

Gradual Distributed Real-Coded Genetic Algorithms

Francisco Herrera and Manuel Lozano

Abstract—A major problem in the use of genetic algorithms is premature convergence, a premature stagnation of the search caused by the lack of diversity in the population. One approach for dealing with this problem is the distributed genetic algorithm model. Its basic idea is to keep, in parallel, several subpopulations that are processed by genetic algorithms, with each one being independent of the others. Furthermore, a migration mechanism produces a chromosome exchange between the subpopulations. Making distinctions between the subpopulations by applying genetic algorithms with different configurations, we obtain the so-called heterogeneous distributed genetic algorithms. These algorithms represent a promising way for introducing a correct exploration/exploitation balance in order to avoid premature convergence and reach approximate final solutions.

This paper presents the gradual distributed real-coded genetic algorithms, a type of heterogeneous distributed real-coded genetic algorithms that apply a different crossover operator to each subpopulation. The importance of this operator on the genetic algorithm's performance allowed us to differentiate between the subpopulations in this fashion. Using crossover operators presented for real-coded genetic algorithms, we implement three instances of gradual distributed real-coded genetic algorithms. Experimental results show that the proposals consistently outperform sequential real-coded genetic algorithms and homogeneous distributed real-coded genetic algorithms, which are equivalent to them and other mechanisms presented in the literature. These proposals offer two important advantages at the same time: better reliability and accuracy.

Index Terms—Crossover operator, distributed genetic algorithms, multiresolution, premature convergence, selective pressure.

I. INTRODUCTION

THE BEHAVIOR of genetic algorithms (GA's) is strongly determined by the balance between exploiting what already works best and exploring possibilities that might eventually evolve into something even better. The loss of critical alleles due to selection pressure, selection noise, schemata disruption due to a crossover operator, and poor parameter settings may make this exploration/exploitation balance disproportionate, and produce a lack of *diversity* in the population [39], [43], [53]. Under these circumstances, the search is likely to be trapped in a region that does not contain the global optimum. This problem, called *premature convergence*, has long been recognized as a serious failure mode for GA's [20], [23].

Diversity preservation methods based on *spatial separation* have been proposed in order to avoid premature convergence [13], [14], [44], [49]–[51], [62]. One of the most important rep-

resentatives are the *distributed* GA's (DGA's). Their premise lies in partitioning the population into several subpopulations, each one of them being processed by a GA, independently of the others. Furthermore, a *migration* mechanism produces a chromosome exchange between the subpopulations. DGA's attempt to overcome premature convergence by preserving diversity due to the semi-isolation of the subpopulations. Another important advantage is that they may be implemented easily on parallel hardware. This concept was offered as early as [8].

Making distinctions between the subpopulations of a DGA through the application of GA's with different configurations (control parameters, genetic operators, codings, etc.), we obtain the so-called *heterogeneous* DGA's [2], [17], [51], [62]. They are suitable tools for producing *parallel multiresolution* in the search space associated with the elements that differentiate the GA's applied to the subpopulations. This means that the search occurs in multiple exploration and exploitation levels. In this way, a distributed search and an effective local tuning may be obtained simultaneously, which may allow premature convergence to be avoided and approximate final solutions to be reached.

The availability of crossover operators for *real-coded* GA's (RCGA's) [34] that generate different exploration or exploitation degrees makes the design of heterogeneous distributed RCGA's based on this operator feasible [33]. This paper presents a proposal of such algorithms, the *gradual distributed* RCGA's (GD-RCGA's). They apply a different crossover operator to each subpopulation. These operators are differentiated according to their associated exploration and exploitation properties and the degree thereof. The effect achieved is a parallel multiresolution with regard to the crossover operator's action. This seems very adequate for introducing reliability and accuracy into the search process. Furthermore, subpopulations are adequately connected for exploiting this multiresolution in a *gradual* way.

II. CROSSOVER OPERATORS FOR REAL-CODED GA'S

Let us assume that $X = (x_1 \cdots x_n)$ and $Y = (y_1 \cdots y_n)$ ($x_i, y_i \in [a_i, b_i] \subset \mathfrak{R}, i = 1 \cdots n$) are two real-coded chromosomes that have been selected for crossover. Most crossover operators presented for RCGA's generate the genes for the offspring via some form of combination of the genes in the parents X and Y [34].

In short, the action interval of the genes x_i and y_i , $[a_i, b_i]$, may be divided into three intervals, $[a_i, x_i]$, $[x_i, y_i]$, and $[y_i, b_i]$, that bound three regions to which the resultant genes of some combination of x_i and y_i may belong. Fig. 1 shows this graphically.

These intervals may be classified as exploration or exploitation zones, as is shown in Fig. 1. The interval with both genes

Manuscript received January 22, 1998; revised October 13, 1998, and April 19, 1999. This work was supported by CICYT TIC96-0778.

The authors are with the Department of Computer Science and Artificial Intelligence, University of Granada, 18071, Granada, Spain (e-mail: {herrera; lozano}@decsai.ugr.es).

Publisher Item Identifier S 1089-778X(00)00628-7.

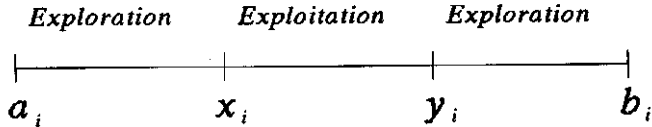


Fig. 1. Action interval for x_i and y_i .

being the extremes is an exploitation zone in the sense that any gene g_i generated by crossover in this interval fulfills $\max\{|g_i - x_i|, |g_i - y_i|\} \leq |x_i - y_i|$. The two intervals that remain on both sides are exploration zones in the sense that this property is not fulfilled. Therefore, exploration and/or exploitation degrees may be assigned to any crossover operator for RCGA's with regard to the way in which these intervals are considered for generating genes. Since the use of exploitative crossover operators does not guarantee the generation of offspring being better than their parents, it seems reasonable to apply them accompanied by exploratory ones [29], [32].

We use the following crossover operators for RCGA's: *fuzzy connectives-based crossovers* (FCB crossovers) [32], *BLX- α* [9], [21], and an extended version of the *fuzzy recombination* presented in [66]. All of these operators allow different exploration or exploitation degrees to be generated. In the following subsections, we comment on their main features.

A. Fuzzy Connectives-Based Crossover Operators

To describe the FCB-crossover operators, we follow two steps: 1) define functions for the combination of genes (Section II-A-1), and 2) use these functions to define crossover operators between two chromosomes (Section II-A-2).

1) *Functions for the Combination of Genes:* With regard to the intervals shown in Fig. 1, in [32], three monotone and non-decreasing functions are proposed: F , S , and M , defined from $[a, b] \times [a, b]$ into $[a, b]$, $a, b \in \mathbb{R}$, and which fulfill

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

and

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

Each 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.

Fuzzy connectives, t norms, t conorms, and averaging functions [48] were used to obtain F , S , and M functions. These functions are defined from $[0, 1] \times [0, 1]$ into $[0, 1]$ and fulfill: 1) t norms are less than the minimum, 2) t conorms are greater than the maximum, and 3) averaging functions are between the minimum and maximum. F was associated to a t norm T , S to a t conorm G , and M to an averaging operator P . In order 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 $[a, b]$. Four families of fuzzy connectives were used for obtaining F , S , and

M functions, which are shown in Table I. These fuzzy connectives accomplish the following property:

$$T_E \leq T_A \leq T_H \leq T_L \leq P_H \leq P_A \leq P_E \leq P_L \\ \leq G_L \leq G_H \leq G_A \leq G_E.$$

2) *F-, S-, and M-Crossover Operators:* Now, if $Q \in \{F, S, M\}$, we may generate the offspring $Z = (z_1 \cdots z_n)$ as

$$z_i = Q(x_i, y_i), \quad i = 1, \dots, n.$$

This crossover operator applies the same F , S , or M function for all of the genes in the chromosomes to be crossed. For this reason, they were called F crossover, S crossover, and M crossover, respectively. Four families of FCB-crossover operators may be obtained using the families of fuzzy connectives in Table I. Each one is termed the same as the related fuzzy connective family.

These crossover operators have different properties: the F - and S -crossover operators show exploration, and the M -crossover operators show exploitation. According to the associated property of the families of fuzzy connectives in Table I, the degree in which each crossover operator shows its related property depends on the fuzzy connective on which it is based. On the one hand, the Einstein F - and S -crossover operators show the maximum exploration, whereas the logical ones represent the minimum exploration. On the other, the logical M -crossover operator shows the maximum level of exploitation since it uses the maximum level of information from both genes, i.e., it is not biased toward either of them. The effects of these crossover operators, along with their associated exploration or exploitation degrees, may be observed in Fig. 2.

B. BLX- α Crossover Operator

BLX- α generates an offspring $Z = (z_1 \cdots z_n)$ where z_i is a randomly (uniformly) chosen number from the interval $[\min_i - I \cdot \alpha, \max_i + I \cdot \alpha]$, where $\max_i = \max\{x_i, y_i\}$, $\min_i = \min\{x_i, y_i\}$, and $I = \max_i - \min_i$. Fig. 3 shows its operation.

In the absence of selection pressure, all values $\alpha < 0.5$ will demonstrate 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 reached between convergence (exploitation) and divergence (exploration), the probability that a gene will lie in the exploitation interval is then equal to the probability that it will lie in an exploration interval [21].

C. Extended Fuzzy Recombination Operator

Here, we extend the *fuzzy recombination* operator presented in [66] (the resultant operator will be called *extended fuzzy recombination*). In this operator, the probability that the i th gene in the offspring has the value z_i is given by the distribution $p(z_i) \in \{\phi_{x_i}, \phi_{av}, \phi_{y_i}\}$, where ϕ_{x_i} , ϕ_{av} , and ϕ_{y_i} are triangular probability distributions having the following features ($x_i \leq y_i$ is assumed), where $I = |y_i - x_i|$, $av = (x_i + y_i)/2$, and $d \in [0, 1]$. Fig. 4 shows two examples of applying this crossover operator.

TABLE I
FAMILIES OF FUZZY CONNECTIVES

Family	T-norm	T-conorm	Averaging Function. ($0 \leq \lambda \leq 1$)
Logical	$T_L(x, y) = \min(x, y)$	$G_L(x, y) = \max(x, y)$	$P_L(x, y) = (1 - \lambda)x + \lambda y$
Hamacher	$T_H(x, y) = \frac{xy}{x+y-xy}$	$G_H(x, y) = \frac{x+y-2xy}{1-xy}$	$P_H(x, y) = \frac{1}{\frac{y-\lambda x-xy+\lambda}{xy} + 1}$
Algebraic	$T_A(x, y) = xy$	$G_A(x, y) = x + y - xy$	$P_A(x, y) = x^{1-\lambda}y^\lambda$
Einstein	$T_E(x, y) = \frac{xy}{1+(1-x)(1-y)}$	$G_E(x, y) = \frac{x+y}{1+xy}$	$P_E(x, y) = \frac{2}{1+(\frac{2-x}{\lambda})^{1-\lambda}(\frac{2-y}{\lambda})^\lambda}$

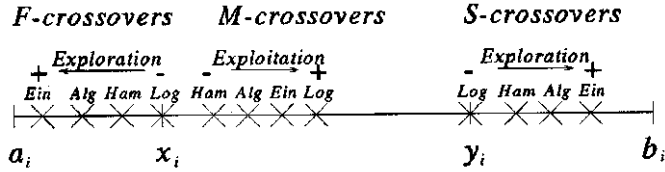


Fig. 2. FCB-crossover operators.

Dist.	Min. Value	Modal Value	Max. Value
ϕ_{x_i}	$x_i - d \cdot I$	x_i	$\min\{x_i + d \cdot I, av\}$
ϕ_{av}	$\min\{x_i + d \cdot I, av\}$	av	$\max\{y_i - d \cdot I, av\}$
ϕ_{y_i}	$\max\{y_i - d \cdot I, av\}$	y_i	$y_i + d \cdot I$

If $d \leq 0.5$, the probability of generating genes belonging to the exploitation interval $[x_i, y_i]$ is higher than that of generating genes in the exploration intervals $[a_i, x_i]$ and $[y_i, b_i]$, as shown in Fig. 4(a). Alternatively, when $d \geq 0.5$, the opposite effect occurs. Fig. 4(b) shows this.

The three crossover operators presented above may be ordered with regard to the way randomness is used for generating the genes of the offspring: 1) FCB crossovers are deterministic, i.e., given two parents, the resultant offspring will always be the same; 2) BLX- α includes a random component, i.e., it is non-deterministic; and 3) extended fuzzy recombination is nondeterministic as well; however, it uses triangular probability distributions, whereas BLX- α uses uniform distributions. In this way, it may be considered as a hybrid between the FCB crossovers and BLX- α . For example, for $d = 0$, it looks like a hybrid between the logical M crossover and BLX-0.0, and for $d = 0.5$, among the logical F crossover, the logical S crossover, and BLX-0.5.

Another important property of these crossover operators is that they fit their action range, depending on the diversity of the population using specific information held by the parents [21], [32].

III. GRADUAL DISTRIBUTED REAL-CODED GENETIC ALGORITHMS

In this section, we propose the GD-RCGA's. They are heterogeneous distributed RCGA's that apply a different crossover operator to each subpopulation. Fig. 5 outlines their basic structure.

They are based on a hypercube topology with three dimensions. There are two important sides to be differentiated.

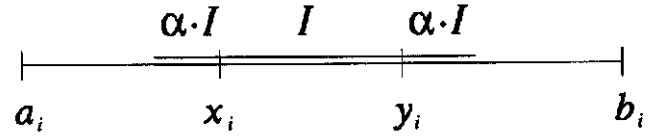


Fig. 3. BLX- α crossover operator.

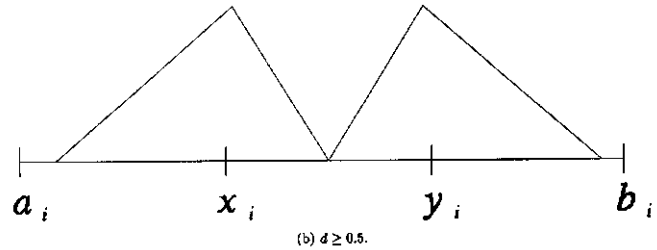
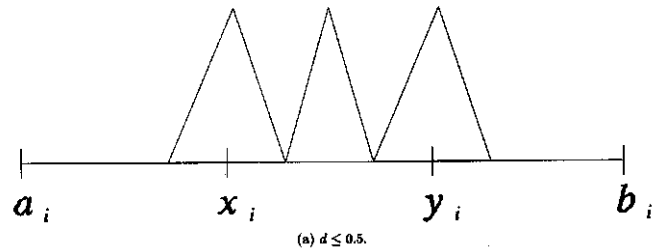


Fig. 4. Extended fuzzy recombination.

