] C. García-Martínez,[3,§]
D. Molina,[4,¶] F. Herrera[2]
[1]*Department of Software Engineering, University of Granada, 18071, Granada, Spain*
[2]*Department of Computer Science and Artificial Intelligence, University of Granada 18071, Granada, Spain*
[3]*Department of Computing and Numerical Analysis, University of Córdoba 14071, Córdoba, Spain*
[4]*Department of Software Engineering, University of Cádiz 11002, Cádiz, Spain*

Crossover operators with multiple descendents produce more than two offspring for each pair of parents. They were suggested as an alternative method to the common practice of generating only two offspring per couple. An offspring selection mechanism is responsible for choosing the two offspring that become the children contributed by the mating. Recently, there has been an increasing interest in incorporating this crossover scheme into real-coded genetic algorithm models because its operation was particularly suitable to attain reliable and accurate solutions for many continuous optimization problems.

In this paper, we undertake an extensive empirical study of the main factors that affect the performance of real-parameter crossover operator with multiple descendents. To do this, we focus our attention on three well-known neighborhood-based real-parameter crossover operators, BLX-α, fuzzy recombination, and PNX. The experimental results obtained confirm that the generation of multiple descendents along with the offspring selection mechanism that chooses the two best offspring may enhance the operation of these three crossover operators. Another important finding from our experiments is that real-coded genetic algorithms with crossover operators with multiple descendents are more efficient than standard real-coded genetic algorithms, that is, they offer solutions with higher quality, requiring fewer fitness function evaluations.

*Author to whom all correspondence should be addressed: e-mail: herrera@decsai.ugr.es.
†e-mail: amlopez@ugr.es.
‡e-mail: lozano@decsai.ugr.es.
§e-mail: cgarcia@uco.es.
¶e-mail: daniel.molina@uca.es.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 23, 246–268 (2008)
 Published online in Wiley InterScience (www.interscience.wiley.com). • DOI 10.1002/int.20258

1. INTRODUCTION

In the initial formulation of the *genetic algorithms* (GAs),[1,2] the candidate solutions were coded using the binary alphabet; however, other coding types, such as the *real coding*, have also been taken into account to deal with the representation of the problem. The real coding approach seems to be adequate when tackling optimization problems of parameters with variables in continuous domains.[3–7] GAs based on real-number representation are called *real-coded* GAs (RCGAs). Over the past few years, many researchers have been paying attention to RCGAs,[5,8–14] and recently, there has been an increasing interest in solving real-world optimization problems using these algorithms.

The *crossover operator* has always been regarded as one of the main search operators in GAs[15–17] because it exploits the available information in previous samples to influence future searches. This is why most RCGA research has been focused on developing effective real-parameter crossover operators, and as a result, many different possibilities have been proposed.[5,6,18,19] *Neighborhood-based real-parameter crossover operators* are a family of operators that have currently received special attention.[18] They determine the genes of the offspring extracting values from intervals defined on neighborhoods associated with the genes of the parents, throughout *probability distributions*. Examples are *BLX-α*,[20,21] *PNX*,[22] and *fuzzy recombination*,[23] which are based on uniform, normal, and triangular probability distributions, respectively. The degree of diversity induced by these operators may be easily adjusted by means of varying an associated *crossover step size parameter*. The greater its value is, the higher the variance (*diversity*) introduced into the population.

Usually, the crossover operator is applied to pairs of chromosomes, generating two offspring for each one of them, which are introduced in the population.[2] However, *crossover operators with multiple descendants* (CX-MDs) have been presented,[24–29] which produce more than two offspring for each pair of parents. In this case, an *offspring selection mechanism* limits the number of offspring that will be population members. The most widely used mechanism selects the two best offspring to form part of the next population.[24,26,29] This particular scheme will be called *two-best-offspring* strategy.

Nowadays, most RCGA models appeared in the literature are based on *neighborhood-based real-parameter CX-MDs*.[7,22,30–34] This success may be because they implement the strategy *few points, many neighbors* by a repeated application of the crossover operator.[35] This strategy, which is the basis of *memetic algorithms*[36] as well, arises as a flexible way to reach an *exploration/exploitation* relationship that allows *reliability* and *accuracy* to be achieved simultaneously.[9,37,38] This is one of the primordial objectives for most continuous optimization problems.

The main purpose of this paper is to determine the importance of real-parameter CX-MDs as a way to improve the performance of real-parameter crossover operators. In particular, we analyze the combination of three important factors that affect the behavior of neighborhood-based real-parameter CX-MDs: (1) the number of offspring per pair of parents, (2) the type of probability distribution of the real-parameter crossover operator (BLX-α, fuzzy recombination, and PNX), and (3) the crossover step size. We have considered the two-best-offspring strategy as offspring selection mechanism to carry out the experiments. One of the essential issues in this study is to examine the synergetic effects among the *diversity* associated with real-parameter crossover operators and the *selection pressure* derived from the offspring selection mechanism.

We set up the paper as follows: In Section 2, we overview some important aspects of neighborhood-based real-parameter crossover operators and describe BLX-α, fuzzy recombination, and PNX. In Section 3, we review different CX-MDs models appeared in the specialized literature, paying particular attention to the models proposed to deal with real coding. In addition, we present the RCGA model with CX-MDs that will be used for the experiments. In Section 4, we describe the experimental study aimed at determining the goodness associated with the neighborhood-based real-parameter CX-MDs. The study is made from two perspectives: the point of view of *efficacy* (quality of the solutions returned) and from the point of view of *efficiency* (whether algorithms are able to find solutions with acceptable quality requiring few fitness function evaluations). Finally, in Section 5, we point out some concluding remarks and summarize a few new promising research directions on the topic. In Appendix A, we include the features of the test suite used for the experiments, and in Appendix B, we include tables with all the results obtained.

2. NEIGHBORHOOD-BASED REAL-PARAMETER CROSSOVER OPERATORS

Neighborhood-based real-parameter crossover operators use probability distributions for creating the genes of the offspring in restricted search spaces around the regions marked by the genes of the parents.[18] Examples are BLX-α,[20,21] fuzzy recombination,[23] and PNX,[22] which are based on uniform, triangular, and normal probability distributions, respectively.

Let us assume that $C_1 = (c_1^1, \ldots, c_n^1)$ and $C_2 = (c_1^2, \ldots, c_n^2)$ are two chromosomes that have been selected to apply the crossover operator to them. Below, we describe the way the three crossover operators create one offspring.

- *BLX-α*.[20,21] The offspring $H = (h_1, \ldots, h_i, \ldots, h_n)$ is generated, h_i is a randomly (uniformly) chosen number from the interval $[C_{\min} - I\alpha, C_{\max} + I\alpha]$ (see Figure 1), where

$$C_{\max} = \max\left\{c_i^1, c_i^2\right\}, \quad C_{\min} = \min\left\{c_i^1, c_i^2\right\}, \text{ and } I = C_{\max} - C_{\min}.$$

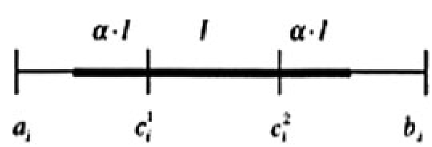

Figure 1. BLX-α.

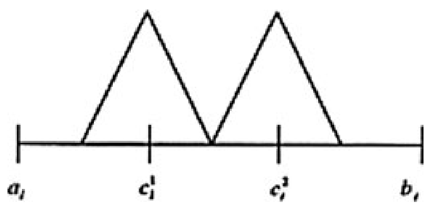

Figure 2. Fuzzy recombination.

Nomura and Shimohara[39] have demonstrated theoretically that BLX-α has the ability to promote *diversity* in the population of an Evolutionary Algorithm (EA). In particular, both authors provide a formalization of this operator to analyze the relationship between the solution probability density functions before and after its application, assuming an infinite population. They state that BLX-α spreads the distribution of the chromosomes when $\alpha > (\sqrt{3} - 1)/2$, otherwise reduces it. This property was verified through simulations.

- *Fuzzy recombination* (FR).[23] The probability that the ith gene in an offspring has the value v_i is given by the distribution $p(v_i) \in \{\phi(c_i^1), \phi(c_i^2)\}$, where $\phi(c_i^1)$ and $\phi(c_i^2)$ are triangular probability distributions having the following features ($c_i^1 \leq c_i^2$ is assumed and $I = |c_i^1 - c_i^2|$):

Probability Distribution	Minimum	Modal	Maximum
$\phi(c_i^1)$	$c_i^1 - d \cdot I$	c_i^1	$c_i^1 + d \cdot I$
$\phi(c_i^2)$	$c_i^2 - d \cdot I$	c_i^2	$c_i^2 + d \cdot I$

where d is the parameter associated with FR. Figure 2 shows an example of applying this crossover operator for the case of $d = 0.5$.

