Information Sciences 178 (2008) 4421–4433

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Replacement strategies to preserve useful diversity in steady-state genetic algorithms

Manuel Lozano [a,*], Francisco Herrera [a], José Ramón Cano [b]

[a] *Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain*
[b] *Department of Electronic Engineering, Computer Systems and Automatics, Escuela Superior de La Rábida, University of Huelva, 21819 Huelva, Spain*

ARTICLE INFO

Keywords:
Steady-state genetic algorithms
Useful diversity
Replacement strategy

ABSTRACT

In this paper, we propose a replacement strategy for steady-state genetic algorithms that considers two features of the candidate chromosome to be included into the population: a measure of the contribution of diversity to the population and the fitness function. In particular, the proposal tries to replace an individual in the population with worse values for these two features. In this way, the diversity of the population becomes increased and the quality of the solutions gets better, thus preserving high levels of useful diversity. Experimental results show the proposed replacement strategy achieved significant performance for problems with different difficulties, with regards to other replacement strategies presented in the literature.

1. Introduction

There are two primary factors in the search carried out by a genetic algorithm (GA): population diversity and selective pressure [49]. In order to have an effective search there must be a search criteria (the fitness function) and a selection pressure that gives individuals with higher fitness a higher chance of being selected for reproduction, mutation, and survival. Without selection pressure, the search process becomes random and promising regions of the search space would not be favoured over regions offering no promise. On the other hand, population diversity is crucial to a GA's ability to continue the fruitful exploration of the search space [24]. If the lack of population diversity takes place too early, a premature stagnation of the search is caused. Under these circumstances, the search is likely to be trapped in a region not containing the global optimum. This problem, called *premature convergence*, has long been recognized as a serious failure mode for GAs [13,17].

Selective pressure and population diversity are inversely related [49]. Increasing selective pressure results in a faster loss of population diversity, while maintaining population diversity offsets the effect of increasing selective pressure. These two factors should be controlled in order to obtain their beneficial advantages simultaneously, allowing the most promising search space regions to be reached and refined. Under these circumstances, we will say that the population has achieved *useful diversity*, i.e., population diversity that in some way helps produce good solutions [26].

Steady-state GAs (SSGAs) [42,49] usually only produce one or two offspring in each generation. Parents are selected to produce offspring and then a decision is made as to which individuals in the population to select for deletion to make room for the new offspring. SSGAs are *overlapping* systems, because parents and offspring compete for survival. Different studies have shown that improved performance in SSGAs on generational GAs (which replace the entire population by the offspring population every generation) is because of their higher selection pressure and changes in the exploration/exploitation

---

[*] Corresponding author.
E-mail address: lozano@decsai.ugr.es (M. Lozano).

0020-0255/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2008.07.031

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

Fig. 1. Pseudocode algorithm for the SSGA model.

balance caused by using different parent selection and replacement strategies and is not because of the use of an overlapping model [8]. This justifies the study on SSGAs, and in particular, the design of effective replacement strategies having the aim of propitiating useful diversity [2,16].

In this paper, we propose a replacement strategy that considers two features of the individual to be included into the population: a measure of the contribution of diversity to the population and the fitness function. It tries to replace a chromosome in the population with poorer values for these two features. In this way, the diversity of the population increases and the quality of the solutions improves, simultaneously. The goal of this strategy is to protect those individuals that preserve the highest levels of useful diversity.

We set up the paper as follows. In Section 2, we describe the SSGA model. In Section 3, we detail different replacement strategies presented in the literature to promote SSGA population diversity. In Section 4, we propose the replacement strategy based on the contribution of diversity. In Section 5, we analyse the empirical experiments carried out to study the performance of this strategy. Finally, we draw some conclusions in Section 6.

2. Steady-state genetic algorithms

The generational GA creates new offspring from the members of an old population using the genetic operators and places these individuals in a new population which becomes the old population when the whole new population is created.

The SSGA [42,49] is different to the generational model in that there is typically one single new member inserted into the new population at any one time. A replacement/deletion strategy defines which member in the current population is forced to perish (or vacate a slot) in order to make room for the new offspring to compete (or, occupy a slot) in the next iteration. SSGAs are overlapping systems, because parents and offspring compete for survival. Fig. 1 shows the basic algorithm step of SSGA[1].