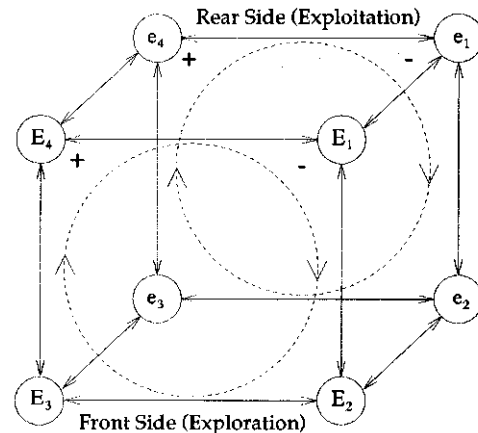


Fig. 5. Structure of a GD-RCGA.

- The front side is devoted to exploration. It is made up of four subpopulations E_1, \dots, E_4 , to which exploratory crossover operators are applied. The exploration degree

increases clockwise, starting at the lowest E_1 , and ending at the highest E_4 .

- The *rear side* is for exploitation. It is composed of subpopulations e_1, \dots, e_4 that undergo exploitative crossover operators. The exploitation degree increases clockwise, starting at the lowest e_1 , and finishing at the highest e_4 .

With this structure, a parallel multiresolution is obtained using the crossover operator, which allows a diversified search (reliability), and an effective local tuning (accuracy) to be achieved simultaneously. Furthermore, subpopulations are adequately connected for exploiting the multiresolution in a *gradual* way since the migrations between subpopulations belonging to different categories may induce the refinement or the expansion of the best zones emerging.

- *Refinement*: This may be induced if migrations are produced from an exploratory subpopulation toward an exploitative one, i.e., from E_i to e_i , or between two exploratory subpopulations from a higher degree to a lower one, i.e., from E_{i+1} to E_i , or between two exploitative subpopulations from a lower degree to a higher one, i.e., from e_i to e_{i+1} .
- *Expansion*: In the case of migrations in the opposite direction, the chromosomes included may be reference points for generating diversity (with different degrees) on zones showing promising properties.

These two effects may improve, even more, the proper reliability and accuracy achieved through multiresolution.

Topology is an important factor in the performance of the DGA's because it determines the speed at which a good solution spreads to other subpopulations. If the topology has a dense connectivity, or a short diameter, or both, good solutions will spread quickly to all of the subpopulations [10]. The short diameter of the cubic topology is suitable for favoring refinement and expansion since genetic material will be quickly exchanged between subpopulations with a wide spectrum of properties, as well as degrees of exploration and exploitation.

Since GD-RCGA's are implemented easily on parallel hardware, they may solve the fundamental conflict among accuracy, reliability, and computation time, which appears when searching for the global optimum in complex problems, especially for problems with many local optima [55]. This conflict was previously tackled by means of heterogeneous DGA's (see subsection E in the Appendix) and other different methods. For example, in [55], GA's are hybridized with hill-climbing methods such as the quasi-Newton and Nelder-Mead's simplex. A similar solution is presented in [51], where local search procedures are integrated to DGA's. In [46], a very different model is presented: each subpopulation of a DGA receives information regarding the progress of other subpopulations, and checks its own relative progress. If this is lower, new genetic material is typically introduced by completely reinitializing the subpopulation.

Although GD-RCGA's have arisen as effective and efficient models for dealing with complex problems, they may suffer two problems: *conquest* and *noneffect*. In Section III-A, we describe these problems, and in Sections III-B and III-C, we propose an adequate migration schema and selection mechanism for over-

coming these problems, and for establishing correct coordination between refinement and expansion.

A. The Conquest and Noneffect Problems

One of the drawbacks of DGA's is that the insertion of a new individual from another subpopulation may not be effective. The new individual may be grossly incompatible with that subpopulation, and therefore either be ignored or dominate the subpopulation [40]. This will probably occur when the subpopulations are at different levels of evolution. The arrival of highly evolved migrants from a strong population will result in a higher rate of selection than for local, less-evolved individuals. Thus, the sending population's solution is often imposed on that of the receiver. Conversely, migrants arriving from a less-evolved population are not selected for reproduction, and are wasted [45]. The first problem is called the *conquest problem* [45]. Here, the second one will be called a *noneffect problem*.

The conquest and noneffect problems may appear in a GD-RCGA because the different subpopulations are likely to converge at different rates, and therefore they may differ markedly. The exploitative subpopulations will converge faster than the exploratory ones. Furthermore, the convergence speed will be different on each side since the subpopulations show different exploration or exploitation degrees. In this way, an individual from an exploitative subpopulation (e_i) that is copied into an exploratory one (E_i) is immediately selected more often. If the differential is sufficiently great, or if both the incoming subpopulation and the surrounding area have converged sufficiently, the new individuals are almost always selected. Alternatively, if an individual belonging to an exploratory subpopulation with low fitness is inserted into an exploitative one, it has little chance of being selected for crossover, and is replaced without the population benefiting in any way.

The harmful effects of these problems may be increased due to the short diameter of the cubic topology. Good solutions will spread rapidly to all of the subpopulations, and may quickly take over the population [10].

The use of an *elitist strategy* [15] by the subpopulations is another important factor that may have some influence on rapid convergence. It involves making sure that the best performing chromosome always survives intact from one generation to the next. This is necessary since it is possible that the best chromosome may disappear due to crossover or mutation. The elitist strategy has arisen as a very suitable element for improving the behavior of DGA's [26], [40]. However, in the case of GD-RCGA's, it may have a dangerous effect. The continuous presence of good elements in the exploitative subpopulations will produce an early convergence toward such elements. The small sizes of these subpopulations contributes to the appearance of this problem. These strong elements will reach the exploratory subpopulations, and may produce the conquest problem. Thus, the elitist strategy should be treated with care by the GD-RCGA's.

Next, we describe the migration schema and the selection mechanism chosen for the GD-RCGA's in order to avoid all of

the problems presented above, and to allow the refinement and expansion to be carried through to a suitable conclusion.

B. Migration Schema

DGA behavior is strongly determined by the migration mechanism's action [10], [24]. In most implementations of this mechanism, copies of the individuals who are subject to migration are sent to one or more neighboring subpopulations. Kröger *et al.* [37] call this *immigration*. Additionally, they investigated *emigration*, in which individuals leave their subpopulation, and migrate to exactly one of the neighboring subpopulations. Experimental results indicated that the migration strategy of emigration works best.

We propose an emigration model where migrants are sent only toward immediate neighbors along a dimension of the hypercube, and each subsequent migration takes place along a different dimension of the hypercube. Particularly, the best element of each subpopulation is sent toward the corresponding subpopulation every five generations, as shown in Fig. 6. The sequence of application is from left to right, i.e., first, the refinement migrations, second, the refinement/expansion migrations, third, the expansion migrations, and then, the sequence starts again. The place of an emigrant is taken by an immigrant.

In this way, the best elements (emigrants) do not affect the same subpopulation for a long time, which would probably occur using the elitist strategy, and therefore the conquest is more difficult for them. Furthermore, these good elements may undergo refinement or expansion after being included in the destination subpopulations.

Finally, we point out that with this migration schema, a global elitist strategy persists since the best element of all subpopulations is never lost, although it is moved from one subpopulation to another.

C. Selection Mechanism

The selection mechanism is an important responsibility for the diversity of the population. It may maintain or eliminate diversity, depending on its current *selective pressure*, which represents the degree to which the selection mechanism favors the better individuals. The higher the selection pressure, the greater likelihood that the better individuals are favored, contributing with a large number of copies to the next generation. A larger number of copies for some individuals means fewer copies for the rest of the population. When many individuals do not receive any copies, the result is the loss of diversity. On the other hand, if the selective pressure is low, similar chances to survive are provided, even for worse individuals, and so diversity is maintained.

The crossover operators for RCGA's (Section II) adjust the intervals for the generation of genes, depending on the current population diversity. As we have mentioned, this diversity is limited by the selective pressure of the selection mechanism. In order to iron out the conquest and noneffect problems in the subpopulations of the GD-RCGA's, a suitable combination should be established between the degree of exploration or exploitation of the crossover operators and the degree of selective pressure

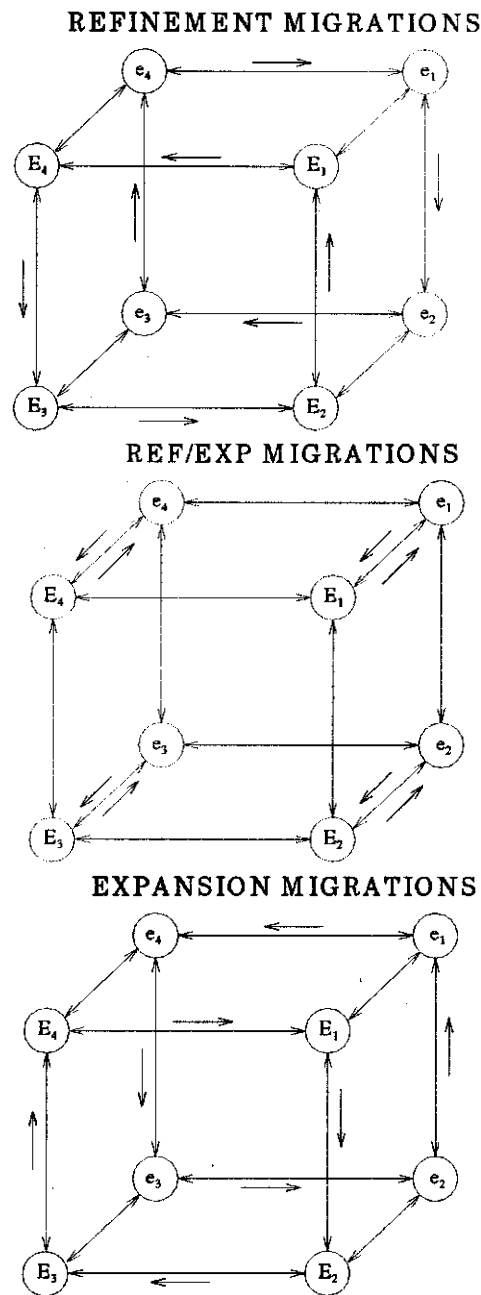


Fig. 6. Three types of migration in a GD-RCGA.

of the selection mechanism. In this subsection, we carry out this task.

A selection mechanism that seems particularly interesting for GD-RCGA's is *linear ranking selection* [5] since the selective pressure produced by it may be easily adjusted by means of varying an associated control parameter. In Section III-C-1, we describe this selection mechanism, and in Section III-C-2, we assign a different selective pressure degree to every subpopulation of the GD-RCGA's.

Finally, we should point out that other authors have built mechanisms for improving the GA that are based on the interactions between the crossover operator and the selection mechanism. In [19], for example, a GA called CHC is proposed which

combines a disruptive crossover operator with a conservative selection strategy (which keeps the N best elements appearing so far). In [30], a fuzzy logic controller is used for tuning the population diversity in a suitable way, which complements the role of the selection mechanism, i.e., either maintaining or eliminating diversity, with the role of the crossover operator, i.e., either creating (exploring) or using (exploiting) diversity.

1) *Linear Ranking Selection*: In linear ranking selection, the chromosomes are sorted in order of raw fitness, and then the selection probability of each chromosome C_i is computed according to its rank $\text{rank}(C_i)$ [with $\text{rank}(C_{\text{best}}) = 1$] by using the following nonincreasing assignment function:

$$p_s(C_i) = \frac{1}{N} \cdot \left(\eta_{\max} - (\eta_{\max} - \eta_{\min}) \cdot \frac{\text{rank}(C_i) - 1}{N - 1} \right)$$

where N is the population size and $\eta_{\min} \in [0, 1]$ specifies the expected number of copies for the worst chromosome (the best one has $\eta_{\max} = 2 - \eta_{\min}$ expected copies). The selective pressure of linear ranking selection is determined by η_{\min} . If η_{\min} is low, high pressure is achieved, whereas if it is high, the pressure is low.

With this selection mechanism, every individual receives an expected number of copies that depends on its rank, independent of the magnitude of its fitness. This may help prevent premature convergence by preventing super migrants from taking over the subpopulations within a few generations (conquest problem), and avoid having inferior migrants fail to have a chance to take part in the next generations (noneffect problem).

Linear ranking will go with *stochastic universal sampling* [6]. This procedure guarantees that the number of copies of any chromosome is bounded by the floor and ceiling of its expected number of copies.

2) *Assignment of Selective Pressure Degrees*: We have assigned to the subpopulations of GD-RCGA's the η_{\min} values shown in Table II.

Table II shows that the more exploratory an E_i subpopulation is, the higher the selective pressure it will undergo. Accordingly, we may comment on the following aspects.

- The most exploratory subpopulations will follow the idea stated in [19], i.e., to put together a disruptive crossover operator and a conservative selection strategy. The main goal of this strategy is to “filter” the high diversity by means of a high selective pressure.
- Although selective pressure is high in these subpopulations, they do not run the risk of being conquered because the constant generation of diversity prevents any type of convergence.
- The less exploratory subpopulations lose selective pressure, and so possible conquerors do not have many advantages against their resident chromosomes.

Alternatively, Table II shows that the more exploitative an e_i subpopulation is, the less selective pressure it will undergo. This allows emigrants sent from exploratory subpopulations to have a chance of surviving in higher exploitative subpopulations, and the noneffect problem is eradicated.

Now, we need to reflect about an important question. From the description in this subsection, it seems that each subpopu-

TABLE II
 η_{\min} VALUES FOR EACH SUBPOPULATION

Crossover Exploitation				Crossover Exploration			
+	←	−		−	→	+	
e_4	e_3	e_2	e_1	E_1	E_2	E_3	E_4
0.8	0.7	0.6	0.5	0.3	0.2	0.1	0.0

lation is “balanced” by combining exploratory crossovers with high-intensity selection or exploitative crossovers with low-intensity selection. Then, *what happens to the gradualism?* since it seems that the heterogeneous nature of GD-RCGA's cancels out. This means that the effects produced in E_i and e_i would be the same, and, when migration occurs, it is likely that they are at about the same stage in the search. However, this does not happen. With the distribution proposed for the η_{\min} values and the crossover configuration chosen, a wide spectrum of different combinations of the possible crossover operator's effects is obtained (generation or use of diversity) and the ones in the selection mechanism (the maintenance or elimination of diversity). In this way, in E_i ($i = 1, \dots, 4$), diversity is created by exploratory crossover operators, and it is filtered by high-intensity selection, whereas in e_i ($i = 1, \dots, 4$), diversity is kept by low-intensity selection, and it used by exploitative crossover operators. Therefore, the stages of E_i and e_i in the search will be different. Furthermore, since these facts will occur even at different degrees, the heterogeneous nature (with its implicit gradualism) of GD-RCGA's does not cancel out.