- *PNX*.[22] First, one of the parents is selected randomly (let us consider, e.g., C_1; Figure 3). Then, the probability that the ith gene in an offspring has the value v_i is given by

$$p(v_i) = N\left(c_i^1, \frac{|c_i^1 - c_i^2|}{\eta}\right),$$

where $N(\mu, \sigma)$ is a random number drawn from a Gaussian distribution with mean μ and standard deviation σ, and η is a tunable parameter.

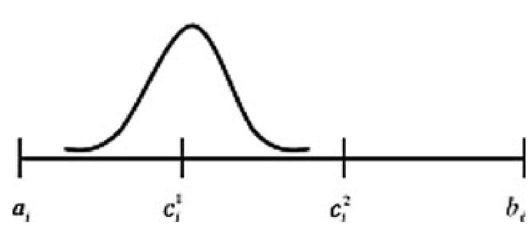

Figure 3. PNX.

FR and PNX are *gene-centric* crossover operators, because each gene of the offspring is generated in the neighborhood of the corresponding gene of one of the parents. PNX chooses all the genes of the same parent to generate the offspring (it is a *parent-centric* crossover operator[30,40]), whilst FR may select genes of the two parents. On the other hand, BLX-α is a *blend* operator that does not show this clear preference toward zones near the genes of the parents.

BLX-α, FR, and PNX share two important features:

- They include in its definition a parameter (α, d, and η, respectively), the *crossover step size*, which determines the spread associated with the probability distributions used to create offspring. The degree of diversity induced by these operators may be easily adjusted by means of varying these *parameters*. The greater their value is, the higher the variance (diversity) introduced into the population. Although some theoretical work exists about them[39,41] and there are adaptive techniques to adjust their values during the GA run,[42] normally these operators are applied considering a fixed value for their crossover step sizes, which was suggested by their creators (Eshelman and Schaffer[21] recommend $\alpha = 0.5$, Ballester and Carter[22] suggested $\eta = 2$, and Voigt et al.[23] proposed $d = 0.5$).
- They define a probability distribution of offspring solutions based on some measure of distance among the parent solutions. If the parents are located closely to each other, the offspring generated by the crossover might distribute densely around the parents. On the other hand, if the parents are located far away from each other, then the offspring will be sparsely distributed around them. An emergent property of this setting is that these crossover operators allow the RCGA to convergence, divergence, or adapt to changing fitness landscapes without incurring into extra parameters or mechanisms to achieve the mentioned behavior. In fact, in the recent past, RCGAs with some of these crossovers have been demonstrated to exhibit *self-adaptive* behavior similar to that observed in evolution strategies and evolutionary programming approaches.[17,43] Moreover, Beyer and Deb[41] argue that a variation operator that harnesses the difference of the parents in the search space is essential for the resulting evolutionary algorithm to exhibit self-adaptive behavior on the population level.

3. CROSSOVER OPERATORS WITH MULTIPLE DESCENDENTS

One of the first versions of CX-MDs is *brood recombination*, which was studied in *genetic programming*.[44] Drawing on the work by Altenberg,[25] Tackett[26] devised this method to compensate for the highly disruptive type of crossover used in genetic programming. Brood recombination promotes the idea that by exploring more combinations of an individual crossover operation, there is more opportunity for success. Thus, instead of a single or pair of crossover offspring, there is a large number, or *brood*. A tournament is held between the members of the brood and the winner is considered to be the offspring contributed by the mating. With this mechanism, Tackett attempted to model the observed fact that many animal species produce far more offspring than are expected to live. Although there are many different mechanisms, the excess offspring die. This is a hard but effective way to cull out the results of bad crossover.

Esquivel et al. were pioneering researchers to study the effects of CX-MDs on standard GAs (with binary coding).[28,45] They present *multiple crossovers per couple*, which deeply explore the recombination possibilities of previously found solutions by allowing more than one crossover operation for each mating pair. Since all of the generated offspring are introduced in the new population, the number of children per couple is fixed as a maximum number and the process of producing offspring is controlled, for each mating pair, in order to not exceed the population size. The authors applied multiple crossovers per couple to optimize classic testing functions and some harder (nonlinear, nonseparable) functions. They found that (1) multiple crossovers per couple may provide an extra benefit in processing time and similar quality of solutions when contrasted against the conventional approach, which applies a single crossover operation per couple and (2) best quality results were obtained allowing between two and four crossovers per couple. Esquivel et al.[46–48] proposed an extended version of their proposal, the *multiparent CX-MDs*. In this case, more than two parents produce more than two offspring. These operators were presented to tackle multiobjective optimization problems.

The next section describes some approaches of CX-MDs for real coding. We should point out that other CX-MDs models appeared to cope with different types of problems, such as the *job shop scheduling problem*[49] and the *traveling salesman problem*.[50]

3.1. Real-Parameter Crossover Operators with Multiple Descendents

Nowadays, there exists a clear tendency to design RCGAs using *neighborhood-based real-parameter CX-MDs*.[7,22,30,32–34] Next, we attempt to explain why. For function optimization problems in continuous search spaces, an important difficulty must be addressed: *solutions of high precision must be obtained by the solvers*.[17] In this sense, specific genetic operators for RCGAs have been presented that favor the *local tuning* of the solutions, such as the *nonuniform mutation*.[4] In addition, *real-coded memetic algorithms* have been proposed, which result from the hybridization of RCGAs and local search techniques that efficiently refine solutions.[35]

RCGAs with neighborhood-based real-parameter CX-MDs may be seen as a kind of real-coded memetic algorithm.[35] The justification is the following: Once a *standard* RCGA has found fit areas of the search space, it searches over only a small fraction of the neighborhood around each search point. It must derive its power from integrating multiple single neighborhood explorations in parallel over successive generations of a population. This *many points, few neighbors* strategy is in direct contrast to a hill climber that potentially focuses effort on a greater fraction of the search neighborhood of one point but only around one point at a time. This strategy might be called *few points, many neighbors*.[37] Precisely, with the use of CX-MDs, RCGAs may follow this strategy, with the aim of inducing an effective local search on the neighborhood of the parents involved in crossover.

Thus, neighborhood-based real-parameter CX-MDs are *crossover-based local search techniques*. With the passing of generations, the RCGA loses diversity due to the selective pressure. Under this circumstance, the self-adaptive nature of neighborhood real-parameter crossover operators (they can generate offspring adaptively according to the distribution of parents without any adaptive parameter (Section 2)) allows the creation of offspring distributed densely around the parents, favoring an effective *local tuning*.

Different RCGAs based on neighborhood-based real-parameter CX-MDs have been proposed in the literature. Below, we describe the fundamentals of three of them. Their excellent behavior reveals that the study of this topic is a promising research area for the improvement of the performance of RCGAs.

- *Minimal generation gap* (MGG). This steady-state RCGA model was originally suggested by Satoh et al.[27] and later used in a number of studies.[51–55] A generation alternation is done through applying a crossover operation λ times to a pair of parents randomly chosen from the population. From the parents and their offspring, the best individual is selected and a random one using the roulette wheel technique. The original parents are replaced with them. The elite individual is selected for producing selective pressure and the random one for introducing diversity into the population. No mutation is applied under this mechanism.
- *Generalized generation gap* (G3). Deb et al.[30] modify the MGG model to make it computationally faster by replacing the roulette-wheel selection with a block selection of the best two solutions. The G3 model also preserves elite solutions from the previous iteration. In G3, the recombination and selection operators are intertwined in the following manner:

    1. From the population $P(t)$ select the best parent and μ other parents randomly.
    2. Generate λ offspring from the chosen parents using a multiparent real-parameter crossover operator.
    3. Choose two elements at random from the population $P(t)$.
    4. Form a combined subpopulation of chosen two elements and offspring, choose the best two solutions and replace the chosen two elements with these solutions.