These steps are repeated until a termination condition is achieved. In step 4, one can chose the *replacement strategy* (e.g., replacement of the worst, the oldest, or a randomly chosen individual). In step 5, one can choose the *replacement condition* (e.g., replacement if the new individual is better, or unconditional replacement). A widely used combination is to replace the worst individual only if the new individual is better [48]. In the paper, this strategy will be called *replace worst strategy* (RW). Goldberg and Deb [18] suggest that the deletion of the worst individuals induces a high selective pressure, even when the parents are selected randomly.

3. Replacement strategies to preserve population diversity

There are different replacement strategies that try to enforce population diversity. Most of them are instances of the *crowding* methods [10,26]. They work as follows: new individuals are more likely to replace existing individuals in the parent population that are similar to themselves based on genotypic similarity. In this manner, the population does not build up an excess of similar solutions. Crowding methods promote the formation of stable subpopulations in the neighbourhood of optimal solutions (*niches*), favouring the preservation of multiple local optima in multimodal problems [33]. Next, we describe different examples of crowding methods.

3.1. Restricted tournament selection

An effective crowding method is the *restricted tournament selection* (RTS) [20]. RTS initially selects two element at random, A and B, from the population and perform crossover and mutation on these two elements resulting in a new element A'. Then, RTS scans ω (*window size*) more members of the population and picks the individual that most closely resembles A' from those ω elements. A' then competes with this element, and if A' wins, it is allowed to enter the population.

---

[1] Different theoretical studies on SSGAs appear in [4,8,32,39,47].

3.2. Worst among most similar replacement policy

Implementation of the worst among most similar replacement policy (WAMS) follows these steps [5,6]. First, C_f *crowding groups* are created by randomly picking C_s *crowding group size* individuals (with replacement) per group from the population. Second, one individual from each group that is most similar to the offspring is identified. This gives C_f individuals that are candidates for replacement by virtue of their similarity to the offspring. The offspring will replace one of them. From this group of most similar candidates, we pick the one with the lowest fitness to die and be replaced by the offspring.

The effect of WAMS replacement can also be explained by the values of the crowding group size and crowding factor parameters [6]. The value of the crowding factor controls the selective pressure in the SSGA. Increasing the value of this parameter increases the fitness pressure for the individuals in the population. The higher the value of C_f, the higher the chances of eliminating low fit individuals using WAMS replacement. The value of C_s controls the amount of inter-niche and intra-niche competition. The lower the value of C_s, the higher the competition between members of different niches. Both values can be combined to achieve the desire selective pressure and niche competition in a run. The values of C_s and C_f control the classical tradeoff between exploration and exploitation in SSGAs.

Goldberg [17] uses a similar technique in classifier systems (called *enhanced crowding*) but he replaced the most similar individual out of a group of lowest fitness candidates. Stadnyk [40] and Sedbrook et al. [34] applied other *closest of the worst* replacement strategies as well.

3.3. Family competition replacement schemes

Another type of crowding methods assumes that the parents would be one of the members of the population closest to the new elements. In this way, children compete with their parents to be included in the population, i.e., a *family competition* is held. These methods include *deterministic crowding* [25], *elitist recombination* [43], *keep-best reproduction* [51], and *correlative family-based selection* [27].

- In deterministic crowding (DC), each pair of parents, (P_1,P_2), undergoes crossover in combination with mutation to yield two offspring, (O_1,O_2). Then, each child replaces the nearest parent if it has a higher fitness. In particular, DC results in two sets of tournaments: P_1 against O_1 and P_2 against O_2 or P_1 against O_2 and P_2 against O_1. The set of tournaments that yields the closest competitions is held. Fig. 2 shows this process, where $f(\cdot)$ is the fitness function and $d(\cdot,\cdot)$ is a distance measure between two chromosomes. Mengshoel and Goldberg [28] proposed *probabilistic crowding* as a probabilistic extension of the original DC method. In probabilistic crowding, the winner of the parent–offspring tournament is chosen by using the probability proportional to the fitness.
- In the elitist recombination, the best two of these four individuals (parents and offspring) go to the next generation.
- Keep-best maintains the best parent and the best offspring in order to introduce good new genetic material into the population.
- Correlative family-based selection chooses the best of the four individuals as first survivor. From the remaining three elements, the one with the highest distance from the best becomes the second survivor.