IV. EXPERIMENTS

Minimization experiments on the test suite, described in Section IV-A, were carried out in order to determine the performance of three GD-RCGA's based on the crossover operators presented in Section II. In Section IV-B, we describe the performance measures used. In Section IV-C, we propose the GD-RCGA based on FCB-crossover operators, and we compare its results with the ones of equivalent sequential versions and other implementations of homogeneous distributed RCGA's; in Section IV-D, the same is done for the GD-RCGA based on BLX- α ; and in Section IV-E, for the one based on extended fuzzy recombination. Then, in Section IV-F, we study, from an empirical point of view, the gradualism associated with GD-RCGA's and the effectiveness of the refinement and expansion. On the basis of this study, in Section IV-G, we propose a restart operator for GD-RCGA's. Finally in Section IV-H, we compare the best GD-RCGA's found in the previous subsections with other mechanisms proposed in the GA literature for avoiding the premature convergence problem.

A. Test Suite

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

1) *Test Functions*: We have considered six frequently used test functions: the *sphere* model (f_{sph}) [15], [59], *generalized Rosenbrock's* function (f_{Ros}) [15], *Schwefel's problem 1.2* (f_{Sch}) [59], *generalized Rastrigin's* function (f_{Ras}) [3], [64],

$$\begin{array}{l}
 \mathbf{f}_{Sph} \\
 f_{Sph}(\vec{x}) = \sum_{i=1}^n x_i^2 \\
 -5.12 \leq x_i \leq 5.12 \\
 f_{Sph}(x^*) = 0
 \end{array}
 \qquad
 \begin{array}{l}
 \mathbf{f}_{Ros} \\
 f_{Ros}(\vec{x}) = \sum_{i=1}^{n-1} (100 \cdot (x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \\
 -5.12 \leq x_i \leq 5.12 \\
 f_{Ros}(x^*) = 0
 \end{array}$$

$$\begin{array}{l}
 \mathbf{f}_{Sch} \\
 f_{Sch}(\vec{x}) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2 \\
 -65.536 \leq x_i \leq 65.536 \\
 f_{Sch}(x^*) = 0
 \end{array}
 \qquad
 \begin{array}{l}
 \mathbf{f}_{Ras} \\
 f_{Ras}(\vec{x}) = a \cdot n + \sum_{i=1}^n x_i^2 - a \cdot \cos(\omega \cdot x_i) \\
 a = 10, \omega = 2\pi \\
 -5.12 \leq x_i \leq 5.12 \\
 f_{Ras}(x^*) = 0
 \end{array}$$

$$\begin{array}{l}
 \mathbf{f}_{Gri} \\
 f_{Gri}(\vec{x}) = \frac{1}{d} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \\
 d = 4000 \\
 -600.0 \leq x_i \leq 600.0 \\
 f_{Gri}(x^*) = 0
 \end{array}
 \qquad
 \begin{array}{l}
 \mathbf{ef}_{10} \\
 ef_{10}(\vec{x}) = f_{10}(x_1, x_2) + \dots + f_{10}(x_{i-1}, x_i) \dots + f_{10}(x_n, x_1) \\
 f_{10}(x, y) = (x^2 + y^2)^{0.25} \cdot [\sin^2(50 \cdot (x^2 + y^2)^{0.1}) + 1] \\
 x, y \in (-100, 100] \\
 f_{ef_{10}}(x^*) = 0
 \end{array}$$

Fig. 7. Test functions.

Griewangk's function (f_{Gri}) [28], and *expansion of* f_{10} (ef_{10}) [69]. Fig. 7 shows their formulation. The dimension of the search space is 10 for ef_{10} and 25 for the remaining test functions.

- f_{sph} is a continuous, strictly convex, and unimodal function.
- f_{Ros} is a continuous and unimodal function, with the optimum located in a steep parabolic valley with a flat bottom. This feature will probably cause slow progress in many algorithms since they must continually change their search direction to reach the optimum. This function has been considered by some authors to be a real challenge for any continuous function optimization program [57]. A great part of its difficulty lies in the fact that there are nonlinear interactions between the variables, i.e., it is *nonseparable* [68].
- f_{Sch} is a continuous and unimodal function. Its difficulty concerns the fact that searching along the coordinate axes only gives a poor rate of convergence since the gradient of f_{Sch} is not oriented along the axes. It presents similar difficulties to f_{Ros} , but its valley is much narrower.
- f_{Ras} is a scalable, continuous, and multimodal function, which is made from f_{sph} by modulating it with $a \cdot \cos(\omega \cdot x_i)$.
- f_{Gri} is a continuous and multimodal function. This function is difficult to optimize because it is *nonseparable* [51], and the search algorithm has to climb a hill to reach the next valley. Nevertheless, one undesirable property exhibited is that it becomes easier as the dimensionality is increased [68].
- f_{10} is a function that has nonlinear interactions between two variables. Its expanded version ef_{10} is built in such a way that it induces nonlinear interaction across multiple variables. It is *nonseparable* as well.

A GA does not need too much diversity to reach the global optimum of f_{sph} since there is only one optimum which could be easily accessed. Alternatively, for multimodal functions

(f_{Ras} , f_{Gri} , and ef_{10}), the diversity is fundamental for finding a way to lead toward the global optimum. Also, in the case of f_{Ros} and f_{Sch} , diversity can help to find solutions close to the parabolic valley, and so avoid slow progress.

2) *Real-World Problems:* We have chosen the following three real-world problems which, in order to be solved, are translated to optimization problems of parameters with variables on continuous domains: *systems of linear equations* [22], *frequency modulation sounds parameter identification problem* [65], and *polynomial fitting problem* [60]. They are described below.

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

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

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

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

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

b) *Frequency modulation sounds parameter identification problem*: The problem is to specify six parameters $a_1, w_1, a_2, w_2, a_3, w_3$ of the frequency modulation sound model represented by

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

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

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

where the model data are given by the following equation:

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

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

c) *Polynomial fitting problem*: This problem lies in finding the coefficients of the following polynomial in z :

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

such that

$$P(z) \in [-1, 1], \quad \text{for } z \in [-1, 1], \quad \text{and} \\ P(1.2) \geq T_{2k}(1.2) \quad \text{and} \quad P(-1.2) \geq T_{2k}(-1.2)$$

where $T_{2k}(z)$ is a Chebyshev 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 Chebyshev 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 in order to transform the constraints of this problem into an objective function to be minimized, called P_{Cheb} . We consider that $C = (c_0, \dots, c_8)$ is the solution to be evaluated and $P_C(z) = \sum_{j=0}^8 c_j \times z^j$.

```

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

```

Return R ;

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

B. Performance Measures

The performance measures listed below have been used in order to study the behavior of GD-RCGA's, and allow their comparison with other genetic algorithms to be made. All of the algorithms have been executed 30 times, each one with 5000 generations.

- *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. If the global optimum has been reached sometimes, this performance will represent the percentage of runs in which this happens.
- *O* performance: average of the final *on-line* measure [15], average of the fitness of all of the elements appearing throughout the GA's execution. On line is considered here as a population diversity measure.

Moreover, a *t test* (at 0.05 level of significance) was applied in order to ascertain if differences in the *A* performance for the GD-RCGA's are significant when compared to the one for the other algorithms in the respective table. The direction of any significant differences is denoted either by

- a plus sign (+) for an improvement in *A* performance, or
- a minus sign (−) for a reduction, or
- an approximate sign (≈) for nonsignificant differences.

The places in the tables of results (Tables IV, VI, VIII–X) where these signs do not appear correspond to the performance values for GD-RCGA's.

C. GD-RCGA Based on FCB-Crossover Operators

A GD-RCGA based on FCB-crossover operators (Section II-A) was implemented with the crossover configuration shown in Table III. It was called GD-FCB. These assignments between subpopulations and FCB-crossover operators allow GD-FCB to produce the gradual effects shown in Fig. 5, thanks to the properties of these operators (which may be observed in Fig. 2).

All GD-RCGA's proposed in this paper use 20 individuals per subpopulation [7]. The mutation operator applied is nonuniform mutation [47]. This operator has been used widely, reporting good results [34], [52]. The probability of updating a chromosome by mutation (p_u) is 0.125, and the crossover probability (p_c) is 0.6.

Along with GD-FCB, we have executed algorithms belonging to two families of sequential RCGA's, R-S2 and R-S4, and to one family of homogeneous distributed RCGA's, D-S4.

- The algorithms in the R-S2 family are R-S2-Log, -Ham, -Alg, and -Ein. They apply the corresponding type of FCB-crossover operators following strategy *ST2* presented in [32]. For each pair of chromosomes from the total population that undergoes crossover, four offspring are generated, the result of applying two exploratory

TABLE III
CROSSOVER CONFIGURATION FOR GD-FCB

Rear Side	M-crossover	Front Side	F- and S-crossover
e_1	Hamacher	E_1	Logical
e_2	Algebraic	E_2	Hamacher
e_3	Einstein	E_3	Algebraic
e_4	Logical	E_4	Einstein

crossover operators and two exploitative ones to them. The two most promising offspring of the four replace their parents in the population. The population size for these algorithms is set at 80, instead of 160 (total size of the GD-RCGA's) since they need four evaluations for each crossover event.

- The R-S4 family is composed of R-S4-Log, -Ham, -Alg, and -Ein. These algorithms use the FCB-crossover operators using the strategy *ST4* proposed in [32]. For each pair of chromosomes from a total of $1/2 \cdot p_c \cdot N$, four offspring are generated, the result of applying two exploratory crossover operators, an exploitative one, and an operator with “relaxed” exploitation, which puts together the two properties. All four offspring will form part of the population in such a way that two of them substitute their parents, and the other two substitute two chromosomes belonging to the remaining 1/2 of the population that should undergo crossover. The population size is set at 160.
- The algorithms in D-S4, D-S4-Log, -Ham, -Alg, and -Ein are homogeneously distributed versions of the corresponding ones in R-S4. They use a cubic topology with a subpopulation size of 20 individuals, and the migration scheme is the same as the one for GD-RCGA's. These algorithms are good reference points for comparing the effectiveness of the GD-RCGA structure since elements generated using a wide spectrum of crossover operators are included in the subpopulations at the same time, just as GD-RCGA's do.

Linear ranking ($\eta_{\min} = 0.75$), stochastic universal sampling, and an elitist strategy were assumed for RCGA's and homogeneous distributed RCGA's. p_c and p_u are the same as the ones for the GD-RCGA's.

1) *Results:* Table IV shows the results obtained. In general, GD-FCB returns better *A* and *B* results than R-S2-Ham, R-S2-Alg, and R-S2-Ein (see *t*-test results). Furthermore, the *O* measure is much greater. This means that the diversity level of GD-FCB, produced by its exploratory side, was higher, but also that the convergence, introduced by its exploitative side, was effective. Thus, reliability and accuracy were improved simultaneously. Alternatively, R-S2-Log provides better solutions than GD-FCB for most test functions, except for f_{Ros} and f_{Sch} . It has a similar performance in real-world problems. R-S2-Log shows a very good convergence level (see the low *O* measure) due to: 1) the *ST2* strategy is very exploitative since it chooses the two best elements from a total of four [32], and 2) the use of the logical FCB crossovers increases this effect since they do not produce any diversity. However, this fact induces a negative effect on f_{Ros} and f_{Sch} since they are complex, and high diversity levels are needed in order to obtain reliability for them. In these cases, the diversity of GD-FCB helped to

achieve better results than R-S2-Log. In particular, it achieved very good *A* and *B* results for f_{Ros} : 9 and $3e-5$, respectively.

GD-FCB do better than algorithms in R-S4. These algorithms show a high level of exploration (see the high *O* measure). This is due to the fact that they are based on the *ST4* strategy, as was indicated in [32]. Convergence may not be carried out in a suitable way. GD-FCB generates too much diversity as well (compare the *O* measure of this algorithm with the one for the algorithms in R-S4); however, its exploitative side allows good convergence to be produced, and so the best elements are found.

Comparison of the GD-FCB algorithm and the D-S4 family allows the behavior of the former to be studied in detail. We observe that the *O* measure associated with GD-FCB shows average values. This is reasonable since it comprises the main properties of all algorithms in D-S4. Also, it may be seen that, in general, its *A* and *B* results are better than the ones for the algorithms in D-S4.

To sum up, we may underline that GD-FCB has allowed reliability and accuracy to be improved simultaneously.

- The exploratory side has produced suitable diversity levels for finding promising regions in the search space, which becomes very useful for the case of the complex functions.
- The exploitative side has generated a suitable local tuning for reaching good final approximations.

D. GD-RCGA Based on BLX- α Crossover Operator

With regard to the properties of BLX- α (Section II-B), we have built a GD-RCGA based on this operator, called GD-BLX, with the α values for each subpopulation shown in Table V. With these assignments, GD-BLX may produce the gradual effects shown in Fig. 5.

GD-BLX is compared with two algorithms: a sequential one, R-BLX, and a homogeneous distributed RCGA, D-BLX. Both are based on the BLX- α crossover with $\alpha = 0.5$. This value is chosen from [34], where experiments with several values of α are tried, being the most effective one. The results for the three algorithms are found in Table VI.

In general, GD-BLX obtains the best *A* and *B* results for all functions, except for P_{ste} and P_{Cheb} . The results for f_{sph} are very approximate, which shows that the exploitation of GD-BLX is highly effective. Exploration is useful as well, as indicated by the good result for the complex f_{Sch} , for the multimodal ef_{10} , f_{Ras} , and f_{Gri} (the global optimum of f_{Ras} and f_{Gri} was found in 100 and 60% of the runs, respectively), and for the P_{fms} real-world problem (with a 66.7% percentage in reaching the global optimum).

E. GD-RCGA Based on Extended Fuzzy Recombination

We have implemented a GD-RCGA using the extended fuzzy recombination operator (Section II-C), called GD-EFR. In order to produce the adequate gradual effects (Fig. 5), we have assigned to each subpopulation the *d* values shown in Table VII.