  The results of the G3 method were compared with two commonly used real-coded EAs, the correlated self-adaptive evolution strategy and differential evolution and with a commonly used classical unconstrained optimization method, namely the quasi-Newton algorithm with the BFGS update method. Compared to all these algorithms, G3 consistently and reliably performed better.
- *Family competition* (FC). The FC model of Yang and Kao[9] introduces an alternative variation of CX-MDs that explores the neighborhood of an individual by applying it repeatedly crossover with *different mates*. During the FC procedure, each individual I_p sequentially becomes the family father. With a probability p_c, this family father and another solution I_1 randomly chosen from the rest of the parent population are used as the parents in a crossover operation. Then, the new offspring is operated by mutation to generate an offspring C_1. For each family father, this procedure is repeated L times. Finally, L solutions $(C_1, \ldots, C_L)$ are produced but only the solution C_b with the best value of fitness function survives. Later, a replacement selection is used to select the better one from the family parent and its best individual.

  The FC principle is that each individual in the population does a local search with length L and only the best offspring survives. Since L solutions are created from the same family father and perform selection, the family competition strategy is similar to (1, λ) selection. The authors suggested that FC is a good way to avoid premature convergence but also to keep the spirit of local searches. Experiments made by the authors verified that their approach is more robust than other EAs, such as genetic algorithms, evolution strategies, and evolutionary programming.

Finally, we may remark that the CX-MDs model allows the design of *hybrid real-parameter CX-MDs*. These crossovers use different kinds of crossover operators to produce diverse offspring from the same parents. For example, treating the parents as two points, p_1 and p_2, Wright[24] proposed a *linear crossover* that generates three offspring, $O_1 = 0.5 \cdot p_1 + 0.5 \cdot p_2$, $O_2 = 1.5 \cdot p_1 - 0.5 \cdot p_2$, and $O_3 = -0.5 \cdot p_1 + 1.5 \cdot p_2$. The two most promising points are selected to substitute the parents in the population. In, Herrera et al.[29] another instance of hybrid real-parameter CX-MDs is presented, which generates four offspring for each pair of parents, applying two *explorative* crossovers and two *exploitative* crossovers to them. The two most promising offspring of the four become members of the new population, replacing their parents.

3.2. RCGA Model with Real-Parameter CX-MDs

In this section, we describe the RCGA with CX-MDs that we have used for the experiments. We have taken a *generational* RCGA model that applies the *non uniform mutation operator*.[14] This operator has been widely used with good results.[5] The selection probability calculation follows *linear ranking*[56] ($\eta_{\min} = 0.75$), and the sampling algorithm is *the stochastic universal sampling*.[57] The *elitist* strategy[58] is also considered. This involves making sure that the best performing chromosome always survives intact from one generation to the next.

The CX-MDs mechanism applied is very simple[31]; for each pair of parents, n_d offspring are created applying repeatedly the crossover operator to them. Then, the *two-best-offspring* strategy is applied, providing two children. These children will undergo mutation before they replace their parents in the population.

With this CX-MDs model, the generation of two new individuals requires $n_d + 2$ evaluations, resulting from the evaluation of the n_d offspring and the evaluation of the chromosomes obtained from the mutation of the selected children. In our experiments, we consider different values for n_d and maintain the same number of fitness function evaluations for all the algorithms compared.

4. EXPERIMENTS

Minimization experiments on the test suite described in Appendix A were carried out to study the behavior of the RCGA model with the CX-MDs model presented in Section 3.3. In Section 4.1, we describe the algorithms built to do this. In Section 4.2, we show the results and draw some conclusions from the point of view of the efficacy. In Section 4.3, we compare the results of the best algorithms. Finally, in Section 4.4, we analyze the RCGAs from the point of view of efficiency.

4.1. Algorithms

We have implemented different instances of RCGA with CX-MDs. They are distinguished with regard to crossover operator and crossover step size value (BLX-α with $\alpha = 0.1, 0.3, 0.5$, and 0.7, FR with $d = 0.1, 0.3, 0.5$, and 0.7, and PNX with $\eta = 1, 2, 3$, and 4), and n_d value ($n_d = 2, 4, 6, 8, 16, 32, 64$, and 128). All the possible combinations of these values were investigated.

The population size of the RCGAs is 61 individuals, the probability of updating a chromosome by mutation is 0.125, and the crossover probability is 0.6. We carried

Table I. TB/St obtained using BLX-α with different α values for each n_d value.

2	4	6	8	16	32	64	128
0.3 (90.90)	0.5 (72.72)	0.3 (72.72)	0.3 (63.63)	0.3 (54.54)	0.3 (63.63)	0.3 (72.72)	0.3 (90.90)
0.1 (54.54)	0.3 (63.63)	0.5 (63.63)	0.5 (54.54)	0.5 (54.54)	0.5 (63.63)	0.5 (54.54	0.1 (36.36)
0.5 (36.36)	0.1 (27.27)	0.7 (27.27)	0.1 (45.45)	0.7 (45.45)	0.7 (45.45)	0.7 (36.36)	0.5 (36.36)
0.7 (9.09)	0.7 (0.0)	0.1 (18.18)	0.7 (36.36)	0.1 (18.18)	0.1 (18.18)	0.1 (0.0)	0.7 (18.18)

out all the algorithms 30 times, each one with a maximum of 100,000 fitness function evaluations.

4.2. Analysis of Results

The results obtained are outlined in Tables B1–B3 in Appendix B. The performance measure used is the *A performance*: average of the best-fitness function found at the end of each run. In addition, to facilitate the comparison between a set of algorithms, we have introduced a performance measure called *total best/similar t-test* (TB/St). For each algorithm, this measure specifies the percentage of test functions in which this algorithm obtains either the best A performance or one similar to the best according to a *t-test* (at 0.05 level of significance), considering the algorithms in the set.

To facilitate the analysis of the results in Tables B1–B3, we have introduced two tables for each crossover operator. For each n_d value, the first table (Tables I, III, and V) ranks the crossover step size values (α, d, and η) attending on the TB/St performance obtained (its values appears in brackets as well). The second table (Tables II, IV, and VI) compares (by means of the ST/St performance) the best step size values for each n_d value (they are the first in the corresponding aforementioned table). It displays the step size along with the n_d value (in brackets) and the corresponding TB/St performance obtained in the comparison.

Tables II, IV, and VI show that the best results are returned using $n_d > 2$. This means that our simple CX-MDs model (Section 3.3) with $n_d > 2$ outperforms the

Table II. Comparison of the best combinations of α and n_d values.

$\alpha(n_d)$	TB/St
0.3 (16)	63.63
0.5 (4)	54.54
0.3 (8)	54.54
0.3 (32)	45.45
0.3 (64)	36.36
0.3 (2)	27.27
0.3 (128)	27.27
0.3 (6)	18.18

Table III. TB/St obtained using FR with different d values for each n_d value.

2	4	6	8	16	32	64	128
0.3 (81.81)	0.3 (81.81)	0.5 (63.63)	0.5 (63.63)	0.3 (63.63)	0.3 (81.81)	0.5 (90.0)	0.5 (90.90)
0.5 (54.54)	0.7 (63.63)	0.7 (63.63)	0.7 (63.63)	0.7 (63.63)	0.5 (72.72)	0.7 (63.63)	0.7 (81.81)
0.7 (45.45)	0.1 (45.45)	0.1 (45.45)	0.1 (45.45)	0.5 (54.54)	0.7 (72.72)	0.3 (54.54)	0.3 (36.36)
0.1 (36.36)	0.5 (45.45)	0.3 (45.45)	0.3 (45.45)	0.1 (36.36)	0.1 (45.45)	0.1 (9.09)	0.1 (18.18)

Table IV. Comparison of the best combinations of d and n_d values.

d (n_d)	TB/St
0.5 (8)	90.9
0.3 (16)	81.81
0.3 (4)	63.63
0.5 (6)	54.54
0.5 (64)	54.54
0.3 (32)	45.45
0.5 (128)	36.36
0.3 (2)	18.18

Table V. TB/St obtained using PNX with different η values for each n_d value.

2	4	6	8	16	32	64	128
4 (100)	3 (90.9)	3 (100)	3 (81.81)	3 (100)	3 (100)	2 (81.81)	2 (81.81)
3 (45.45)	4 (81.81)	4 (63.63)	2 (45.45)	2 (54.54)	2 (63.63)	3 (54.54)	3 (54.54)
1 (0)	2 (9.09)	2 (45.45)	4 (45.45)	4 (45.45)	4 (45.45)	4 (45.45)	4 (45.45)
2 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)

Table VI. Comparison of the best combinations of η and n_d values.

η (n_d)	TB/St
3 (8)	90.9
3 (6)	81.81
3 (32)	63.63
3 (4)	36.36
3 (16)	36.36
2 (64)	36.36
2 (128)	9.09
4 (2)	0

standard way of applying BLX-α, FR, and PNX (i.e., with $n_d = 2$). For the case of PNX (Table VI), the performance with $n_d = 2$ is the worst, as compared with the other n_d values, achieving a TB/St performance of 0%. This result suggests that our CX-MDs process arises as a key mechanism for improving the behavior of neighbourhood-based real-parameter crossover operators.