4. Replacement strategy based on contribution of diversity

We may see the population diversity versus selective pressure problem as a biobjective problem. The GA may search simultaneously the best chromosomes and the ones that provide more diversity to the GA population, that is, promoters

```
If $(d(P_i,O_i) + d(P_j,O_j)) \le (d(P_i,O_j) + d(P_j,O_i))$ 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 $O_j$.
```

Fig. 2. Pseudocode algorithm for deterministic crowding.

of useful diversity. In fact, several generational GA approaches were proposed which use this idea to avoid premature convergence and improve GA behaviour [9,22,23,29,37,44]. During the selection phase, they use the fitness function and some measures to evaluate the diversity introduced by the competing chromosomes, with the aim of favouring those individuals with both high fitness function values and high diversity contributions.

We propose to solve the population diversity versus selective pressure problem in the replacement phase of an SSGA. In particular, we present a replacement strategy based on the contribution of diversity of the new offspring to the population where it will be included. The idea is to replace an individual of the population, with a poorer fitness function value and with a lower contribution to diversity, by the offspring.

In this way, we deal with two underlying objectives simultaneously: to optimize the fitness function and to enhance population diversity, that is, to promote useful diversity.

4.1. Contribution of diversity

In this section, we present a measure that quantifies the importance of a chromosome as promoter of diversity in the population. In particular, we define the contribution of diversity of the chromosome c to the population P, $CD(c,P)$, as the similarity between itself and its nearest neighbour in the population P, that is:

$$CD(c,P) = \min_{x \in P, x \neq c} d(c,x),$$

where $d(\cdot,\cdot)$ is a distance function (e.g., Hamming distance, Euclidean distance, etc.). Under this definition, a chromosome will provide a great contribution of diversity when it is too different from its most similar individual in the population. This measure was used already in [3] as solution density metric.

4.2. Proposed replacement strategy

Let us assume that an offspring, c_{new}, is returned from the recombination phase of an SSGA, then we propose to include c_{new} in a population P with the strategy in Fig. 3. The proposed replacement will be called CD/RW because it is composed of two replacement strategies, the one based on the contribution of diversity (Step 2) and the RW strategy (Step 3) described in Section 2.

Firstly, CD/RW considers the individuals in the population with poorer fitness values than the offspring and finds the one with the lowest contribution of diversity, $c_{\min}$. Then, it compares the contribution of diversity of this element with the contribution of diversity of the offspring to the population (removing $c_{\min}$ from it). The offspring replaces $c_{\min}$ whenever it wins (Step 2). In this way, we introduce into the population an individual with better fitness value and higher contribution of diversity, increasing the population diversity and improving the quality of its solutions. If the offspring does not provide more diversity than $c_{\min}$, then the RW strategy is applied (Step 3). Next, we list several reasons for the inclusion of this replacement policy in our proposal:

- For most practical problems, the refinement of the best regions of the search space (i.e., the convergence) requires the loss of population diversity. If we consider the CD replacement (Step 2) as unique replacement procedure, the SSGA may reach a stationary point without any improvement, because it will not accept promising chromosomes (which enforce convergence) because they do not contribute diversity (we have observed this effect in our experimental study in Section 5.3). The incorporation of the RW replacement ensures the best elements found during the run to enter the population (*elitist* scheme), allowing the SSGA convergence to be possible.

```
(1) Find $c_{min} = \min_{c \in D} CD(c,P)$, where
        $D = \{x \in P | f(c_{new}) \text{ is better than } f(x)\}$.
(2) CD Replacement: If $CD(c_{new}, P - \{c_{min}\}) > CD(c_{min}, P)$ Then
        replace $c_{min}$ by $c_{new}$.
(3) RW Replacement: Else apply the replace worst strategy to
        insert $c_{new}$ in $P$.
```

Fig. 3. Pseudocode algorithm for the proposed replacement strategy.