We run a sequential RCGA, called R-EFR, that uses the fuzzy recombination operator proposed in [66] with $d = 0.5$. This value seemed a good choice for a large class of functions. The operator is equivalent to extended fuzzy recombination with $d =$

TABLE IV
RESULTS FOR R-S2, R-S4, D-S4, AND GD-FCB

ALG.	f_{Sph}				f_{Ros}				f_{Sch}			
	A	SD	B	O	A	SD	B	O	A	SD	B	O
R-S2-Log	2e-16(-)	1e-16	3e-17	6e-1	2e1(+)	1e-1	2e1	6e2	1e1(+)	6e0	3e0	1e3
R-S2-Ham	3e-11(+)	2e-11	6e-12	2e0	2e1(+)	6e-1	2e1	9e2	6e1(+)	4e1	6e0	7e3
R-S2-Alg	3e-11(+)	2e-11	4e-12	1e0	2e1(+)	9e-2	2e1	8e2	3e1(+)	4e1	1e0	5e3
R-S2-Ein	3e-11(+)	2e-11	1e-11	1e0	2e1(+)	1e-1	2e1	8e2	2e1(+)	1e1	5e0	3e3
R-S4-Log	2e-10(+)	7e-11	3e-11	1e0	2e1(+)	1e-1	2e1	1e3	4e1(+)	1e1	1e1	5e3
R-S4-Ham	6e-2(+)	3e-2	2e-2	9e1	2e1(+)	1e1	4e0	7e4	2e2(+)	8e1	8e1	3e6
R-S4-Alg	2e0(+)	1e0	3e-1	6e2	4e2(+)	2e2	1e2	2e6	3e3(+)	1e3	9e2	4e5
R-S4-Ein	2e0(+)	1e0	6e-1	6e2	4e2(+)	2e2	7e1	2e6	4e3(+)	2e3	2e3	4e5
D-S4-Log	2e-12(+)	8e-13	5e-13	9e-1	2e1(+)	3e-1	2e1	8e2	3e0(-)	1e0	8e-1	4e3
D-S4-Ham	6e-4(+)	3e-4	1e-4	1e2	6e0(≈)	1e1	6e-2	8e4	3e2(+)	1e2	8e1	3e6
D-S4-Alg	1e-1(+)	9e-2	2e-2	3e2	1e2(+)	9e1	3e1	5e5	4e3(+)	1e3	1e3	2e6
D-S4-Ein	1e-1(+)	6e-2	2e-2	4e2	1e2(+)	1e2	3e1	8e5	4e3(+)	2e3	1e3	2e6
GD-FCB	2e-13	1e-13	4e-14	7e1	9e0	1e1	3e-5	8e4	4e0	3e0	6e-1	1e6
ALG.	f_{Ros}				f_{Gri}				e_{f10}			
R-S2-Log	5e-14(-)	3e-14	6.7%	6e0	8e-3(-)	1e-2	3e-14	2e0	5e-5(-)	7e-5	7e-6	5e0
R-S2-Ham	2e1(+)	8e0	6e0	1e2	1e-2(≈)	2e-2	1e-8	6e0	1e-2(+)	5e-3	4e-3	2e1
R-S2-Alg	7e0(+)	2e0	2e0	4e1	2e-2(≈)	2e-2	2e-8	5e0	1e-2(+)	6e-3	5e-3	1e1
R-S2-Ein	1e1(+)	3e0	7e0	4e1	2e-2(≈)	2e-2	1e-8	5e0	1e-2(+)	4e-3	4e-3	1e1
R-S4-Log	4e0(+)	3e0	2e-8	4e1	1e-2(≈)	2e-2	8e-8	4e0	2e-2(+)	9e-3	8e-3	1e1
R-S4-Ham	4e1(+)	1e1	2e1	4e2	1e0(+)	1e-1	1e0	3e2	7e0(+)	3e0	2e0	1e2
R-S4-Alg	5e1(+)	1e1	2e1	7e2	7e0(+)	4e0	2e0	2e3	4e0(+)	2e0	8e-1	1e2
R-S4-Ein	4e1(+)	1e1	1e1	7e2	1e1(+)	6e0	2e0	2e3	5e0(+)	3e0	1e0	1e2
D-S4-Log	7e-2(≈)	2e-1	1e-10	2e1	1e-2(≈)	1e-2	5e-10	4e0	3e-3(+)	1e-3	1e-3	1e1
D-S4-Ham	2e1(+)	4e0	1e1	4e2	4e-1(+)	1e-1	1e-1	3e2	2e0(+)	7e-1	9e-1	1e2
D-S4-Alg	3e1(+)	7e0	2e1	6e2	1e0(+)	2e-1	1e0	9e2	6e-1(+)	8e-1	1e-1	1e2
D-S4-Ein	3e1(+)	7e0	2e1	6e2	1e0(+)	4e-1	1e0	1e3	2e-1(+)	1e-1	1e-1	1e2
GD-FCB	4e-11	2e-11	9e-12	2e2	2e-2	2e-2	4e-11	2e2	2e-3	2e-3	8e-4	5e1
ALG.	P_{ste}				P_{fms}				P_{Cheb}			
R-S2-Log	3e1(≈)	2e1	1e0	2e2	1e1(≈)	6e0	4e-16	2e1	2e2(≈)	1e2	5e1	1e5
R-S2-Ham	3e1(≈)	3e1	2e0	5e2	2e1(+)	4e0	1e1	3e1	5e2(+)	6e2	3e0	3e5
R-S2-Alg	2e1(-)	1e1	2e0	4e2	1e1(+)	8e0	4e-2	3e1	1e2(-)	1e2	2e1	2e5
R-S2-Ein	1e1(-)	1e1	5e-1	4e2	2e1(+)	6e0	6e-2	3e1	1e2(-)	8e1	1e1	2e5
R-S4-Log	2e1(-)	1e1	2e0	5e2	2e1(+)	7e0	7e-2	4e1	2e2(≈)	1e2	3e1	3e5
R-S4-Ham	6e1(+)	2e1	1e1	2e4	2e1(+)	6e0	2e-2	3e2	8e2(+)	6e2	2e2	3e7
R-S4-Alg	6e2(+)	2e2	2e2	8e3	2e1(+)	4e0	1e1	7e2	4e4(+)	4e4	2e3	2e7
R-S4-Ein	7e2(+)	3e2	2e2	9e3	2e1(+)	6e0	1e-1	9e2	3e4(+)	2e4	4e3	3e7
D-S4-Log	3e1(-)	2e1	1e0	5e2	1e1(≈)	6e0	1e-15	3e1	2e2(≈)	1e2	2e1	2e5
D-S4-Ham	8e1(+)	4e1	1e1	2e4	1e1(≈)	5e0	2e-5	3e2	1e3(+)	7e2	1e2	4e7
D-S4-Alg	5e2(+)	2e2	1e2	1e4	2e1(+)	5e0	2e-2	8e2	2e4(+)	1e4	2e3	3e7
D-S4-Ein	6e2(+)	3e2	6e1	1e4	2e1(+)	6e0	2e-3	1e3	2e4(+)	2e4	6e2	3e7
GD-FCB	4e1	2e1	3e0	6e3	1e1	6e0	7e-16	2e2	2e2	2e2	4e1	1e7

TABLE V
 α VALUES FOR GD-BLX

e_4	e_3	e_2	e_1	E_1	E_2	E_3	E_4
0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8

0.5. A distributed version of this algorithm was also executed, which was called D-EFR. Table VIII contains the results.

For most problems, GD-EFR improves the A and B results of the other two algorithms. Only for P_{ste} and P_{fms} do the t -test results indicate that GD-EFR has a similar A performance to R-EFR and D-EFR, and for f_{Gri} , a worse one than R-EFR. However, it should be emphasized that, for two of these problems, f_{Gri} and P_{fms} , GD-EFR found the global optimum in 53.3 and 43.3% of the runs, respectively, whereas none of the remaining algorithms reached the global optimum of these prob-

lems. These results show the profitable effects of the gradual multiresolution, refinement, and expansion in GD-EFR.

Finally, comparing the results of GD-EFR and GD-BLX with the ones of the another GD-RCGA proposed, GD-FCB, we may observe that they outperform it for most functions. Furthermore, we may consider that these algorithms achieve a *robust* operation, in the sense that they obtain a significant performance for each one of the test functions, which have different difficulties. Hence, BLX- α and extended fuzzy recombination arise as suitable crossover operators for building GD-RCGA's.

F. Study of the Gradualism, Refinement, and Expansion in GD-RCGA's

In this section, first we study the effects of the gradualism in GD-RCGA's, investigating the way in which the subpopula-

TABLE VI
RESULTS FOR R-BLX, D-BLX, AND GD-BLX

ALG.	f_{Sph}				f_{Ros}				f_{Sch}			
	A	SD	B	O	A	SD	B	O	A	SD	B	O
R-BLX	9e-7 (+)	6e-7	2e-7	1e1	3e1 (+)	3e1	2e1	1e4	1e3 (+)	3e2	6e2	1e5
D-BLX	1e-14 (+)	7e-15	5e-15	3e0	2e1 (+)	1e1	2e1	2e3	2e-3 (+)	2e-3	2e-4	1e4
GD-BLX	8e-53	4e-52	8e-58	9e-1	2e1	4e-1	2e1	9e2	2e-6	3e-6	7e-8	2e3

ALG.	f_{Ras}				f_{Gri}				ef_{10}			
	A	SD	B	O	A	SD	B	O	A	SD	B	O
R-BLX	4e1 (+)	9e0	2e1	3e2	6e-1 (+)	1e-1	4e-1	4e1	2e-2 (+)	5e-3	9e-3	3e1
D-BLX	1e1 (+)	3e0	7e0	9e1	2e-2 (+)	2e-2	4e-12	1e1	1e-3 (+)	3e-4	6e-4	2e1
GD-BLX	0e0	0e0	100%	1e1	5e-3	7e-3	60%	3e0	6e-36	3e-35	5e-48	6e0

ALG.	P_{ste}				P_{fms}				P_{Cheb}			
	A	SD	B	O	A	SD	B	O	A	SD	B	O
R-BLX	8e0 (-)	6e0	1e0	2e3	1e1 (+)	4e0	4e-1	4e1	2e2 (~)	2e2	2e0	1e6
D-BLX	9e0 (-)	8e0	8e-1	9e2	6e0 (~)	6e0	1e-18	3e1	3e2 (~)	2e2	2e0	5e5
GD-BLX	3e1	3e1	2e0	3e2	3e0	5e0	66.7%	2e1	2e2	2e2	1e1	2e5

TABLE VII
d VALUES FOR GD-EFR

e_4	e_3	e_2	e_1	E_1	E_2	E_3	E_4
0.0	0.1	0.2	0.4	0.6	0.8	0.9	1.0

tions evolve during the run (Section IV-F-1), then we attempt to detect the effectiveness of the refinement and expansion, finding which subpopulations generated the best elements over time (Section IV-F-2).

1) *Gradualism*: Here, we investigate the way in which the subpopulations of GD-RCGA's evolve during the run. In particular, we are interested in observing whether the evolution in the subpopulations is similar (i.e., one of them dominated the others) or whether each subpopulation follows a different search line.

Figs. 8 and 9 were introduced in order to do this. Fig. 8 outlines the averages of the objective function of three subpopulations of GD-EFR (e_3 , e_4 , and E_4) during the first 1000 generations on f_{Ras} . Fig. 9 shows the same for the case of D-EFR (homogeneous DGA based on the extended fuzzy recombination with $d = 0.5$). We may observe that there is a notable difference between the evolution in the subpopulations of D-EFR and GD-EFR. The subpopulations of D-EFR show similar evolution levels. They seem very influenced by each other, which probably occurs since they suffer the conquest problem, leading all subpopulations to have the same search biases. Alternatively, the subpopulations of GD-EFR have different evolution levels. The gradualism associated with GD-RCGA's has allowed this effect to be produced, avoiding the conquest problem.

2) *Effectiveness of the Refinement and Expansion*: Although GD-EFR shows signs of gradualism, we have to check whether this one, along with migrations, causes the subpopulations to produce better elements, i.e., we need to study if the refinement and expansion are really effective.

Fig. 10 was included for this purpose. It shows the subpopulations of GD-EFR that generate the best elements during the first 2500 generations on f_{Ras} . For each generation where a best element is found, a mark is printed in the subpopulation where this occurs. We see that most subpopulations contribute during continuous periods of time with the best elements. This is made

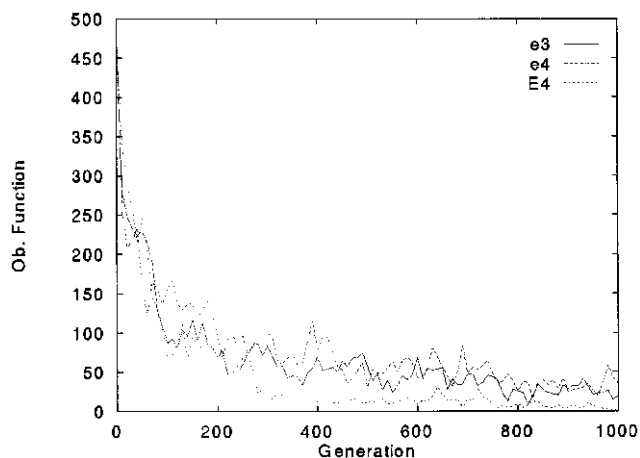


Fig. 8. GD-EFR for f_{Ras} .