For FR and PNX, the combinations $d = 0.5$ and $n_d = 8$ (Table IV) and $\eta = 3$ and $n_d = 8$ (Table VI) obtain a high TB/St value (90.9%), respectively. Thus, these settings allow a *robust behavior* for PNX and FR to be obtained for test problems with different features.

For BLX-α, $\alpha = 0.3$ offers the best TB/St performance for most n_d values (Table I). However, Table II shows that it was not possible to find a suitable combination of α and n_d values to achieve a robust operation for BLX-α (the best combination presents a TB/St performance of 63.63%). This means that different combinations of these parameters become well suited to different types of problems.

We may extract an important conclusion from this study: Gene-centric crossover operators (FR and PNX) are particularly adequate to design real-parameter CX-MDs models, because they may offer outstanding levels of robustness on problems with different difficulties.

4.3. Comparison of the Best Performing RCGAs

In this section, we undertake the comparison of the RCGAs based on BLX-α, FR, and PNX, respectively, which reached the best solutions. Their features along with the TB/St performance resulting from their comparison are shown in Table VII.

The most remarkable observation is that FR with d=0.5 and $n_d = 8$ obtains a TB/St value of 100%. On the one hand, FR may collect information from the genes of the two parents to build the offspring. This allows the generation of offspring that are more *diverse* than the ones generated by PNX. On the other hand, FR is an instance of gene-centric crossover operator, which has arisen as suitable kind of crossover operator for design CX-MDs models (Section 4.2). This would explain its advantages on the best RCGA based on BLX-α. The joint effects of these two features make *FR with multiple descendents* a robust real-parameter crossover operator.

4.4. Study of the Efficiency

There exist at least two ways to study the performance of search algorithms:

- The first one studies it from the point of view of the *efficacy* of the algorithms. This quality determines the accuracy levels of the solutions returned by the algorithm, without taking into account the time needed to obtain them. Effective algorithms are recommendable for real problems, where the objective function may be evaluated very quickly and a precise

Table VII. Comparison of the best performing RCGAs.

Cross over operator	Crossover step size	n_d	TB/St
BLX-α	$\alpha = 0.3$	16	63.63
FR	$d = 0.5$	8	100
PNX	$\eta = 3$	8	45.45

Table VIII. Comparison of the standard RCGAs with different n_{ev} values.

BLX-α ($\alpha = 0.5$ and $n_d = 2$)		FR ($d = 0.5$ and $n_d = 2$)		PNX ($\eta = 2$ and $n_d = 2$)	
n_{ev}	TB/St	n_{ev}	TB/St	n_{ev}	TB/St
50,000	9.09	50,000	27.27	50,000	45.45
100,000	54.54	100,000	54.54	100,000	54.54
150,000	**100**	150,000	36.36	150,000	63.63
200,000	9.09	**200,000**	**63.63**	**200,000**	**100**

solution is needed. To study the efficacy, we assign enough objective function evaluations to the algorithms and analyze the quality of the final solutions achieved (Section 4.3).

- The second view concentrates on of the *efficiency* of the algorithms. Search algorithms are efficient when they return solutions with an *acceptable* quality requiring few objective function evaluations. They are advisable for the real problem with a time-consuming objective function and where obtaining the exact solution is not the crucial objective.

In this section, we access the *efficiency* of our RCGA model with CX-MDs by running it with different number of fitness function evaluations (n_{ev} = 50,000, 100,000, 150,000, and 200,000) and comparing the results with the ones of the *standard* RCGAs ($n_d = 2$). First, we have executed the standard RCGA based on BLX-α, FR, and PNX, using the step size values recommended by their authors (α = 0.5, d = 0.5, and $\eta = 2$, respectively). For each crossover operator, Table VIII outlines the TB/St performance derived from the comparison of the results obtained by these algorithms, throughout the different n_{ev} values.

In Table VIII, we may observe (in boldface) that the best n_{ev} values are 150,000 when using BLX-α and 200,000 for both FR and PNX.

Now, we compare the results of the standard RCGAs with the best n_{ev} values previously obtained against the corresponding RCGA with CX-MDs using $n_d = 8$ (this value was a promising setting in Section 4.3) and the different n_{ev} values. Tables IX–XI contain the TB/St values obtained from this comparison, for BLX-α, FR and PNX, respectively.

The most significant remark from Table IX is that the RCGA based on BLX-α with multiple descendents with $n_d = 8$ and n_{ev} = 50,000 (and with all the n_{ev} values higher that this one) achieves better TB/St performance than the standard RCGA based on this crossover operator that ends its runs after 150,000 evaluations. Analyzing the FR crossover, the approach with $n_d = 8$ and n_{ev} = 100,000 improves

Table IX. Standard RCGA based on BLX-α with $\alpha = 0.5$ and n_{ev} = 150,000 versus RCGA with CX-MDs.

Algorithms	50,000	100,000	150,000	200,000
RCGA with CX-MDs and $n_d = 8$	90.9	100	100	100
Standard, RCGA based on BLX-α, $\alpha = 0.5$, and n_{ev} = 150,000	27.27	18.18	18.18	18.18

Table X. Standard RCGA based on FR with d = 0.5 and n_{ev} = 200,000 versus RCGA with CX-MDs.

Algorithms	50,000	100,000	150,000	200,000
RCGA with CX-MDs and $n_d = 8$	54.54	90.9	81.81	72.72
Standard RCGA based on FR, $d = 0.5$, and n_{ev} = 200,000	90.9	36.36	63.63	54.54

Table XI. Standard RCGA based on PNX with $\eta = 2$ and n_{ev} = 200,000 versus RCGA with CX-MDs.

Algorithms	50,000	100,000	150,000	200,000
RCGA with CX-MDs and $n_d = 8$	72.72	100	100	100
Standard RCGA based on PNX, $\eta = 2$, and n_{ev} = 200,000	90.9	45.45	18.18	27.27

the results of the standard RCGA with n_{ev} = 200,000, and the same situation occurs using PNX crossover.

From these results, we may conclude that RCGAs with CX-MDs consistently outperform standard RCGAs from the point of view of efficiency. Thus, the implementation of the strategy *few points, many neighbors* by means of the CX-MDs mechanism imposes an *efficient* exploration/exploitation relationship that allows *reliability* and *accuracy* of RCGAs to be highly intensified.

5. CONCLUSIONS

This paper presented an empirical study of a simple CX-MDs model for RCGAs. For each pair of parents, it generates n_d offspring and selects the two best offspring as the children contributed by the mating. Three different instances of this model were implemented by considering three well-known neighborhood-based real-parameter crossover operators, BLX-α, FR, and PNX. The results of combining different step size values and n_d values were analyzed and the efficiency of the model was examined. The main conclusions achieved were

- The application of CX-MDs (with $n_d > 2$) may enhance the operation of BLX-α, FR, and PNX.
- CX-MDs based on gene-centric crossover operators (FR and PNX) achieved the most promising results. They obtained an acceptable robustness on problems with different difficulties. In particular, FR with multiple descendents is the best alternative. This operator makes a better use of the information contained in the genes of the parents than BLX-α does, but, in addition, FR offers a higher exploration potential than PNX.
- RCGAs with CX-MDs are most efficient than standard RCGAs. Putting into practice the strategy “few points, many neighbors” induces a suitable exploration/exploitation relationship for empowering both *reliability* and *accuracy*.

In essence, the research line initiated with the present work is indeed worth of further studies. We are currently extending our investigation to different test suites and real-world problems. We also intend to

- study the behavior of CX-MDs considering other real-parameter crossover operators, such as *multiparent crossover operators*, which combine the features of more than two parents for generating the offspring.[30,51,53,59,60,61]
- Some forms of crossover operators are more suitable to tackle certain problems than others. For this reason, techniques that combine multiple crossovers have been suggested as alternative schemes to the common practice of applying only one crossover model to all the elements in the population.[11] The generation of the multiple descendents may be carried out by using different kinds of real-parameter crossover operators. Thus, *hybrid real-parameter CX-MDs* arises as a promising topic to be investigated.

Acknowledgments

This research has been supported by the project TIN2005-08386-C05-01.