- Different replacement strategies perform differently with respect to different problems, even at the different stages of the genetic process in the same problem. No free lunch theorems confirm this fact [52]. An interesting idea to devise promising replacement techniques, which may be suited to most practical problems, would consist of the simultaneous employment of diverse replacement operations during the SSGA run [11], i.e., *hybrid* replacement schemes. The idea of improving GA performance by combining different genetic operators appear in [7,53] as well.

  The proposed strategy is a hybrid replacement scheme that combines a replacement that favours high diversity levels with another that introduces high selective pressure. Other authors, with the aim of providing an effective search, have suggested GA schemes that connect techniques with high exploration and techniques with high exploitation. For example, Shimodaira [38] presents an evolutionary algorithm that employs large mutation rates and population-elitist selection. Another example is the CHC algorithm [12], which combines a disruptive crossover operator with a conservative selection strategy (which keeps the N best individuals appearing so far).

5. Experiments

We have carried out minimization experiments on the test suite described in Appendix A in order to study the behaviour of the CD/RW replacement strategy presented in the previous section. In Section 5.1, we compare the results for the proposal with the ones for other replacement strategies that preserve population diversity (Section 3). In Section 5.2, we undertake the analysis of its behaviour by means of the study of two important aspects associated with the SSGA execution, the population diversity and the variability (which quantifies the renewal activity in the population). In Section 5.3, we detect whether there exists synergy between the two components of the proposal, the CD replacement and the RW replacement. In Section 5.4, we consider other possible measures of contribution of diversity for the design of alternative replacement strategies. Finally, in Section 5.5, we analyse the computational time of the proposal.

5.1. Comparison with other replacement strategies

In this section, we compare the proposed replacement strategy, CD/RW, with the replacement strategies described in Section 3, which favour the maintenance of population diversity. In order to do this, we have implemented several SSGAs that are distinguished uniquely by the replacement policy. They use real coding [21] and apply the BGA mutation operator [31] and the BLX-α operator ($\alpha = 0.5$) [14] (the description of these operators is found in Appendix B). The mutation probability is $\frac{1}{n}$ and the population size is 60 chromosomes. Parents are selected at random. All the algorithms were executed 50 times, each one with a maximum of 100,000 evaluations. Table 1 shows these algorithms along with their corresponding replacement strategy. We have considered different combination of values for the C_f and C_s parameters associated with the WAMS replacement, following the recommendations of its authors [6]: values between 2% and 10% of the population size are recommended values for C_f whereas values between 5% and 15% of the population size are also acceptable for C_s. We have tried the values 2, 5, 10, 30, and 60 for the ω parameter of the RTS technique. In addition, we have executed an SSGA with the RW strategy as standard SSGA.

Table 2 shows the results obtained. The performance measure used is the average of the best fitness function found at the end of each run. In addition, a two-sided t-test (H_o: means of the two groups are equal, H_a: means of the two group are not equal) at 0.05 level of significance was applied in order to ascertain if differences in the performance for CD/RW are significant when compared against the ones for the other algorithms. We denote the direction of any significant differences as follows:

- A plus sign (+): the performance of CD/RW is better than the one of the corresponding algorithms.
- A minus sign ($-$): the algorithm improves the performance of CD/RW.
- An approximate sign ($\sim$): non-significant differences.

We have included the last three columns to facilitate the analysis of the results. They have the number of improvements, reductions, and non-differences (according to the t-test) in the performance for CD/RW with regards to the one for all the other algorithms.

Table 1
Algorithms based on other replacement strategies

Algorithm	Replacement strategy	Section
RTS_ω	Restricted tournament selection	3.1
$\mathrm{WAMS}_{C_s}^{C_f}$	Worst among most similar replacement	3.2
DC	Deterministic crowding	3.3
ER	Elitist recombination	3.3
KB	Keep-best reproduction	3.3
CS	Correlative family-based selection	3.3