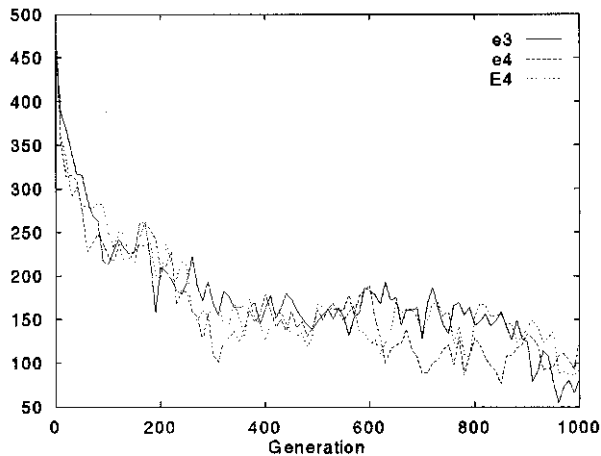


Fig. 9. D-EFR for f_{Ras} .

in a parallel way with the other subpopulations. *They collaborate with each other for generating the best elements by means of the refinement and expansion of the elements brought by the migrations from other subpopulations.* This situation may be compared with the one in Fig. 13, which shows the same information for the case of GD-EFR without migrations. Here, the generation of the best elements is produced in the same

TABLE VIII
RESULTS FOR R-EFR, D-EFR, AND GD-EFR

ALG.	f_{Sph}				f_{Ros}				f_{Sch}			
	A	SD	B	O	A	SD	B	O	A	SD	B	O
R-EFR	6e-14 (+)	3e-14	2e-14	4e0	2e1 (+)	2e-1	2e1	3e3	2e0 (+)	1e0	7e-1	3e4
D-EFR	1e-16 (+)	5e-17	6e-17	2e0	2e1 (+)	1e1	2e1	2e3	2e-4 (+)	2e-4	3e-5	7e3
GD-EFR	2e-47	5e-17	8e-51	8e-1	2e1	6e-1	1e1	8e2	4e-6	3e-6	2e-7	2e3
ALG.	f_{Ras}				f_{Gri}				ef_{10}			
R-EFR	1e1 (+)	4e0	6e0	2e2	4e-11 (-)	2e-11	1e-11	1e1	3e-3 (+)	4e-4	2e-3	2e1
D-EFR	5e0 (+)	2e0	3.3%	5e1	2e-2 (+)	2e-2	4e-14	8e0	4e-4 (+)	6e-5	3e-4	2e1
GD-EFR	0e0	0e0	100%	1e1	7e-3	8e-3	53.3%	3e0	9e-35	5e-34	6e-47	6e0
ALG.	P_{ste}				P_{fms}				P_{Cheb}			
R-EFR	1e1 (~)	1e1	4e0	1e3	9e0 (~)	6e0	4e-17	4e1	4e2 (+)	4e2	7e0	6e5
D-EFR	2e1 (~)	2e1	1e0	7e2	4e0 (~)	6e0	1e-19	3e1	5e2 (+)	4e2	5e0	4e5
GD-EFR	2e1	2e1	9e-1	3e2	6e0	7e0	43.3%	2e1	2e2	1e2	1e1	1e5

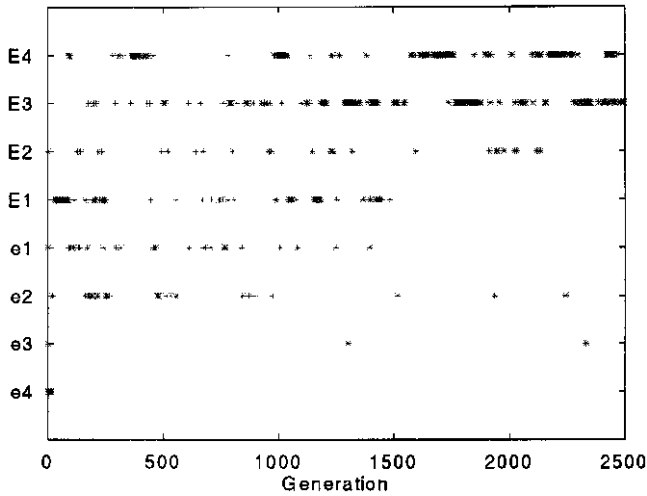


Fig. 10. GD-EFR for f_{Ras} .

subpopulations over large periods of time. No best elements are created by refining or expanding elements coming from other subpopulations.

Finally, we should consider the situation in Fig. 11, which is for the case of D-EFR on f_{Ras} . The continuity of generating the best elements is missing in each subpopulation. The best elements are produced in the subpopulations in an isolated way. Since the evolution in the subpopulations of D-EFR is altered frequently by the migrations with the other subpopulations, there is not a continuous line of generation of the best elements. Migrations are not effective; they do not help to create better elements, but break possible defined search lines.

We finish this section by studying the performance of refinement and expansion in the different subpopulations when GD-RCGA's are applied on problems with different features. In order to do this, we included Fig. 12, which has the same information as Fig. 10, but for the case of GD-EFR on f_{Sch} . f_{Ras} and f_{Sch} are very different: f_{Ras} is an easy multimodal function, whereas f_{Sch} is a complex unimodal one. Comparing these two figures, we may observe that the most fruitful subpopulations (the ones generating the best elements) for f_{Ras} are different from the ones for f_{Sch} .

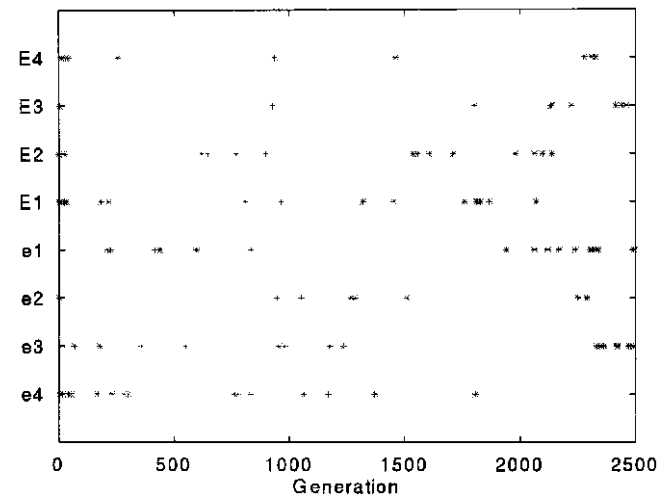


Fig. 11. D-EFR for f_{Ras} .

- For f_{Ras} , the subpopulations applying exploratory crossover operators E_4, \dots, E_1 are the most effective. For this multimodal function, the generation of diversity and its filtering (through a high selective pressure) is useful for finding a way to lead toward the global optimum.
- For f_{Sch} , the most prolific subpopulations are E_1, e_1 , and e_2 . A medium selective pressure and crossover operators with low exploitation properties are adequate for obtaining better values for this function each time. The role of other subpopulations with exploratory crossover operators, such as E_4 and E_3 , is significant as well. For this unimodal function the diversity is not the most determining factor; however, it may help to find good elements because this function is highly complex.

These results show the way in which GD-RCGA's may act suitably on functions with different features. This is possible since they dispose of a wide spectrum of different combinations of crossover operator's exploration/exploitation properties and selective pressure degrees.

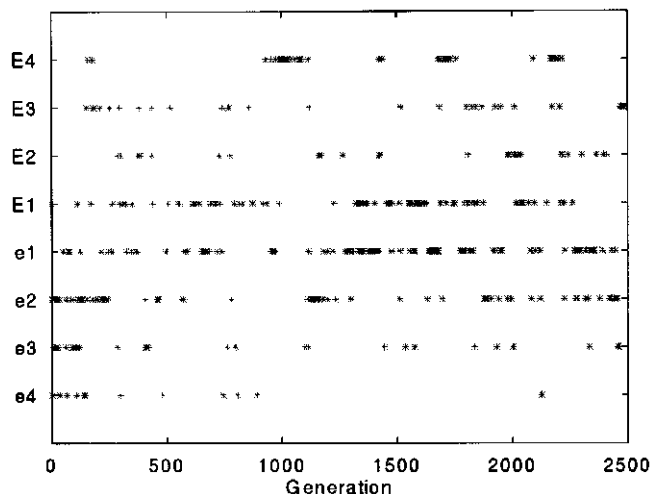


Fig. 12. GD-EFR for f_{Sch} .

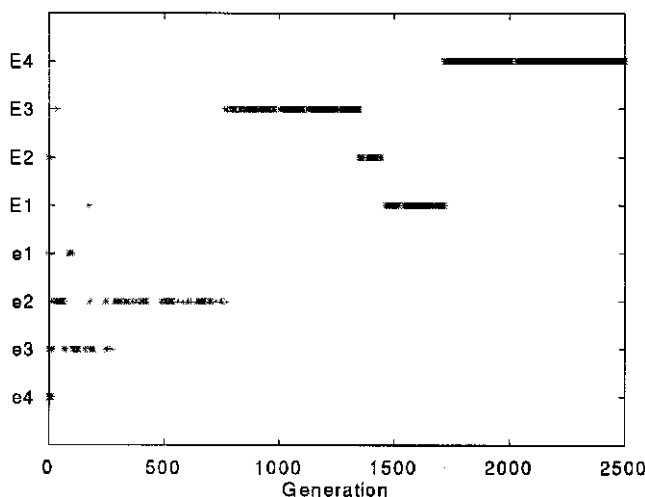


Fig. 13. GD-EFR without migrations for f_{Ras} .

G. A Restart Operator for GD-RCGA's

In the previous subsection, we have seen that an effect produced by an effective operation of refinement and expansion is that most subpopulations contribute during continuous periods of time with the best elements in a parallel way. This may provide some clues about possible situations where refinement and expansion do poorly. In particular, a situation in which the generation of the best elements is located only in one subpopulation during a long time, accompanied by insignificant improvements on the best element, may be an indication of a nonprofitable working of refinement and expansion in the search region being currently handled. Under these circumstances, the resources of the GD-RCGA would be better utilized in restarting the search in a new area with a new population.

In this way, we propose to include the following *restart operator* into GD-RCGA's: if the best elements are being generated in the same subpopulation over the last 50 generations and $(f_b(t_a)/f_b(t_b)) > 0.99$, with $f_b(t_b)$ and $f_b(t_a)$ being the fitness of the best chromosome before and after this time interval, respectively (which represents a low improvement on the best element), then the subpopulations will be reinitialized using

randomly generated individuals. Furthermore, since the nonuniform mutation operator works depending on the current generation t and the total number of generations T , both parameters are replaced by 1 and $T - t$, respectively.

Experiments were carried out for studying the behavior of GD-RCGA's with this restart operator. Table IX shows the results of the two best GD-RCGA's, GD-BLX, and GD-EFR, and the ones of their versions with the restart operator, called GD-BLX^r and GD-EFR^r.

Looking over the results, we may report the following considerations.

- The t -test highlights improvements on the A performance when using the restart operator on the most complex functions, f_{Gri} , P_{ste} , and P_{Cheb} . Furthermore, in the case of P_{fms} and f_{Gri} , the percentage of runs reaching the global optimum increased as well. Since these functions are very complex, GD-RCGA's have a higher probability of being trapped in regions that do not contain the global optimum, finding it difficult to escape from them. However, the restart operator might help GD-RCGA's to do this, giving more opportunities to obtain better elements.
- The performance on the remaining functions (all of the unimodal ones, f_{sph} , f_{Ros} , and f_{Sch} , and the noncomplex multimodal f_{Ras} , and e_{f10}) was found to be insensitive to the incorporation of the restart operator (see the t -test results). This indicates that the conditions for reinitializing the subpopulations were almost never fulfilled, which means that the operation of the refinement and expansion on these functions has been effective along each run.

In summary, the participation of the restart operator allowed the reliability of GD-RCGA's to be improved on complex functions. An important conclusion derived from this fact is that the conditions proposed for firing the restart operator really describe stationary states for GD-RCGA's, which lead to a significant drop in their performance. With the restart operator, GD-RCGA's may recover from these states. Finally, other authors have proposed restart operators for DGA's [46].

H. Comparison of the GD-RCGA's with Other Mechanisms for Dealing with Premature Convergence

In this subsection, we compare the performance of the two best GD-RCGA's, GD-BLX^r and GD-EFR^r, with other mechanisms proposed in the literature for monitoring the population diversity in order to avoid the premature convergence problem. These are the following: *ECO-GA model* [14], *CHC algorithm* [19], *deterministic crowding* [42], and *disruptive selection* [38].

In Sections IV-H-1-IV-H-4, we review these techniques, respectively, and in Section IV-H-5, we compare them with GD-BLX^r and GD-EFR^r.

1) *ECO-GA*: ECO-GA employs a two-dimensional grid having its opposite edges connected together so that each grid element has eight adjacent elements. To begin, the grid is initialized randomly, one population member per node. At each iteration, a grid element i, j is selected at random, and defines a nine-element subpopulation around it. Two chromosomes are selected probabilistically from this subpopulation according to

TABLE IX
RESULTS FOR GD-BLX^r AND GD-EFR^r

ALG.	f_{Sph}				f_{Ros}				f_{Sch}			
	A	SD	B	O	A	SD	B	O	A	SD	B	O
GD-BLX	8e-53 (~)	4e-52	8e-58	9e-1	2e1 (~)	4e-1	2e1	9e2	2e-6 (~)	3e-6	7e-8	2e3
GD-BLX ^r	8e-53	4e-52	8e-58	9e-1	2e1	9e0	1e1	1e3	2e-6	3e-6	7e-8	2e3
ALG.	f_{Ros}				f_{Gri}				ef_{10}			
GD-BLX	0e0 (~)	0e0	100%	1e1	5e-3 (+)	7e-3	60%	3e0	6e-36 (~)	3e-35	5e-48	6e0
GD-BLX ^r	0e0	0e0	100%	2e1	4e-4	2e-3	96.7%	5e0	9e-39	5e-38	3e-47	6e0
ALG.	P_{ste}				P_{fms}				P_{Cheb}			
GD-BLX	3e1 (+)	3e1	2e0	3e2	3e0 (~)	5e0	66.7%	2e1	2e2 (+)	2e2	1e1	2e5
GD-BLX ^r	8e0	5e0	1e0	7e2	1e0	4e0	80%	2e1	6e1	3e1	9e0	5e5
ALG.	f_{Sph}				f_{Ros}				f_{Sch}			
GD-EFR	2e-47 (~)	5e-47	8e-51	8e-1	2e1 (~)	6e-1	1e1	8e2	4e-6 (~)	3e-6	2e-7	2e3
GD-EFR ^r	2e-47	5e-47	8e-51	8e-1	2e1	9e-1	1e1	1e3	3e-5	1e-4	2e-7	2e3
ALG.	f_{Ros}				f_{Gri}				ef_{10}			
GD-EFR	0e0 (~)	0e0	100%	1e1	7e-3 (+)	8e-3	53.3%	3e0	9e-35 (~)	5e-34	6e-47	6e0
GD-EFR ^r	0e0	0e0	100%	1e1	0e0	0e0	100%	4e0	1e-28	7e-28	3e-45	6e0
ALG.	P_{ste}				P_{fms}				P_{Cheb}			
GD-EFR	2e1 (+)	2e1	9e-1	3e2	6e0 (+)	7e0	43.3%	2e1	2e2 (+)	1e2	1e1	1e5
GD-EFR ^r	7e0	6e0	3e-1	6e2	4e-1	2e0	80%	2e1	7e1	7e1	1e1	4e5

their relative fitness. These two individuals undergo crossover and mutation, producing two offspring. After calculating the fitness of the offspring, each one is introduced into the grid by selecting a grid node at random from the nine-grid member environment. Each offspring competes with the individual currently occupying the chosen grid node. This represents an additional selection stage, in which the survival probability of each competitor is proportional to its relative fitness.