References

1. Holland JH. Adaptation in natural and artificial systems. Ann Arbor, MI: The University of Michigan Press; 1975. (London: The MIT Press; 1992).
2. Goldberg DE. Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley; 1989.
3. Davis L. Handbook of genetic algorithms. New York: Van Nostrand Reinhold; 1991.
4. Michalewicz Z. Genetic algorithms + data structures = Evolutionary programs. Berlin: Springer-Verlag; 1992.
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. Deb K. Multi-objective optimization using evolutionary algorithms. Chichester, UK: Wiley; 2001.
7. Deb K. A population-based algorithm-generator for real-parameter optimization. Soft Comput 2005;9(4):236–253.
8. Chelouah R, Siarry P. (2000). A continuous genetic algorithm designed for the global optimization of multimodal functions. J Heuristics 2000;6(2):191–213.
9. Yang JM, Kao CY. Integrating adaptive mutations and family competition into genetic algorithms as function optimiser. Soft Comput 2000;4:89–102.
10. Hart WE. Rethinking the design of real-coded evolutionary algorithms: making discrete choices in continuous search domains. Soft Comput 2005;9(4):225–235.
11. Herrera F, Lozano M, Sánchez AM. Hybrid crossover operators for real-coded genetic algorithms: an experimental study. Soft Comput 2005;9(4):280–298.
12. Hervás-Martínez C, Ortiz-Boyer D. Analyzing the statistical features of CIXL2 crossover offspring. Soft Comput 2005;9(4):270–279.
13. Someya, H, Yamamura M. A robust real-coded evolutionary algorithm with toroidal search space conversion. Soft Comput 2005;9(4):254–269.
14. Winter G, Galvan B, Alonso S, Gonzalez B, Jiménez JI, Greiner D. A flexible evolutionary agent: Cooperation and competition among real-coded evolutionary operators. Soft Comput 2005;9(4):299–323.
15. 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.
16. Liepins GE, Vose MD. Characterizing crossover in genetic algorithms. Ann Math Artif Intell 1992;5(1):27–34.
17. Kita H. A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms. Evol Comput 2001;9(2):223–241.

18. Herrera F, Lozano M, Sánchez AM. A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. Int J Intell Syst 2003;18(3):309–338.
19. Herrera F, Lozano M. Gradual distributed real-coded genetic algorithms. IEEE Trans. Evol Comput 2000;4(1):43–63.
20. 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.
21. 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; 1993. pp 187–202.
22. Ballester PJ, Carter JN. An effective real-parameter genetic algorithm with parent centric normal crossover for multimodal optimisation. In: Proc Genetic and Evolutionary Computation Conf. 2004. Lecture Notes in Computer Science 3102; Berlin: Springer-Verlag; 2004. pp 901–913.
23. Voigt HM, Mühlenbein H, Cvetkovic D. Fuzzy recombination for the breeder genetic algorithm. In: Eshelman L, editor. Proc Sixth Int. Conf. on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann; 1995. pp 104–111.
24. 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.
25. Altenberg L. The evolution of evoluability in genetic programming. In: Kinnear KE, editor. Advances in genetic programming. Cambridge, MA MIT Press; 1994. pp 47–74.
26. Tackett WA. Greedy recombination and genetic search on the space of computer programs. In: Foundations of genetic algorithms 3. San Francisco, CA: Morgan Kaufmann; 1994. pp 271–297.
27. 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.
28. Esquivel S, Leiva A, Gallard R. Multiple crossover per couple in genetic algorithms. In: Proc 4th IEEE Int. Conf. on Evolutionary Computation (ICEC'97). Piscataway, NJ: IEEE Press; 1997. pp 103–106.
29. 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.
30. Deb K, Anand A, Joshi D. A computationally efficient evolutionary algorithm for real-parameter evolution. Evol Comput J 2002;10(4):371–395.
31. Herrera F, Lozano M, Pérez E, Sánchez AM, Villar P. Multiple crossover per couple with crossover selection of the two best offspring: An experimental study with the BLX- operator for real-coded genetic algorithms. In: Jarijo FJ, Riquelme JC, Toro M, editors. Advances in artificial intelligence—IBERAMIA 2002. Lecture Notes in Artificial Intelligence, vol 2527. Berlin: Springer-Verlag; 2002. pp 392-401.
32. Ballester PJ, Carter JN. Real-parameter genetic algorithms for finding multiple optimal solutions in multi-modal optimization. In: Cantú-Paz E et al., editors. Proc. of the Genetic and Evolutionary Computation Conference. Lecture Notes in Computer Science, vol 2723; Berlin: Springer-Verlag 2003. pp 706–717.
33. Ballester PJ, Stephenson J, Carter JN, Gallagher K. Real-parameter optimization performance study on the CEC-2005 benchmark with SPC-PNX. In: Proc 2005 IEEE Congress on Evolutionary Computation. Piscataway, NJ: IEEE Press; 2005. pp. 498–505.
34. Sinha A, Tiwari S, Deb K. A population-based, steady-state procedure for real-parameter optimization evolutionary computation, 2005. In: Proc 2005 IEEE Congress on Evolutionary Computation. Piscataway, NJ: IEEE Press; 2005. pp 514–521.
35. Lozano M, Herrera F, Krasnogor N, Molina D. Real-coded memetic algorithms with crossover hill-climbing. Evol Comput J 2004;12(3):273–302.
36. Moscato PA. Memetic algorithms: A short introduction. In: Corne D, Dorigo M, Glower F, editors. New ideas in optimization. London: McGraw-Hill; 1999. pp. 219–234.

37. O'Reilly UM, Oppacher F. Hybridized crossover-based search techniques for program discovery. In: IEEE Int Conf on Evolutionary Computation 1995. Piscataway, NJ: IEEE Press; 1995. pp 573–578.
38. Affenzeller M, Wagner S. Offspring selection: A new self-adaptive selection scheme for genetic algorithms. In: Adaptive and natural computing algorithms. Berlin: Springer-Verlag: 2005. pp 218–221.
39. 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.
40. García-Martínez C, Lozano M, Herrera F, Molina D, Sánchez AM. Global and local real-coded genetic algorithms based on parent-centric crossover operators. Eur J Oper Res. 185 (2008) 1088–1113.
41. Beyer HG, Deb K. On self-adaptive features in real-parameter evolutionary algorithms. IEEE Trans Evol Comput 2001;5(3):250–270.
42. Herrera F, Lozano M. Adaptive genetic operators based on coevolution with fuzzy behaviours. IEEE Trans Evol Comput 2001;5(2):1–18.
43. Deb K, Beyer HG. Self-adaptive genetic algorithms with simulated binary crossover. Evol Comput J 2001;9(2):195–219.
44. Koza JR. Genetic Programming. Cambridge, MA: MIT Press; 1992.
45. Esquivel S, Gallard R, Michalewicz Z. MPC: Another approach to crossover in genetic algorithms. In: Actas del Primer Congreso de Ciencias de la Computación; 1995. pp 141–150.
46. Esquivel S, Leiva A, Gallard R. Multiple crossovers between multiple parents to improve search in evolutionary algorithms. In: Proc 1999 Cong. on Evolutionary computation. Piscataway, NJ: IEEE Press; 1999. pp 1589–1594.
47. Esquivel S, Leiva A, Gallard R. Multiplicity in genetic algorithms to face multicriteria optimization. In: Proc 1999 Cong. on Evolutionary Computation. Piscataway, NJ: IEEE Press; 1999. pp 85–90.
48. Leiva HA, Esquivel SC, Gallard RH. Multiplicity and local search in evolutionary algorithms to build the Pareto front. In: Proc XX Int Conf. of the Chilean Computer Science Society. Piscataway, NJ: IEEE Press; 2000. pp. 7–13.
49. 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 Know Based Syst 2002;15(1-2):12–25.
50. Walters T. Repair and brood selection in the traveling salesman problem. In: Eiben AE, Back T, Schoenauer M, Schwefel HP, editors. Proc Parallel problem solving from nature, vol V. Berlin: Springer-Verlag. Lecture Notes in Computer Science vol 1498; 1998. pp 813–822.
51. Kita H, Kobayashi S. Multi-parental extension of the unimodal normal distribution crossover for real-coded genetic algorithms. In: Proc Int Conf. on Evolutionary Computation'99. Piscataway, NJ: IEEE Press; 1999. pp 646–651.
52. Takahashi O, Kita H, Kobayashi S. A distance alternation model on real-coded genetic algorithms. In: Proc 1999 IEEE Int Conf on Systems, Man, and Cybernetics. Piscataway, NJ: IEEE Press; 1999. pp 619–624.
53. Tsutsui S, Yamamura M, Higuchi T. Multi-parent recombination with simplex crossover in real-coded genetic algorithms. In: Proc Genetic and Evolutionary Computation Conf (GECCO-99). San Mateo, CA: Morgan Kaufmann, 1999. pp 657–664.
54. Zhou Y, Li Y, He J, Kang L. Multi-objective and MGG evolutionary algorithm for constrained optimization. In: Congress on evolutionary computation 2003. Piscataway, NJ: IEEE Press; 2003. pp. 1–5.
55. Takahashi O, Kobayashi S. An angular distance dependent alternation model for real-coded genetic algorithms. In: Proc Congress on Evolutionary Computation 2004. Piscataway, NJ: IEEE Press; 2004. pp 2159–2165.
56. Baker JE. Adaptative selection methods for genetic algorithms. In: Grefenstette JJ, editor. Proc. First Int Conf on Genetic Algorithms and Their Applications. Hillsdale, MA: Lawrence Erlbraum Associates; 1985, pp 101–111.