Table 2
Comparison with other replacement strategies

Alg.	f_{sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}	P_{Cheb}	P_{fms}	+	∼	−
RW	2e−4 ∼	2.29e1 ∼	2.9 +	1.2 +	1.4e−2 +	1.2e2 +	1.4e1 +	7.8e2 +	6	2	0
RTS$_2$	7e−32 +	2.11e1 +	1.1e1 +	1.9e1 +	1.5e−3 −	2.8e1 ∼	3.0 ∼	3.1e2 ∼	4	3	1
RTS$_5$	6e−46 +	2.17e1 ∼	3.1e−1 +	2.9 +	2.6e−3 −	5.6e1 +	3.6 ∼	3.4e2 +	5	2	1
RTS$_{10}$	5e−12 +	2.21e1 ∼	1.6e−1 +	2.6e−1 ∼	1.2e−2 +	6.7e1 +	9.2 +	3.6e2 +	6	2	0
RTS$_{30}$	3e−2 +	2.89e1 +	1.6e3 +	1.2e2 +	7.3e−1 +	1.7e2 +	1.3e1 +	4.9e3 +	8	0	0
RTS$_{60}$	9e−35 ∼	2.28e1 ∼	1.7 +	6.6e−1 +	1.7e−2 +	7.9e1 +	7.9 +	5.3e2 +	6	2	0
WAMS$_3^2$	2e−22 +	2.00e1 ∼	9.2 +	8.9e−1 +	1.7e−2 +	5.9e1 +	7.1 +	4.7e2 +	7	1	0
WAMS$_9^2$	5e−34 ∼	1.99e1 −	9.6e−1 +	9.2e−1 +	1.7e−2 +	6.2e1 +	7.7 +	4.8e2 +	6	1	1
WAMS$_6^4$	1e−16 ∼	2.01e1 ∼	5.6e−1 +	6.0e−1 +	1.3e−2 +	1.1e2 +	1.0e1 +	6.2e2 +	6	2	0
WAMS$_3^6$	2e−45 +	1.99e1 −	5.7e−1 +	6.7e−1 +	1.2e−2 +	6.0e1 +	8.8 +	4.5e2 +	7	0	1
WAMS$_9^6$	1e−31 ∼	2.14e1 ∼	1.8 +	8.4e−1 +	2.1e−2 +	1.1e2 +	8.4 +	5.3e2 +	6	2	0
DC	1e−17 +	1.95e1 −	1.6e1 +	9.9e−1 +	2.0e−2 +	7.3e1 +	2.7 ∼	5.5e2 +	6	1	1
ER	7e−14 +	2.05e1 ∼	6.8e1 +	5.7 +	2.0e−2 +	5.3e1 +	3.6 ∼	6.0e2 +	6	2	0
KB	9e−18 +	2.08e1 ∼	1.3e2 +	1.6e1 +	2.3e−2 +	6.6e1 +	1.7 −	8.3e2 +	6	1	1
CS	3e−20 +	2.13e1 +	1.4e2 +	2.8e1 +	6.3e−2 +	6.6e1 +	4.4 ∼	1.1e3 +	7	1	0
CD/RW	5e−51	2.05e1	1.0e−1	3.2e−1	7.6e−3	3.7e1	3.9	2.3e2			

We may observe that CD/RW consistently outperforms all the other algorithms, because it achieves remarkable amount of improvements and few reductions. In addition, the number of non-differences is too small. The source of diversity for these algorithms is the same as CD/RW, because they use the same mating and recombination strategy. The difference between them concerns the way in which diversity is retained by the different replacement strategies. The promising results of the proposal reveal that it conserves those individuals (promoters of useful diversity) that allow the search process to be driven towards better regions than by means of the other strategies. Thus, we may conclude that it is an adequate replacement strategy for SSGAs.

5.2. *Study of diversity and variability*

Next, we attempt to discover those behavioural characteristics that allow the proposed replacement strategy to decisively affect SSGA performance. In order to do this, we analyse two prominent aspects of the SSGAs executed:

- The population diversity kept during the run. In particular, we have studied the following diversity measure:

$$\mathrm{Div}(P) = \sum_{x \in P} CD(x, P),$$

where $CD(\cdot,\cdot)$ is the measure of contribution of diversity presented in Section 4.1. We recorded the value for this diversity measure every 1000 iterations, and then, we calculated the average value.
- The percentage of chromosomes accepted to enter the population. We will call *variability* of the SSGA to this value.

For each test problem, we have ranked the algorithms attending on their supplied diversity (Table 4) and variability (Table 3). We have only considered representative algorithms providing the best results (RW, RTS$_2$, WAMS$_9^6$, DC, ER, and CD/RW). The values for the diversity measure and the variability appear in brackets.