Two ECO-GA's were implemented, called ECO-BLX and ECO-EFR, which use BLX- α ($\alpha = 0.5$) and extended fuzzy recombination ($d = 0.5$), respectively. They apply the nonuniform mutation operator, depending on the number of objective function evaluations. A 16×16 grid was considered.

2) *The CHC Algorithm:* During each generation, the CHC algorithm uses a parent population of size N to generate an intermediate population of N individuals, which are randomly paired and used to generate N potential offspring. Then, a survival competition is held, where the best N chromosomes from the parent and offspring populations are selected to form the next generation.

CHC also employs heterogeneous recombination as a method of incest prevention. In order to do this, the real values of the two individuals' parameters are encoded into bit strings using binary reflected Gray coding, and the Hamming distance between the parents is measured. Only those string pairs which differ from each other by some number of bits (mating threshold) are mated. The initial threshold is set at $L/4$, where L is the length of the string ($L = 30 \cdot n$ in the experiments). When no offspring are inserted into the new population, the threshold is reduced by 1.

No mutation is applied during the recombination phase. Instead, when the population converges or the search stops making progress (i.e., the difference threshold has dropped to zero, and no new offspring are being generated which are better than any

members of the parent population), the population is reinitialized. The restart population consists of random individuals, except for one instance of the best individual found so far [22].

Two instances of the CHC algorithm, CHC-BLX, and CHC-EFR, were built using BLX- α ($\alpha = 0.5$) and extended fuzzy recombination ($d = 0.5$), respectively. The population size is 50 chromosomes.

3) *Deterministic Crowding:* Crowding methods attempt to preserve the population diversity during the replacement procedure as follows: new individuals are more likely to replace existing individuals in the parent population that are similar to themselves based on genotypic similarity. They have been used for locating, and preserve multiple local optimum in multimodal functions.

An effective crowding method is *deterministic crowding*. It works by randomly pairing all population elements in each generation. Each pair of parents (P_i, P_j) undergoes crossover in combination with mutation to yield two offspring (O_i, O_j) which compete against the parents for inclusion in the population through the following method of competition:

```

IF [ $d(P_i, O_i) + d(P_j, O_j)$ ]  $\leq$  [ $d(P_i, O_j) + d(P_j, O_i)$ ] THEN
  IF  $f(O_i)$  is better than  $f(P_i)$  THEN replace  $P_i$  with  $O_i$ .
  IF  $f(O_j)$  is better than  $f(P_j)$  THEN replace  $P_j$  with  $O_j$ .
ELSE
  IF  $f(O_i)$  is better than  $f(P_j)$  THEN replace  $P_j$  with  $O_i$ .
  IF  $f(O_j)$  is better than  $f(P_i)$  THEN replace  $P_i$  with  $O_j$ .

```

TABLE X
RESULTS OF THE COMPARISON

ALG.	f_{Sph}				f_{Ros}				f_{Sch}			
	A	SD	B	O	A	SD	B	O	A	SD	B	O
ECO-BLX	1e-31 (+)	2e-31	8e-33	2e0	2e1 (+)	4e-1	2e1	2e3	6e0 (+)	5e0	1e0	6e3
CHC-BLX	9e-32 (+)	2e-32	5e-32	4e0	2e1 (~)	7e-1	2e1	2e3	1e-1 (~)	3e-1	7e-12	1e3
R-DC-BLX	3e-6 (+)	2e-6	2e-7	9e0	1e1 (-)	5e0	4e0	8e3	3e3 (+)	7e2	2e3	1e4
R-DS-BLX	2e0 (+)	1e0	4e-1	8e1	3e2 (+)	1e2	1e2	9e4	2e3 (+)	4e2	1e3	4e5
GD-BLX ^r	8e-53	4e-52	8e-58	9e-1	2e1	9e0	1e1	1e3	2e-6	3e-6	7e-8	2e3
ALG.	f_{Ros}				f_{Gri}				ef_{10}			
ECO-BLX	1e2 (+)	1e1	8e1	3e2	1e0 (+)	3e-2	9e-1	9e0	5e-9 (~)	2e-8	4e-10	8e0
CHC-BLX	0e0 (~)	0e0	100%	7e1	0e0 (~)	0e0	100%	2e0	1e-7 (+)	2e-8	9e-8	8e0
R-DC-BLX	9e0 (+)	1e0	5e0	5e1	1e-2 (+)	1e-2	1e-4	3e1	2e-1 (+)	1e-1	8e-2	2e1
R-DS-BLX	5e1 (+)	1e1	3e1	3e2	6e0 (+)	3e0	2e0	3e2	1e1 (+)	3e0	7e0	9e1
GD-BLX ^r	0e0	0e0	100%	2e1	4e-4	2e-3	96.7%	5e0	9e-39	5e-38	3e-47	6e0
ALG.	P_{ste}				P_{fms}				P_{Cheb}			
ECO-BLX	3e1 (+)	2e1	6e0	3e2	2e1 (+)	2e0	2e1	3e1	1e3 (+)	1e3	8e1	1e5
CHC-BLX	3e0 (-)	2e0	2e-1	2e2	1e-11 (~)	6e-12	1e-12	2e1	1e2 (+)	1e2	3e0	1e5
R-DC-BLX	4e2 (+)	1e2	2e2	2e3	8e0 (+)	5e0	3e-4	3e1	3e3 (+)	1e3	4e2	3e5
R-DS-BLX	3e2 (+)	8e1	1e2	1e4	1e1 (+)	4e0	4e-1	5e2	6e3 (+)	4e3	2e3	2e7
GD-BLX ^r	8e0	5e0	1e0	7e2	1e0	4e0	80%	2e1	6e1	3e1	9e0	5e5
ALG.	f_{Sph}				f_{Ros}				f_{Sch}			
ECO-EFR	2e-37 (+)	3e-37	3e-39	2e0	2e1 (+)	8e-1	2e1	2e3	9e1 (+)	7e1	2e1	6e3
CHC-EFR	8e-32 (+)	2e-32	4e-32	4e0	2e1 (+)	5e-1	2e1	3e3	4e-2 (+)	1e-1	6e-5	3e3
R-DC-EFR	1e-5 (+)	7e-6	6e-7	1e1	1e1 (-)	5e0	3e0	1e4	4e3 (+)	7e2	3e3	1e4
R-DS-EFR	1e0 (+)	8e-1	2e-1	9e1	4e2 (+)	2e2	2e2	1e5	3e3 (+)	6e2	1e3	7e5
GD-EFR ^r	2e-47	5e-47	8e-51	8e-1	2e1	9e-1	1e1	1e3	3e-5	1e-4	2e-7	2e3
ALG.	f_{Ros}				f_{Gri}				ef_{10}			
ECO-EFR	6e0 (+)	2e0	2e0	9e1	2e-4 (~)	1e-3	1e-15	8e0	1e-10 (~)	6e-10	3e-14	6e
CHC-EFR	0e0 (~)	0e0	100%	6e1	0e0 (~)	0e0	100%	3e0	1e-7 (+)	1e-8	1e-7	9e
R-DC-EFR	9e0 (+)	2e0	5e0	6e1	2e-2 (+)	1e-2	4e-4	3e1	4e-1 (+)	2e-1	1e-1	2e1
R-DS-EFR	5e1 (+)	1e1	3e1	4e2	5e0 (+)	2e0	2e0	3e2	1e1 (+)	5e0	6e0	1e2
GD-EFR ^r	0e0	0e0	100%	1e1	0e0	0e0	100%	4e0	1e-28	7e-28	3e-45	6e
ALG.	P_{ste}				P_{fms}				P_{Cheb}			
ECO-EFR	9e1 (+)	6e1	1e1	4e2	1e1 (+)	7e0	2e-11	2e1	2e3 (+)	2e3	8e1	1e5
CHC-EFR	4e0 (-)	3e0	4e-1	3e2	1e-11 (~)	6e-12	1e-12	2e1	5e1 (~)	7e1	2e0	2e5
R-DC-EFR	4e2 (+)	1e2	2e2	2e3	8e0 (+)	6e0	1e-4	3e1	4e3 (+)	2e3	6e2	4e5
R-DS-EFR	3e2 (+)	1e2	2e2	1e4	1e1 (+)	5e0	4e-2	7e2	8e3 (+)	4e3	1e3	3e7
GD-EFR ^r	7e0	6e0	3e-1	6e2	4e-1	2e0	80%	2e1	7e1	7e1	1e1	4e5

Two RCGA's were implemented, R-DC-BLX and R-DC-EFR, which apply deterministic crowding as a replacement strategy, and whose remaining features are the same as the ones of R-BLX (Section IV-D) and R-EFR (Section IV-E), respectively.

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

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

where \bar{f} is the average value of the objective function of the individuals in the population.

We have included the disruptive selection in the R-BLX and R-EFR algorithms for making use of two RCGA's based on this mechanism, R-DS-BLX and R-DS-EFR.

5) *Comparison*: All of these algorithms were executed 30 times. 500 000 evaluations of the objective function were

allowed in each time for the CHC and ECO-GA algorithms. This number is similar to the number of evaluations performed by GD-RCGA's during 5000 generations. The remaining algorithms were executed during 5000 generations. Table X shows the results obtained. The results of GD-BLX^r and GD-EFR^r were included again.

a) *RCGA's Using Disruptive Selection and Deterministic Crowding*: R-DS-BLX, R-DS-EFR, and R-DC-BLX, R-DC-EFR returned low A and B results. These mechanisms lead to a high diversity level during the GA execution, slowing the convergence too much (the O measures of R-DS-BLX and R-DS-EFR are greater than the ones of the remaining algorithms). Only for the complex f_{Ros} was this useful; R-DC-EFR has returned good A and B results for this function.

b) *ECO-GA Algorithms*: ECO-BLX and ECO-EFR do better than the previous algorithms. In these algorithms, the exploitation is achieved by means of the local interactions between adjacent chromosomes, whereas the exploration is possible thanks to the spatial separation of the chromosomes

through the grid (subsection B in the Appendix). The balance between these properties has allowed good A and B values to be returned for the unimodal f_{sph} and for the multimodal ef_{10} . However, it was not suitable for dealing with the remaining functions. An additional profitable feature of the ECO-GA algorithms is that they may be easily implemented on parallel hardware. In fact, they belong to a class of *parallel GA's* called *cellular GA's* (subsection A in the Appendix).

c) CHC Algorithms: The good results offered by CHC-BLX and CHC-EFR indicate that the CHC algorithm is an effective optimizer. In general, these algorithms improve the results of the algorithms based on the other techniques reviewed. They obtain very good A and B results for most functions, in particular, for f_{sph} , f_{Ras} , f_{Gri} , P_{ste} , and P_{fms} . The CHC algorithm has been tested in other GA works against different GA approaches, giving better results, especially on hard problems [19], [35], [68]. Thus, it has arisen as a reference point in the GA literature.

d) GD-RCGA's: At this point, we consider the results for GD-RCGA's in comparison with the ones for the other algorithms. The t -test indicates that GD-RCGA's improve the performance of ECO-GA's and RCGA's based on deterministic crowding and disruptive selection. With regard to the results of the CHC algorithms, we may observe the following

- GD-BLX r improves the A and B results (see t -test results) of CHC-BLX for f_{sph} , ef_{10} , and P_{Chob} . It is outperformed by CHC-BLX only on P_{ste} . Their results for the remaining test problems are similar.
- GD-EFR r do better than CHC-EFR on f_{sph} , f_{Ros} , f_{Sch} , and ef_{10} . It is worse on P_{ste} . Their performance is similar on the remaining test problems.
- We have observed another notable difference between GD-RCGA's and CHC algorithms: the computation time. For most problems, the CHC algorithms were slower than the GD-RCGA's, even though these were executed in a sequential way. A great part of the slowness of the CHC algorithms is due to the prevention of incest process since it requires, during each generation, the encoding of all chromosomes into binary strings and $N/2$ Hamming distance calculations. For problems where there are few evaluations per generation (i.e., crossover operator applications), this slows the process too much. However, the computational times of GD-BLX r and GD-EFR r are similar to the ones of their corresponding homogeneous DGA's, D-BLX, and D-EFR. The increased complexity of the GD-RCGA's (different α , d , and η_{min} values) does not imply an increased computational time.

Finally, we should point out that GD-BLX r and GD-EFR r found the global optimum of P_{fms} in 80% of the runs. None of the remaining algorithms reached this optimum.

All of these results allow us to conclude that GD-RCGA's solve the conflict among accuracy, reliability, and computation time in a suitable way for obtaining a significant performance on test problems with different difficulties, outperforming other mechanisms presented for dealing with the premature convergence problem.

V. CONCLUSION

This paper presented GD-RCGA's, heterogeneous distributed RCGA's based on a hypercubic topology where the subpopulations of the front side use different crossover operators with exploration, and the ones from the rear side use crossover operators with exploitation. The exploration or exploitation degrees of the crossover operators applied to the subpopulations that belong to the same side are gradual, thus obtaining a parallel multiresolution with regard to the crossover operator. The main goal of the gradualism in the GD-RCGA's is to produce a refinement of the best solutions and an expansion of the most promising zones, in a parallel way. An emigration model was selected along with the assignment of different selective pressures for each subpopulation in order to overcome the conquest and noneffect problems, and tune the GD-RCGA behavior properly.

Three instances of GD-RCGA were implemented, using deterministic crossover operators, the FCB-crossover operators, a random one, BLX- α , and a hybrid one (between random and deterministic), namely the extended version of that fuzzy recombination. The results of the experiments carried out with these GD-RCGA's have shown the following.

- GD-RCGA's achieve a suitable balance between the generation of diversity (for inducing reliability) and the local tuning (for introducing accuracy) so that premature convergence is avoided without sacrificing the obtaining of good approximations. This allows GD-RCGA's to improve the performance of other GA approaches appearing in the GA literature for avoiding premature convergence. The good performance of GD-RCGA's is possible due to the gradualism and the effects of the refinement and expansion.
- Reliability of GD-RCGA's may be improved using a restart operator that helps them to overcome stationary states in which refinement and expansion may not produce improvements.
- Since GD-RCGA's are easily implemented on parallel hardware, accuracy and reliability may be reached in an efficient computation time, thus solving the fundamental conflict existing among these three factors when complex problems are tackled.
- BLX- α and extended fuzzy recombination have arisen as very suitable crossover operators for building GD-RCGA's.