57. Baker JE. Reducing bias and inefficiency in the selection algorithm. In: Grefenstette JJ, editor. Proc Second Int Conf on Genetic Algorithms and Their Applications. Hillsdale, MA: Lawrence Erlbraum Associates; 1987. pp 14–21.
58. De Jong KA. An analysis of the behavior of a class of genetic adaptive systems. PhD dissertation, University of Michigan, 1975.
59. Ono I, Kobayashi S. A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover. In: Bäck T, editors. Proc Seventh Int Conf on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann; 1997. pp. 246–253.
60. Ortiz D, Hervas C, Muñoz J. Genetic algorithm with crossover based on confidence intervals as an alternative to least squares estimation for nonlinear models. In: Fourth Metaheuristics International Conference; 2001. pp 343–347.
61. Someya H, Yamamura M. Genetic algorithm with search area adaptation for the function optimization and its experimental analysis. In: Proc 2001 Cong. on Evolutionary Computation. Piscataway, NJ: IEEE Press; 2001. pp. 933–940.
62. Schwefel HP. Numerical optimization of computer models. Chichester, UK: Wiley; 1981.
63. Törn A, Antanas Z. Global optimization. Lecture Notes in Computer Science, Vol 350. Berlin: Springer-Verlag; 1989.
64. Griewangk AO. Generalized descent of global optimization. J Optim Theory Appl 1981; 34: 11–39.
65. Whitley D, Beveridge R, Graves C, Mathias K. Test driving three 1995 genetic algorithms: New test functions and geometric matching. J Heuristicss 1995;1:77–104.
66. 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; 1997. pp 203–224.
67. Tsutsui S, Fujimoto Y. Forking genetic algorithm with blocking and shrinking modes. In: Forrest S, editor. Proc Fifth Int Conf on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann; 1993. pp 206–213.
68. Storn R, Price K. Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA; 1995.
69. Ackley DH. A Connectionist machine for genetic hillclimbing. Boston, MA: Kluwer Academic Publishers; 1987.
70. Reynolds RC, Chung C. Knowledge-based self-adaptation in evolutionary programming Using Cultural Algorithms. In: Proc 1997 International Conference on Evolutionary Computation. Piscataway, NJ: IEEE Press; 1997. pp 71–76.

APPENDIX A: TEST FUNCTIONS

Sphere model.[58]

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

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

Schwefel's function 1.2.[62]

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

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

Generalized Rastrigin's function.[63]

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

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

Griewangk's function.[64]

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

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

Expansion of F10.[65]

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

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

$$n = 25, x, y \in (-100, 100]\,,\ e_{F10}(x^*) = 0.$$

Generalized Rosenbrock's function.[58]

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

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

Systems of linear equations.[66]

The problem may be stated as solving for the elements of a vector X, given the matrix A and vector B in the expression: $A{\cdot}X = B$. The evaluation function used for these experiments is

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

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

We have studied an example of a 10-parameter problem. We have considered that $-127 \le x_i \le 127$ and the following matrices:

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

Frequency modulation sounds parameter identification.[67]

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

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

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

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

where the model data are given by the following equation:

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

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

Polynomial-fitting problem.[68]

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 direction of high positive ordinate values. This problem has its roots in electronic filter design and challenges an optimization procedure by forcing it to find parameter values with grossly different magnitudes, something very common in technical systems. The Chebychev polynomial employed here is

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

It is a nine-parameter problem. The pseudocode algorithm shown below was used to transform the constraints of this problem into an objective function to be minimized, called P_{Chev}. We consider that $C = (c_0, \ldots, c_8)$ is the solution to be evaluated and

$$P_C(z) = \sum_{j=0}^{8} c_j \times z^j.$$

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

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

Ackley's function.[69]

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

$$- \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos \omega \cdot x_i\right) + a + e$$

$$a = 20,\ b = 0.2,\ \omega = 2 \cdot \pi,\ -32.768 \leq x_i \leq 32.768,$$

$$n = 25,\ f_{Ack}(x^*) = 0.$$

Bohachevsky's function.[70]

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

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

APPENDIX B: RESULTS OF THE EXPERIMENTS

Table B.I. Results for BLX-α.

n_d	α	*Sph*	*Sch*	*Ras*	*Gri*	*EF10*	*SLE*	*Ros*	*PFP*	*FMS*	*Ack*	*Boh*
2	0.1	4.2E−09	5.5E+01	4.6E+00	2.5E−02	1.6E+00	2.3E+01	2.2E+01	2.7E+02	1.7E+01	2.9E−04	6.0E−12
4	0.1	1.9E−11	7.3E+01	3.4E+00	6.7E−03	1.3E+00	6.6E+01	2.2E+01	7.7E+02	1.7E+01	2.1E−05	1.3E−14
6	0.1	7.1E−12	6.8E+01	2.2E+00	4.4E−03	3.6E+00	8.2E+01	2.3E+01	6.0E+02	1.8E+01	1.1E−05	0.0E+00
8	0.1	8.8E−11	8.0E+01	1.5E+00	3.7E−03	5.4E+00	7.8E+01	2.3E+01	7.8E+02	1.4E+01	3.7E−05	1.0E−15
16	0.1	3.7E−08	1.0E+02	1.3E+00	7.5E−03	9.0e+00	6.8E+01	2.2E+01	6.9E+02	1.7E+01	6.2E−04	4.4E−17
32	0.1	2.6E−06	8.4E+01	2.0E+00	1.4E−02	1.0e+01	5.7E+01	2.3E+01	7.2E+02	1.4E+01	7.2E−03	1.9E−14
64	0.1	1.5E−04	6.3E+01	3.0E+00	1.1E−01	1.0E+01	3.6E+01	2.8E+01	5.2E+02	1.8E+01	3.9E−02	7.0E−11
128	0.1	4.2E−04	6.1E+01	4.7E+00	2.7E−01	1.3E+01	2.8E+01	3.4E+01	2.7E+02	1.7E+01	1.4E−01	1.1E−08
2	0.3	7.5E−11	3.3E+01	7.8E+00	1.5E−02	3.1E−01	2.0E+01	2.1E+01	2.1E+02	1.3E+01	3.9E−05	7.5E−14
4	0.3	9.0E−15	3.4E+01	2.6E+00	3.0E−03	2.0E−02	5.7E+01	2.1E+01	3.7E+02	1.2E+01	4.6E−07	0.0E+00
6	0.3	1.1E−15	2.7E+01	1.0E+00	2.9E−03	8.7E−03	6.8E+01	2.1E+01	4.2E+02	9.1E+00	1.5E−07	0.0E+00
8	0.3	3.2E−16	2.7E+01	8.2E−01	4.0E−03	6.7E−03	5.5E+01	2.1E+01	3.9E+02	9.3E+00	1.0E−07	0.0E+00
16	0.3	3.7E−16	2.5E+01	1.0E+00	1.8E−03	8.0E−03	4.1E+01	2.1E+01	3.2E+02	7.2E+00	9.1E−08	0.0E+00
32	0.3	1.6E−14	2.5E+01	1.1E+00	3.2E−04	1.6E−02	3.8E+01	2.3E+01	3.4E+02	5.2E+00	6.0E−07	0.0E+00
64	0.3	3.1E−11	2.2E+01	9.2E−01	2.2E−05	1.3E−01	2.5E+01	2.1E+01	3.3E+02	4.7E+00	1.7E−05	9.2E−16
128	0.3	3.5E−07	1.9E+01	3.4E+00	1.0E−03	2.0E+00	1.9E+01	2.1E+01	1.5E+02	3.5E+00	2.4E−03	2.7E−11
2	0.5	6.3E−06	1.3E+03	8.7E+01	5.2E−01	1.4E+01	2.6E+01	2.6E+01	3.1E+02	1.5E+01	1.0E−02	7.8E−12
4	0.5	1.8E−14	2.3E+00	8.3E+00	4.9E−04	6.8E−02	2.1E+01	2.0E+01	2.1E+02	1.0E+01	6.8E−07	0.0E+00
6	0.5	3.3E−15	6.3E−01	5.3E+00	4.1E−04	2.6E−02	1.9E+01	2.0E+01	2.4E+02	8.1E+00	2.9E−07	0.0E+00
8	0.5	2.2E−15	5.0E−01	3.6E+00	5.7E−04	2.1E−02	1.8E+01	2.0E+01	2.2E+02	9.6E+00	2.3E−07	0.0E+00
16	0.5	4.8E−15	7.8E−01	2.0E+00	6.3E−12	2.3E−02	2.4E+01	2.1E+01	1.5E+02	4.1E+00	3.5E−07	0.0E+00
32	0.5	1.0E−13	1.8E+00	1.6E+00	3.3E−10	7.7E−02	2.5E+01	2.1E+01	1.5E+02	4.3E−15	1.4E−06	1.7E−15
64	0.5	2.8E−10	7.1E+00	3.9E+00	2.9E−06	4.3E−01	2.1E+01	2.1E+01	2.1E+02	1.0E+00	8.6E−05	3.0E−14
128	0.5	4.4E−06	4.0E+01	4.4E+01	1.3E−02	4.2E+00	2.3E+01	2.1E+01	2.7E+02	5.0E−01	1.0E−02	3.1E−10
2	0.7	8.8E+00	4.1E+03	1.1E+02	3.0E+01	1.2E+02	4.3E+02	2.1E+03	1.1E+04	1.5E+01	1.2E+01	3.2E+00
4	0.7	1.8E+00	3.9E+03	1.2E+02	6.8E+00	9.7E+01	2.6E+02	8.9E+02	2.4E+03	1.9E+01	7.7E+00	2.6E−12
6	0.7	4.3−07	3.5E+03	1.3E+02	4.3E−01	1.2E+01	2.7E+01	2.5E+01	3.8E+02	1.5E+01	2.4E−03	5.4E−14
8	0.7	3.8E−09	3.1E+03	1.2E+02	3.7E−02	2.1E+00	9.1E+00	2.1E+01	3.2E+02	9.4E+00	2.9E−04	1.2E−14
16	0.7	1.0E−10	2.3E+03	1.3E+02	2.0E−06	5.4E−01	1.1E+01	2.0E+01	2.3E+02	1.2E+00	4.9E−05	1.1E−14
32	0.7	9.2E−10	1.1E+03	1.1E+02	9.2E−03	7.9E−01	1.3E+01	2.1E+01	1.1E+02	3.8E−09	1.3E−04	1.1E−13
64	0.7	3.5E−07	7.1E+02	1.1E+02	7.1E−03	2.8E+00	1.4E+01	2.1E+01	3.2E+02	3.4E−04	2.7E−03	1.6E−12
128	0.7	2.8E−04	8.2E+02	1.1E+02	3.4E−01	1.2E+01	2.1E+01	2.2E+01	6.4E+02	1.2E+00	1.0E−01	8.5E−09