Taking into consideration Tables 3 and 4, we would make the following comments:

- The RW strategy induces high variability values. This occurs because its formulation favours the inclusion of individuals in the population; to be better than the worst chromosome is an unstrained requisite to visit the population. However, this feature introduces strong selective pressure in the search process [18], limiting rapidly the diversity in the population. In fact, we may observe in Table 4 that RW shows the lowest diversity values. This circumstance forces the apparition of premature convergence, which explains the poor results obtained by this strategy.

Table 3
Variability

f_{sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}	P_{Cheb}	P_{fms}
RW (40)	**CD/RW** (50)	RW (31)	RW (22)	WAMS$_9^6$ (10)	RW (25)	RW (31)	**CD/RW** (37)
CD/RW (39)	RW (44)	**CD/RW** (14)	**CD/RW** (18)	**CD/RW** (9)	**CD/RW** (16)	**CD/RW** (12)	RW (36)
ER (26)	ER (34)	RTS$_2$ (12)	WAMS$_9^6$ (13)	RW (8)	RTS$_2$10)	RTS$_2$ (12)	WAMS$_9^6$ (22)
WAMS$_9^6$ (25)	WAMS$_9^6$ (33)	WAMS$_9^6$ (10)	RTS$_2$ (12)	ER (6)	ER (9)	WAMS$_9^6$ (10)	RTS$_2$ (16)
DC (20)	DC (26)	ER (9)	ER (6)	RTS$_2$ (5)	WAMS$_9^6$ (8)	ER (8)	ER (14)
RTS$_2$ (18)	RTS$_2$ (25)	DC (3)	DC (2)	DC (4)	DC (3)	DC (2)	DC (3)

Table 4
Diversity

f_{sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}	P_{Cheb}	P_{fms}
DC (0.12)	DC (0.2)	DC (24.4)	DC (3.19)	DC (7.7)	DC (7.7)	DC (55.9)	DC (1.42)
RTS$_2$ (0.11)	RTS$_2$ (0.15)	WAMS$_9^6$ (10.2)	ER (1.1)	RTS$_2$ (4)	WAMS$_9^6$ (3.8)	WAMS$_9^6$ (15)	ER (0.37)
ER (0.07)	ER (0.11)	ER (9.8)	**CD/RW (0.62)**	WAMS$_9^6$ (3.2)	ER (2.5)	ER (13)	WAMS$_9^6$ (0.15)
CD/RW (0.06)	**CD/RW** (0.1)	**CD/RW** (8.8)	WAMS$_9^6$ (0.56)	ER (2.3)	RTS$_2$ (2.3)	**CD/RW** (8.2)	**CD/RW** (0.12)
WAMS$_9^6$ (0.06)	WAMS$_9^6$ (0.1)	RTS$_2$ (6.8)	RTS$_2$ (0.35)	**CD/RW** (2)	**CD/RW** (2)	RTS$_2$ (7.7)	RTS$_2$ (0.11)
RW (0.02)	RW (0.04)	RW (2.7)	RW (0.14)	RW (0.57)	RW (0.6)	RW (1.8)	RW (0.033)

- DC, ER, WAMS$_9^6$, and RTS$_2$ provide the highest levels of diversity, however, they introduce very low variability. To a certain extent, the last may be responsible of the maintenance of high diversity during the run. Low variability indicates that the population underwent few changes (few individuals enter the population), and thus it remains inalterable during long time periods, conserving its diversity.
- The proposed replacement strategy achieves high variability values and keeps intermediate diversity levels. This means that the population suffers frequent modifications, accepting many dissimilar individuals (which should have promising fitness values associated). The entrance of this type of chromosomes enhances the operation of the crossover operator because enables the combination of diverse genetic material. This fact allows a wide sampling of different promising region of the search space to be possible.

5.3. Synergy among CD replacement and RW replacement

In this section, we examine the synergetic effects among the constituent replacement strategies of the proposal, the CD replacement and the RW replacement (Fig. 3). The objective is to investigate whether or not CD/RW performs better than the sole usage of these two strategies. Table 5 includes the results for these three algorithms. In addition, we have also introduced Table 6 in order to provide a detailed investigation on their behaviour. It outlines the variability values for CD, RW, and CD/RW, and contains in brackets the percentage of times the CD/RW accepted an individual through the CD strategy.