Finally, we should point out that GD-RCGA extensions may be followed in three ways: 1) use dynamic crossover operators, such as the *dynamic* FCB crossovers [29] and the *dynamic heuristic* FCB crossovers [31], for producing dynamic levels of refinement and expansion throughout the GA run; 2) use hypercube topologies with a larger subpopulation number in order to include more gradual levels on each side or for combining sides based on different types of crossover operators; and 3) design gradual distributed binary-coded GA's, which may be based on concepts such as disruption, productivity, and exploration

power, which were presented for characterizing the crossover operator for this type of coding [16].

APPENDIX DISTRIBUTED GENETIC ALGORITHMS

This Appendix is devoted to DGA's. In Section A, they will be presented as a class of *parallel* GA's, called *coarse-grained parallel* GA's. In Section B, spatial separation, a basic principle of DGA's, is justified from a biological point of view through the *shifting balance theory of evolution* [70] and the *theory of punctuated equilibria* [18]. In Section C, we describe the basic structure of DGA's. In Section D, we review the types of DGA's presented previously. Finally, in Section E, we tackle heterogeneous DGA's, reporting on the different approaches presented, and explaining the position of the GD-RCGA's relative to these approaches.

A. Parallel Genetic Algorithms

The availability, over the last few years, of fast and inexpensive parallel hardware has favored research into possible ways for implementing parallel versions of GA's. GA's are good candidates for effective parallelization since they are inspired by the principles of evolution, in parallel, for a population of individuals [17]. In general, three methods were followed for implementing the parallelization of GA's [1], [10], [17], [25], [40].

1) *Global Parallelization*: The evaluation of chromosome fitness, and sometimes the genetic operator application are carried out in a parallel form [4], [27], [54].

2) *Coarse-Grained Parallelization*: The population is divided into small subpopulations that are assigned to different processors. Each subpopulation evolves independently and simultaneously according to a GA. Periodically, a *migration mechanism* exchanges individuals between subpopulations, allowing new diversity to be injected into converging subpopulations. The exchange generally takes the form of copying individuals between the populations. Coarse-grained parallel GA's are known as *distributed* GA's since they are usually implemented in distributed memory MIMD computers. Versions of DGA's appeared in [7], [11], [12], [41], [50], [51], [61]–[63], [67].

3) *Fine-Grained Parallelization*: In this model, the population is divided into a great number of small subpopulations. Usually, a unique individual is assigned to each processor. The selection mechanism and the crossover operator are applied by considering neighboring chromosomes. For example, every chromosome selects the best neighbor for recombination, and the resultant individual will replace it. These types of GA's, known as *cellular* GA's, are usually implemented on massively parallel computers. Examples of cellular GA's are to be found in [13], [14], [44], and [49].

B. Spatial Separation

Both distributed GA's and cellular GA's are instances of models based on spatial separation. One of the main advantages of these models is the preservation of diversity. This property caused them to be considered as an important way to research

into mechanisms for dealing with the premature convergence problem [1], [7], [11], [13], [14], [40], [43], [62], [63].

Many authors [13], [14], [49]–[51] have attempted to justify spatial separation models, starting from the *shifting balance theory of evolution*, developed by Wright [70]. This theory explains the process of evolution on the genetic composition of individuals in natural populations. According to this, large populations of organisms rarely act as a single well-mixed (panmictic) population, but rather, they consist of semi-isolated subpopulations, *demes*, each of which is relatively small in size. Furthermore, the demes communicate with each other through migrations of individuals. For Wright, the evolution process has two phases. During the first one, the allele frequencies drift randomly around a local fitness peak in each deme. One of them might, by chance, drift into a set of gene frequencies that corresponds to a higher peak. Then, the second phase starts; this deme produces an excess of offspring, due to its high average fitness, which then emigrate to the other demes, and will tend to displace them until eventually the whole population has the new favorable gene combination. Finally, the process starts again. The relatively small size of the demes allows drift to play an important role in the evolution of the population, without driving the whole population toward convergence. Even if drift were to drive every local deme to fixation, each one of them would be fixed on a different genotype, thereby maintaining diversity in the population as a whole.

Another biological theory adopted by people who do work on spatial separation is the *theory of punctuated equilibria* [18]. This theory states that evolution is characterized by long periods of relative stasis, punctuated by periods of rapid change associated with speciation events. In [11], it is pointed out that GA's also tend toward stasis, or premature convergence, and that isolated species could be formed by separating the global population into subpopulations. By injecting an individual from a different species into a subpopulation after it had converged, new building blocks would become available; furthermore, immigrants would effectively change the fitness landscape within the subpopulations. In this way, premature convergence may be avoided. This idea was highlighted in [50] as well: the creative forces of evolution take place at migration and a few generations afterwards. Wright's argument that better peaks are found just by chance in small subpopulations does not capture the essential facts of the spatial separation.

C. Basic Structure of Distributed GA's

Although there are many different types of DGA's, all of them are variations on the following basic algorithm.

Distributed Genetic Algorithm

- 1) Generate at random a population P of chromosomes.
- 2) Divide P into SP_1, \dots, SP_{N_S} subpopulations.
- 3) Define a neighborhood structure for $SP_i, i = 1, \dots, N_S$.
- 4) For $SP_i, i = 1, \dots, N_S$, execute *in parallel* the next steps.

- 4.1) Apply, during f_m generations, the selection mechanism and the genetic operators.
- 4.2) Send n_m chromosomes to neighboring subpopulations.
- 4.3) Receive chromosomes from neighboring subpopulations.
- 5) If the stop criterion is not fulfilled, return to 4).

Some additional issues should be considered.

- 1) The neighborhood structure, *topology*, to be used. Hypercubic topologies were frequently considered for building DGA's [11], [12], [51], [61]–[63].
- 2) The *migration rate* n_m that controls how many chromosomes migrate.
- 3) The *migration interval* f_m , the number of generations between each migration.
- 4) The *selection strategy* of the genetic material to be copied. Two methods were widely used. The first one is to select randomly the element from the current subpopulation. The advantage of this approach is the greater mix of genes that will result. A second method is to select the highest performing individual from each subpopulation to be copied to another subpopulation. This would result in more directed evolution than the first case, as the migrant individuals would not be tainted by genes from lower performing individuals. This is not to say that the former method is worse, for the less directed a population is, the greater diversity it will contain [56].
- 5) The *replacement strategy* for including the chromosomes to be received. Some approaches are: replace the worst ones, the most similar to the incoming ones, one randomly chosen, etc.
- 6) The choice of *whether or not to replicate migrating individuals*, i.e., should individuals move to their new home or should a copy of them be sent there? If one does not copy individuals, it is possible that a subpopulation could be set back several generations in evolutionary terms by the mass emigration of its best performers. Alternatively, simply copying individuals across could lead to highly fit individuals dominating several populations [56].

D. Types of Distributed GA's

In [40], the following three categorizations of DGA's are reported.

1) Regarding the Migration Method:

- *Isolated DGA's*: There are no migrations between subpopulations. These DGA's are known as well as *partitioned GA's* [62], [63].
- *Synchronous DGA's*: Migrations between subpopulations are synchronized, i.e., they are produced at the same time [12], [51], [62].
- *Asynchronous DGA's*: Migrations are produced when certain events appear, related to the activity of each subpopulation. Asynchronous behavior is typically found in nature since evolution is produced at different states, depending on the environment [40].

2) Regarding the Connection Schema:

- *Static Connection Scheme*: The connections between the subpopulations are established at the beginning of the run, and they are not modified throughout it.
- *Dynamic Connection Scheme*: The connection topology is dynamically changed throughout the run. The reconfigurations in these connections may occur, depending on the evolution state of the subpopulations. For example, in [40], a connection schema called positive-distance topology was proposed in which an individual is passed to another subpopulation only if the Hamming distance between the best individuals in the two subpopulations is less than 24. An analogous connection schema called negative-distance topology was presented as well.

Finally, we point out that some authors [10], [36] assumed another division, based on the connection schema: the *island* model and the *stepping-stone* model. In the first model, individuals can migrate to any other subpopulation; in the second model, migration is restricted to neighboring subpopulations.

3) Regarding the Subpopulation Homogeneity:

- *Homogeneous DGA's*: Every subpopulation uses the same genetic operators, control parameter values, fitness function, coding schema, etc. Most DGA's proposed in the literature are homogeneous. Their principal advantage is that they are easily implemented.
- *Heterogeneous DGA's*: The subpopulations are processed using GA's with either different control parameter values, or genetic operators, or coding schema, etc.

E. Heterogeneous Distributed GA's

Heterogeneous DGA's have been considered as suitable tools for avoiding the premature convergence problem, and for maximizing the exploration and exploitation on the search space. Next, we review some of the most interesting heterogeneous DGA's presented so far:

1) *Adaptation by Competing Subpopulations*: In [57], a heterogeneous DGA model is presented, in which, for each possible operator configuration, a subpopulation or group is formed. The total number of all individuals is fixed, whereas the size of a single subpopulation varies. Each subpopulation competes with other subpopulations in such a way that it gains or loses individuals, depending on its "evolution quality" in relation to the others. A particular instance based on real coding was proposed with four subpopulations (in this paper, called ACS). They were distinguished by applying a mutation operator with different step sizes (proportion or strength in which genes are mutated), which allows a search with multiresolution to be achieved.

A similar model is presented in [58]. Here, the population sizes are fixed, whereas the strategies (mutation rate, crossover rate, the threshold for the truncation selection, etc.) of the subpopulations are flexible. After a fixed interval, all strategies are ranked, and the parameters of each strategy are adapted to the values of the next best strategy.

2) *GA Based on Migration and Artificial Selection*: In [53], a DGA based on binary coding, called GAMAS, was proposed. GAMAS uses four subpopulations, denoted as *species I–IV*.

Initially, species II–IV are created. Species II is a subpopulation used for exploration. For this purpose, it uses a high mutation probability ($p_m = 0.05$). Species IV is a subpopulation used for exploitation. So, its mutation probability is low ($p_m = 0.003$). Species III is an exploration and exploitation subpopulation; the mutation probability falls between the other two ($p_m = 0.005$). GAMAS selects the best individuals from species II–IV, and introduces them into species I whenever those are better than the elements in this subpopulation. The mission of species I is to preserve the best chromosomes appearing in the other species. At predetermined generations, its chromosomes are reintroduced into species IV by replacing all of the current elements in this species.

3) *Heterogeneous DGA's Based on Different Codings*: In [40], a heterogeneous DGA, called the *injection island GA* (iiGA), is built. In iiGA, each subpopulation stores search space solutions coded with different resolutions. Subpopulations inject their best individual into higher resolution subpopulations for fine-grained modification. This allows search to occur in multiple codings, each focusing on different areas of the search space. An important advantage is that the search space in subpopulations with lower resolution is proportionally smaller; in this way, fit solutions are found quickly, and then, they are injected into higher resolution subpopulations for refinement.

4) *Position of the Gradual Distributed RCGA's*: GAMAS assigns exploration and exploitation properties to the subpopulations by applying different mutation probability values to them. In ACS and iiGA, this feature appears generalized to the concept of parallel multiresolution (to assign exploration and exploitation at different degrees). In ACS, this is made by using different step sizes, whereas in iiGA, it is done by means of different codings.

GD–RCGA's include a parallel multiresolution through the crossover operator, which seems reasonable due to the importance of this operator on the GA performance. But, they also attempt to exploit multiresolution in a gradual way, in order to offer the refinement and expansion of promising regions in the search space. In this way, they extend the idea in iiGA of producing fine-grained modification when subpopulations inject individuals into higher resolution subpopulations.

ACKNOWLEDGMENT

The authors wish to thank D. B. Fogel and the anonymous referees for their valuable comments which have improved the presentation of the paper.

REFERENCES

- [1] P. Adamidis, "Review of parallel genetic algorithms bibliography," Aristotle Univ. Thessaloniki, Thessaloniki, Greece, Tech. Rep., 1994.
- [2] P. Adamidis and V. Petridis, "Co-operating populations with different evolution behaviours," in *Proc. 3rd IEEE Conf. Evolutionary Comput.*, New York: IEEE Press, 1996, pp. 188–191.
- [3] T. Bäck, "Self-adaptation in genetic algorithms," in *Proc. 1st Euro. Conf. Artif. Life*, F. J. Varela and P. Bourguine, Eds. Cambridge, MA: M.I.T. Press, 1992, pp. 263–271.
- [4] T. Bäck, T. Beielstein, B. Naujoks, and J. Heistermann, "Evolutionary algorithms for the optimization of simulation models using PVM," in *Proc. EuroPVM'95: 2nd Euro. PVM User's Group Meeting*, J. Dongarra, M. Gengler, B. Tourancheau, and X. Vigouroux, Eds. Paris, France: Hermes, 1995, pp. 277–282.