Table B.II. Results for FR.

n_d	d	*Sph*	*Sch*	*Ras*	*Gri*	*EF10*	*SLE*	*Ros*	*PFP*	*FMS*	*Ack*	*Boh*
2	0.1	6.2E−11	5.1E+02	2.7E+00	1.2E−02	2.4E−01	2.1E+02	5.0E+01	2.4E+03	9.2E+00	4.0E−05	1.1E−13
4	0.1	2.0E−16	4.8E+02	5.6E−01	1.4E−02	3.3E−03	2.5E+02	4.4E+01	3.7E+03	6.2E+00	6.1E−08	0.0E+00
6	0.1	8.4E−18	4.7E+02	7.9E−01	1.6E−02	1.1E−03	2.2E+02	4.8E+01	3.3E+03	4.3E+00	1.3E−08	0.0E+00
8	0.1	4.7E−18	5.1E+02	1.2E+00	1.1E−02	9.9E−04	1.9E+02	5.0E+01	3.0E+03	5.5E+00	1.1E−08	0.0E+00
16	0.1	8.7E−14	3.5E+02	2.0E+00	1.4E−02	8.3E−02	1.2E+02	5.6E+01	3.3E+03	2.7E+00	1.3E−06	0.0E+00
32	0.1	3.2E−07	2.5E+02	2.8E+00	4.2E−02	1.0E+00	1.5E+02	6.5E+01	2.7E+03	3.4E+00	2.4E−01	0.0E+00

(*continued*)

Table B.II. Continued.

n_d	d	*Sph*	*Sch*	*Ras*	*Gri*	*EF10*	*SLE*	*Ros*	*PFP*	*FMS*	*Ack*	*Boh*
64	0.1	6.5E−03	2.3E+02	2.9E+00	3.3E−01	4.8E+00	1.1E+02	6.2E+01	2.3E+03	4.1E+00	1.2E+00	2.8E−09
128	0.1	4.6E−02	2.8E+02	2.8E+00	1.0E+00	1.2E+01	1.2E+02	5.9E+01	2.1E+03	7.3E+00	1.4E+00	1.3E−01
2	0.3	4.8E−12	4.4E+01	1.1E+01	1.2E−02	1.7E−01	5.3E+01	2.6E+01	7.9E+02	1.0E+01	1.1E−05	1.1E−14
4	0.3	1.4E−16	4.8E+01	9.2E−01	1.0E−03	5.2E−03	1.2E+02	3.3E+01	1.1E+03	6.2E+00	6.2E−08	0.0E+00
6	0.3	1.8E−17	4.8E+01	4.9E−01	1.4E−03	2.4E−03	9.5E+01	3.3E+01	1.3E+03	6.9E+00	2.0E−08	0.0E+00
8	0.3	1.0E−17	4.6E+01	5.3E−01	1.6E−03	1.8E−03	1.1E+02	3.6E+01	1.1E+03	2.8E+00	1.5E−08	0.0E+00
16	0.3	1.3E−17	3.3E+01	1.5E+00	1.3E−03	2.0E−03	8.7E+01	3.8E+01	1.0E+03	3.8E−01	1.8E−08	0.0E+00
32	0.3	4.1E−15	3.5E+01	2.1E+00	2.4E−04	1.0E−02	4.3E+01	2.9E+01	7.9E+02	1.1E+00	3.0E−07	0.0E+00
64	0.3	4.8E−10	2.9E+01	1.8E+00	2.4E−04	1.8E−01	6.3E+01	2.8E+01	8.7E+02	8.3E−01	8.0E−05	1.9E−15
128	0.3	1.4E−05	1.1E+02	6.4E+00	2.0E−02	3.1E+00	7.4E+01	3.4E+01	5.4E+02	1.0E+00	1.3E−02	9.0E−10
2	0.5	1.3E−11	8.9E+00	1.9E+01	7.7E−03	2.4E−01	2.6E+01	2.5E+01	4.5E+02	7.3E+00	1.8E−05	7.3E−14
4	0.5	8.2E−16	3.3E+00	3.8E+00	4.9E−04	1.4E−02	4.0E+01	2.3E+01	3.7E+02	6.6E+00	1.4E−07	0.0E+00
6	0.5	1.6E−16	1.9E+00	1.7E+00	2.4E−04	9.2E−03	5.1E+01	2.4E+01	5.1E+02	6.3E+00	6.3E−08	0.0E+00
8	0.5	1.0E−16	2.5E+00	1.3E+00	9.0E−04	7.8E−03	4.8E+01	2.1E+01	5.4E+02	3.7E+00	5.2E−08	0.0E+00
16	0.5	1.7E−16	4.4E+00	1.3E+00	2.4E−04	8.0E−03	4.8E+01	2.1E+01	5.4E+02	9.0E−01	6.0E−08	0.0E+00
32	0.5	6.9E−15	4.3E+00	1.5E+00	3.2E−04	1.8E−02	3.1E+01	2.4E+01	4.1E+02	6.7E−01	4.1E−07	0.0E+00
64	0.5	1.7E−10	9.8E+00	1.2E+00	2.5E−04	2.7E−01	2.7E+01	2.1E+01	3.5E+02	6.9E−01	5.9E−05	4.0E−15
128	0.5	5.4E−06	6.3E+01	1.6E+01	9.7E−03	3.2E+00	3.2E+01	2.2E+01	5.2E+02	3.9E−01	1.0E−02	2.6E−10
2	0.7	1.5E−06	1.4E+03	7.8E+01	4.7E−01	7.5E+00	3.0E+01	2.6E+01	4.8E+02	1.1E+01	4.1E−03	6.9E−12
4	0.7	1.3E−14	2.6E+00	9.3E+00	1.0E−03	4.8E−02	1.7E+01	2.0E+01	2.4E+02	7.2E+00	6.2E−07	0.0E+00
6	0.7	2.0E−15	1.0E+00	5.4E+00	2.4E−04	2.0E−02	2.9E+01	2.0E+01	3.3E+02	5.2E+00	2.4E−07	0.0E+00
8	0.7	1.4E−15	6.2E−01	3.4E+00	4.9E−04	1.7E−02	3.4E+01	2.0E+01	2.4E+02	4.5E+00	1.7E−07	0.0E+00
16	0.7	2.1E−15	8.4E−01	2.6E+00	2.4E−12	1.8E−02	2.9E+01	2.1E+01	2.3E+02	1.6E+00	2.2E−07	0.0E+00
32	0.7	4.6E−14	2.4E+00	1.7E+00	1.2E−10	4.7E−02	2.7E+01	2.1E+01	2.3E+02	3.7E−01	1.0E−06	0.0E+00
64	0.7	3.0E−10	8.2E+00	1.9E+00	1.8E−05	3.7E−01	1.7E+01	2.1E+01	2.7E+02	1.1E+00	8.0E−05	1.6E−14
128	0.7	5.6E−06	5.1E+01	3.6E+01	1.3E−02	4.1E+00	2.9E+01	2.1E+01	4.7E+02	2.0E−01	1.0E−02	2.9E−10