- [5] J. E. Baker, "Adaptive selection methods for genetic algorithms," in *Proc. 1st Int. Conf. Genetic Algorithms Appl.*, J. J. Grefenstette, Ed. Hillsdale, NJ: Lawrence Erlbaum, 1985, pp. 101–111.
- [6] —, "Reducing bias and inefficiency in the selection algorithm," in *Proc. 2nd Int. Conf. Genetic Algorithms Appl.*, J. J. Grefenstette, Ed. Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 14–21.
- [7] T. C. Belding, "The distributed genetic algorithm revisited," in *Proc. 6th Int. Conf. Genetic Algorithms*, L. Eshelman, Ed. San Francisco, CA: Morgan Kaufmann, 1995, pp. 114–121.
- [8] W. Bossert, *Mathematical Optimization: Are There Abstract Limits on Natural Selection?*, P. S. Moorehead and M. M. Kaplan, Eds. Philadelphia, PA: Wistar Press, 1967, pp. 35–46.
- [9] H. J. Bremermann, M. Rogson, and S. Salaff, "Global properties of evolution processes," in *Natural Automata and Useful Simulations*, H. H. Pattee, E. A. Edsack, L. Frin, and A. B. Callahan, Eds. Washington, DC: Spartan, 1966, pp. 3–41.
- [10] E. Cantú-Paz, "A summary of research on parallel genetic algorithms," Univ. Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, IlliGAL Rep. 95007, 1995.
- [11] J. Cohoon, S. Hedge, W. Martin, and D. Richards, "Punctuated equilibria: A parallel genetic algorithm," in *Proc. 2nd Int. Conf. Genetic Algorithms Appl.*, J. J. Grefenstette, Ed. Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 148–154.
- [12] J. P. Cohoon, W. N. Martin, and D. S. Richards, "Genetic algorithms and punctuated equilibria in VLSI," in *Parallel Problem Solving from Nature 1*, H.-P. Schwefel and R. Männer, Eds. Berlin, Germany: Springer-Verlag, 1990, pp. 134–144.
- [13] R. J. Collins, "Studies in artificial evolution," doctoral dissertation, Univ. California, Los Angeles, CA, 1992.
- [14] Y. Davidor, "A naturally occurring niche species phenomenon: The model and first results," in *Proc. 4th Int. Conf. Genetic Algorithms*, R. Belew and L. B. Booker, Eds. San Mateo, CA: Morgan Kaufmann, 1991, pp. 257–263.
- [15] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, Univ. Michigan, Ann Arbor, 1975.
- [16] K. A. De Jong and W. M. Spears, "A formal analysis of the role of multi-point crossover in genetic algorithms," *Annals Math. Artif. Intell.*, vol. 5, no. 1, pp. 1–26, 1992.
- [17] M. Dorigo and V. Maniezzo, "Parallel genetic algorithms: Introduction and overview of current research," in *Parallel Genetic Algorithms: Theory and Applications: Theory & Applications*, J. Stender, Ed. Amsterdam, The Netherlands: IOS, 1993.
- [18] N. Eldredge and S. J. Gould, "Punctuated equilibria: An alternative to phyletic gradualism," in *Models of Paleobiology*, T. J. M. Schopf, Ed. San Francisco, CA: Freeman, Cooper, 1972, pp. 82–115.
- [19] L. J. Eshelman, "The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination," in *Foundations of Genetic Algorithms 1*, G. J. E. Rawlin, Ed. San Mateo, CA: Morgan Kaufmann, 1991, pp. 265–283.
- [20] L. J. Eshelman and J. D. Schaffer, "Preventing premature convergence in genetic algorithms by preventing incest," in *Proc. 4th Int. Conf. Genetic Algorithms*, R. Belew and L. B. Booker, Eds. San Mateo, CA: Morgan Kaufmann, 1991, pp. 115–122.
- [21] —, "Real-coded genetic algorithms and interval-schemata," in *Foundations of Genetic Algorithms 2*, L. D. Whitley, Ed. San Mateo, CA: Morgan Kaufmann, 1993, pp. 187–202.
- [22] L. J. Eshelman, K. E. Mathias, and J. D. Schaffer, "Convergence controlled variation," in *Foundations of Genetic Algorithms 4*, R. Belew and M. Vose, Eds. San Mateo, CA: Morgan Kaufmann, 1997, pp. 203–224.
- [23] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [24] D. E. Goldberg, H. Kargupta, J. Horn, and E. Cantú-Paz, "Critical deme size for serial and parallel genetic algorithms," Univ. Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, IlliGAL Rep. 95002, 1995.
- [25] V. S. Gordon, D. Whitley, and A. P. W. Böhm, "Dataflow parallelism in genetic algorithms," in *Parallel Problem Solving from Nature 2*, R. Männer and B. Manderick, Eds. Amsterdam, The Netherlands: Elsevier, 1992, pp. 533–542.
- [26] V. S. Gordon and D. Whitley, "Serial and parallel genetic algorithms as function optimizers," in *Proc. 5th Int. Conf. Genetic Algorithms*, S. Forrest, Ed. San Mateo, CA: Morgan Kaufmann, 1993, pp. 177–183.
- [27] J. J. Grefenstette, "Parallel adaptive algorithms for function optimization," Vanderbilt University, Nashville, TN, Tech. Rep. CS-81-19, 1981.
- [28] A. O. Griewangk, "Generalized descent of global optimization," *J. Optim. Theory Appl.*, vol. 34, pp. 11–39, 1981.

- [29] F. Herrera, M. Lozano, and J. L. Verdegay, "Dynamic and heuristic fuzzy connectives based crossover operators for controlling the diversity and convergence of real-coded genetic algorithms," *Int. J. Intell. Syst.*, vol. 11, pp. 1013–1041, 1996.
- [30] F. Herrera and M. Lozano, "Adaptation of genetic algorithm parameters based on fuzzy logic controllers," in *Genetic Algorithms and Soft Computing*, F. Herrera and J. L. Verdegay, Eds. Berlin, Germany: Physica-Verlag, 1996, vol. 8, Studies in Fuzziness and Soft Computing, pp. 95–125.
- [31] —, "Heuristic crossovers for real-coded genetic algorithms based on fuzzy connectives," in *4th Int. Conf. Parallel Problem Solving Nature*, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Berlin, Germany: Springer, 1996, vol. 1141, Lecture Notes on Computer Science, pp. 336–345.
- [32] F. Herrera, M. Lozano, and J. L. Verdegay, "Fuzzy connectives based crossover operators to model genetic algorithms population diversity," *Fuzzy Sets Syst.*, vol. 92, no. 1, pp. 21–30, 1997.
- [33] F. Herrera and M. Lozano, "Heterogeneous distributed genetic algorithms based on the crossover operator," in *2nd IEE/IEEE Int. Conf. Genetic Algorithms Eng. Syst.: Innovations Appl.*, 1997, pp. 203–208.
- [34] F. Herrera, M. Lozano, and J. L. Verdegay, "Tackling real-coded genetic algorithms: Operators and tools for behavioral analysis," *Artificial Intell. Rev.*, vol. 12, no. 4, pp. 265–319, 1998.
- [35] J. L. Keith, E. Mathias, and L. D. Whitley, "Changing representations during search: A comparative study of delta coding," *Evolutionary Comput.*, vol. 2, no. 3, pp. 249–278, 1994.
- [36] U. Kohlmoorgen, "Overview of parallel genetic algorithms," in *Proc. 1st Nordic Workshop Genetic Algorithms*, Vaasa, Finland, 1995, pp. 135–145.
- [37] B. Kröger, P. Schwenderling, and O. Vornberger, "Parallel genetic packing on transputers," in *Parallel Genetic Algorithms: Theory and Applications: Theory Applications*, J. Stender, Ed. Amsterdam, The Netherlands: IOS, 1993.
- [38] T. Kuo and S. Y. Hwang, "A genetic algorithm with disruptive selection," *IEEE Trans. Syst., Man, Cybern.*, vol. 26, no. 2, pp. 299–307, 1996.
- [39] T.-H. Li, C. B. Lucasius, and G. Kateman, "Optimization of calibration data with the dynamic genetic algorithm," *Analytica Chimica Acta*, vol. 2768, pp. 123–134, 1992.
- [40] S.-C. Lin, W. F. Punch III, and E. D. Goodman, "Coarse-grain genetic algorithms: Categorization and new approach," in *Proc. 6th IEEE Parallel Distributed Processing*, New York: IEEE Press, 1994, pp. 28–37.
- [41] A. Loraschi, A. Tettamanzi, M. Tomassini, and P. Verda, "Distributed genetic algorithms with an application to portfolio selection," in *Artificial Neural Networks and Genetic Algorithms*, D. W. Pearson, N. C. Steele, and R. F. Albrecht, Eds. Berlin, Germany: Springer-Verlag, 1995.
- [42] S. W. Mahfoud, "Crowding and preselection revised," in *Parallel Problem Solving from Nature 2*, R. Männer and B. Manderick, Eds. Amsterdam, The Netherlands: Elsevier, 1992, pp. 27–36.
- [43] —, "Nicheing methods for genetic algorithms," Univ. Illinois at Urbana-Champaign, Illinois Genetic Algorithms Lab., IlliGAL Rep. 95001, 1995.
- [44] B. Manderick and P. Spiessens, "Fine-grained parallel genetic algorithms," in *Proc. 3rd Int. Conf. Genetic Algorithms*, J. D. Schaffer, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 428–433.
- [45] J. Maresky, "On efficient communication in distributed genetic algorithms," M.S. thesis, Hebrew Univ. Jerusalem, Inst. Comput. Sci., 1994.
- [46] J. Maresky, Y. Davidor, D. Gitler, G. Aharoni, and A. Barak, "Profiling communication in distributed genetic algorithms," in *Proc. 14th Int. Joint Conf. Artif. Intell.*, San Mateo, CA: Morgan Kaufmann, 1995, pp. 961–966.
- [47] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin, Germany: Springer, 1992.
- [48] M. Mizumoto, "Pictorial representations of fuzzy connectives—Part I: Cases of t -norms, t -conorms and averaging operators," *Fuzzy Sets Syst.*, vol. 31, pp. 217–242, 1989.
- [49] H. Mühlenbein, "Parallel genetic algorithms, population genetics and combinatorial optimization," in *Proc. 3rd Int. Conf. Genetic Algorithms*, J. D. Schaffer, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 416–421.
- [50] —, "Evolution in time and space—The parallel genetic algorithm," in *Foundations of Genetic Algorithms I*, G. J. E. Rawlin, Ed. San Mateo, CA: Morgan Kaufmann, 1991, pp. 316–337.
- [51] H. Mühlenbein, M. Schomisch, and J. Born, "The parallel genetic algorithm as function optimizer," in *4th Int. Conf. Genetic Algorithms*, R. Belew and L. B. Booker, Eds. San Mateo, CA: Morgan Kaufmann, 1991, pp. 271–278.
- [52] J. Périaux, M. Sefrioui, B. Stoufflet, B. Mantel, and E. Laporte, "Robust genetic algorithms for optimization problems in aerodynamic design," in *Genetic Algorithms in Engineering and Computer Science*, G. Winter, J. Périaux, M. Galán, and P. Cuesta, Eds. Chichester, England: Wiley, 1995, pp. 371–396.
- [53] J. C. Potts, T. D. Giddens, and S. B. Yadav, "The development and evaluation of an improved genetic algorithm based on migration and artificial selection," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, pp. 73–86, 1994.
- [54] W. Punch, E. Goodman, M. Pei, L. Chai-Shun, P. Hovland, and R. Enderby, "Further research on feature selection and classification using genetic algorithms," in *Proc. 5th Int. Conf. Genetic Algorithms*, S. Forrest, Ed. San Mateo, CA: Morgan Kaufmann, 1993, pp. 557–564.
- [55] J.-M. Renders and S. P. Flasse, "Hybrid methods using genetic algorithms for global optimization," *IEEE Trans. Syst., Man, Cybern.*, vol. 26, no. 2, pp. 243–258, 1996.
- [56] C. Ryan, "Niche and species formation in genetic algorithms," in *Practical Handbook of Genetic Algorithms: Applications*, L. Chambers, Ed. Boca Raton, FL: CRC Press, 1995, pp. 57–74.
- [57] D. Schlierkamp-Voosen and H. Mühlenbein, "Strategy adaptation by competing subpopulations," in *Parallel Problem Solving from Nature 3*, Y. Davidor, H.-P. Schwefel, and R. Männer, Eds. Berlin, Germany: Springer-Verlag, 1994, pp. 199–208.
- [58] V. Schneck and O. Vornberger, "An adaptive parallel genetic algorithm for VLSI-layout optimization," in *Proc. 4th Int. Conf. Parallel Problem Solving from Nature*, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Berlin, Germany: Springer, 1996, vol. 1141, Lecture Notes on Comput. Sci., pp. 859–868.
- [59] H.-P. Schwefel, *Numerical Optimization of Computer Models*. Chichester, England: Wiley, 1981.
- [60] R. Storn and K. Price, "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces," *Int. Comput. Sci. Inst., Berkeley, CA, Tech. Rep. TR-95-012*, 1995.
- [61] R. Tanese, "Parallel genetic algorithms for a hypercube," in *Proc. 2nd Int. Conf. Genetic Algorithms Appl.*, J. J. Grefenstette, Ed. Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 177–183.
- [62] —, "Distributed genetic algorithms," in *Proc. 3rd Int. Conf. Genetic Algorithms*, J. D. Schaffer, Ed. San Mateo, CA, 1989, pp. 434–439.
- [63] —, "Distributed genetic algorithms for function optimization," doctoral dissertation, Univ. Michigan, Ann Arbor, 1989.
- [64] A. Törn and Ž. Antanas, *Global Optimization*. Berlin, Germany: Springer, 1989, vol. 350, Lecture Notes in Comput. Sci..
- [65] S. Tsutsui and Y. Fujimoto, "Forking genetic algorithm with blocking and shrinking modes," in *Proc. 5th Int. Conf. Genetic Algorithms*, S. Forrest, Ed. San Mateo, CA: Morgan Kaufmann, 1993, pp. 206–213.
- [66] H.-M. Voigt, H. Mühlenbein, and D. Cvetković, "Fuzzy recombination for the breeder genetic algorithm," in *Proc. 6th Int. Conf. Genetic Algorithms*, L. Eshelman, Ed. San Mateo, CA: Morgan Kaufmann, 1995, pp. 104–111.
- [67] D. Whitley and D. Starkweather, "Genitor-II: A distributed genetic algorithm," *J. Exp. Theor. Artif. Intell.*, vol. 2, pp. 189–214, 1990.
- [68] D. Whitley, S. Rana, J. Dzubera, and E. Mathias, "Evaluating evolutionary algorithms," *Artif. Intell.*, vol. 85, pp. 245–276, 1996.
- [69] D. Whitley, R. Beveridge, C. Graves, and K. Mathias, "Test driving three 1995 genetic algorithms: New test functions and geometric matching," *J. Heuristics*, vol. 1, pp. 77–104, 1995.
- [70] S. Wright, "The roles of mutation, inbreeding, crossbreeding and selection in evolution," in *Proc. 6th Int. Congr. Genetics*, vol. 1, 1934, pp. 356–366.



Francisco Herrera received the M.S. degree in mathematics in 1988, and the Ph.D. degree in mathematics, in 1991, both from the University of Granada, Spain.

He is an Associate Professor in the Department of Computer Science and Artificial Intelligence, University of Granada.

Dr. Herrera has coedited a book, *Genetic Algorithms and Soft Computing*, (Physica Verlag, with J. L. Verdegay) and two journal special issues, *Genetic Fuzzy Systems for Control and Robotics*, (*International Journal of Approximate Reasoning*) and *Genetic Fuzzy Systems*, (*International Journal of Intelligent Systems*, with L. Magdalena). His research interests include decision-making problems in fuzzy environment, fuzzy rule-based systems, machine learning, genetic algorithms, genetic fuzzy systems, and the combination of fuzzy logic and genetic algorithms.



Manuel Lozano received the M.Sc. degree in computer science in 1992, and the Ph.D degree, also in computer science in 1996, both from the University of Granada, Spain.

He is an Assistant Professor in the Department of Computer Science and Artificial Intelligence, University of Granada. His research interests include fuzzy rule-based systems, machine learning, genetic algorithms, genetic fuzzy systems, and the combination of fuzzy logic and genetic algorithms.