Table B.III. Results for PNX.

n_d	η	*Sph*	*Sch*	*Ras*	*Gri*	*EF10*	*SLE*	*Ros*	*PFP*	*FMS*	*Ack*	*Boh*
2	1	6.5E+03	6.5E+03	1.3E+02	7.1E+01	1.4E+02	5.8E+02	6.5E+03	2.7E+04	1.6E+01	1.4E+01	5.0E+00
4	1	1.9E+01	6.7E+03	1.5E+02	7.4E+01	1.5E+02	5.8E+02	7.0E+03	2.2E+04	1.8E+01	1.5E+01	6.0E+00
6	1	1.9E+01	6.6E+03	1.6E+02	6.2E+01	1.5E+02	5.6E+02	6.6E+03	2.1E+04	1.7E+01	1.4E+01	5.3E+00
8	1	1.7E+01	6.5E+03	1.7E+02	5.7E+01	1.4E+02	5.3E+02	6.3E+03	1.7E+04	1.8E+01	1.4E+01	4.3E+00
16	1	1.2E+01	6.0E+03	1.7E+02	4.8E+01	1.3E+02	4.4E+02	4.3E+03	1.1E+04	1.7E+01	1.4E+01	2.5E−06
32	1	9.3E+00	5.8E+03	1.6E+02	3.2E+01	1.3E+02	3.2E+02	3.2E+03	9.0E+03	8.1E+00	1.2E+01	3.0E−08
64	1	5.0E+00	5.5E+03	1.6E+02	1.8E+01	1.1E+02	2.1E+02	1.9E+03	6.4E+03	4.8E+00	1.0E+01	1.4E−07
128	1	2.8E+00	4.9E+03	1.6E+02	1.0E+01	1.0E+02	1.9E+02	1.1E+03	4.9E+03	8.8E+00	8.8E+00	3.0E−05
2	2	2.9E+00	3.3E+03	1.1E+02	1.0E+01	1.1E+02	3.3E+02	5.5E+02	7.8E+03	1.5E+01	8.8E+00	2.0E+00
4	2	2.0E−03	2.8E+03	1.2E+02	8.8E−01	7.9E+01	7.1E+01	1.5E+02	8.9E+02	1.3E+01	1.1E+00	1.3E−12
6	2	5.5E−08	2.2E+03	1.2E+02	1.1E−01	1.4E+01	1.2E+01	2.3E+01	3.3E+02	5.6E+00	9.2E−04	8.2E−14
8	2	2.7E−09	1.7E+03	1.2E+02	4.9E−03	3.8E+00	1.0E+01	2.0E+01	3.7E+02	3.9E+00	2.2E−04	4.2E−14
16	2	3.7E−10	7.5E+02	1.1E+02	8.2E−03	1.4E+00	1.7E+01	2.0E+01	2.8E+02	1.9E+00	8.8E−05	7.9E−14
32	2	6.1E−09	1.9E+02	1.0E+02	2.2E−02	2.0E+00	1.2E+01	2.0E+01	2.3E+02	3.9E−01	3.4E−04	9.6E−13
64	2	2.2E−06	2.1E+02	1.0E+02	4.2E−02	6.7E+00	1.5E+01	2.1E+01	2.6E+02	7.8E−01	7.1E−03	2.4E−11
128	2	1.1E−03	4.5E+02	1.0E+02	6.4E−01	2.1E+01	3.2E+01	2.4E+01	5.4E+02	3.1E+00	2.6E−01	1.1E−07
2	3	3.5E+01	2.5E+02	4.3E+01	6.9E−02	1.0E+01	5.6E+01	3.5E+01	5.7E+02	1.3E+01	1.5E−03	1.2E−11
4	3	1.9E−11	1.3E+00	2.5E+01	2.4E−02	6.5E−01	4.7E+01	2.5E+01	3.9E+02	5.9E+00	2.1E−05	1.3E−14
6	3	5.0E−12	9.0E−01	2.5E+01	2.0E−02	4.4E−01	3.0E+01	2.8E+01	3.0E+02	4.4E+00	1.0E−05	6.9E−15
8	3	4.3E−12	1.6E+00	2.0E+01	1.5E−02	4.1E−01	3.4E+01	2.4E+01	3.5E+02	6.9E+00	1.0E−05	1.1E−14
16	3	1.6E−11	4.8E+00	2.1E+01	9.3E−03	5.8E−01	2.8E+01	2.0E+01	2.6E+02	4.0E+00	2.0E−05	4.8E−14
32	3	2.8E−09	1.6E+01	2.1E+01	7.8E−03	1.8E+00	2.4E+01	2.0E+01	2.6E+02	3.3E+00	2.2E−04	1.6E−12
64	3	4.6E−06	5.9E+01	3.8E+01	2.2E−02	9.3E+00	2.4E+01	2.3E+01	3.3E+02	5.5E+00	1.1E−02	1.3E−10
128	3	2.1E−03	1.9E+02	7.9E+01	6.7E−01	2.8E+01	3.0E+01	3.1E+01	3.4E+02	6.5E+00	5.3E−01	1.0E−06
2	4	8.7E−10	7.2E+00	3.1E+01	2.4E−02	6.5E+00	7.2E+01	3.7E+01	7.0E+02	9.3E+00	1.1E−04	5.9E−13

(*continued*)

Table B.III. Continued.

n_d	η	*Sph*	*Sch*	*Ras*	*Gri*	*EF10*	*SLE*	*Ros*	*PFP*	*FMS*	*Ack*	*Boh*
4	4	1.6E−11	1.0E+01	2.0E+01	1.6E−02	1.1E+00	5.8E+01	3.2E+01	5.9E+02	7.1E+00	1.9E−05	1.6E−14
6	4	1.0E−11	1.3E+01	2.0E+01	1.6E−02	8.9E−01	6.2E+01	3.0E+01	4.6E+02	8.4E+00	1.5E−05	1.0E−14
8	4	1.1E−11	1.5E+01	1.9E+01	2.7E−02	8.5E−01	4.9E+01	3.3E+01	3.3E+02	6.4E+00	1.6E−05	3.0E−14
16	4	9.7E−11	3.0E+01	2.4E+01	1.5E−02	1.7E+00	3.6E+01	2.4E+01	3.2E+02	6.1E+00	5.9E−05	1.1E−13
32	4	1.7E−07	6.6E+01	2.6E+01	1.6E−02	6.8E+00	4.0E+01	3.5E+01	3.4E+02	6.0E+00	4.9E−05	5.7E−12
64	4	2.0E−04	1.3E+02	3.2E+01	1.5E−01	2.1E+01	2.7E+01	2.6E+01	3.4E+02	7.4E+00	4.6E−01	6.3E−09
128	4	1.3E−02	2.8E+02	5.7E+01	1.0E+00	4.7E+01	4.2E+01	4.1E+01	4.7E+02	9.4E+00	2.1E+00	1.5E−02