Results in Table 5 offer a clear conclusion; the proposal improves the results of the CD and RW replacement strategies. Then, we may conclude that there exists synergy amongst these two components, because their combination performs better than the sole usage of any of these two components [53]. In addition, we may notice that both components have played an active part to make possible the synergy amongst them (in Table 6, we may observe that the percentage of individual acceptance due to CD replacement shows values around 60%).

This results along with the suitable behaviour shown by CD/RW (Table 2) suggest that the synergy caused by combining the CD and RW strategies allows this hybrid replacement strategy to have a positive influence on SSGA performance. The RW replacement focuses the search towards the most promising regions, whereas the CD replacement spreads the population to cover a wide group of dissimilar individuals in these zones. Thus, we may draw an important conclusion: hybrid replacement policies become attractive, because they may bring together properties that are needed in an effective replacement strategy, which is difficult to achieve from the use of a single replacement strategy.

We may explain the poor behaviour shown by CD (Table 5) by observing its low variability (Table 6). This indicates that very few chromosomes were fed in the SSGA population, and thus, the evolutionary process might not be accomplished. This aspect was discussed in Section 4.2. Finally, it is interesting to note that although a poor performance is achieved from the

Table 5
CD/RW replacement versus CD and RW replacement strategies

Alg.	f_{sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}	P_{Cheb}	P_{fms}	+	∼	−
CD	4.6e1 +	2.49e4 +	1.0e4 +	2.2e2 +	1.57e1 +	1.02e3 +	2.0e1 +	8.3e4 +	8	0	0
RW	2.2e−04 ∼	2.29e1 ∼	2.9e0 +	1.2e0 +	1.4e−2 +	1.2e2 +	1.4e1 +	7.8e2 +	6	2	0
CD/RW	5e−51	2.05e1	1.0e−1	3.2e−1	7.6e−3	3.7e1	3.9	2.3e2			

Table 6
Variability for CD, RW, and CD/RW

Alg.	f_{sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}	P_{Cheb}	P_{fms}
CD	0.23	0.27	0.28	0.25	0.3	0.35	0.34	0.3
WR	40	44	31	22	8	25	31	36
CD/RW	39 (60.2)	50 (63.8)	14 (61.7)	18 (60.7)	9 (59.2)	16 (60.9)	12 (61.1)	37 (63)

use of the CD replacement, its synergy with the RW replacement allows CD/RW to be the most competitive replacement strategy that has been studied in this paper.

5.4. Other measures of contribution of diversity

In this section, we investigate the influence of the measure of contribution of diversity presented in Section 4.1 on the performance of our proposal. In order to do this, we analyse the results obtained using other measures of contribution of diversity. The first one was introduced in [37] :

$$CD_B(c, P) = d(c, c_{\text{best}}),$$

where c_{best} is the current best performing chromosome in the population. The second has the following formulation:

$$CD_\Sigma(c, P) = \sum_{x \in P} d(c, x),$$

that is, the sum over all pairwise distances between the individual c and the chromosomes in the population P. Table 7 compares the results obtained from the use of these measures with the ones of the original CD/RW algorithm.

We may see in Table 7 that CD_B/RW losses performance with regards to CD/RW on most functions. Therefore, we may claim that the first alternative measure is not suitable for building the proposed replacement strategy. The idea of favouring those elements in the population that are located far from the current best individual is not a good diversification strategy, because it may obstruct the refinement of the best solutions, degrading the quality of the final solutions.

On the other hand, the CD_Σ measure neither allows the operation of our replacement strategy to be adequate. With the use of this measure, each chromosome in the population is forced to be very different from the remaining individuals. For most test problems, this property became too strong to accomplish SSGA convergence. However, for the complex f_{Ros}, this situation of extreme diversity allowed the best fitness value to be reached (18.4) with regards to all the algorithms considered in this paper.

Thus, this results suggest that the original measure of contribution of diversity arises as a very adequate measure for designing the proposed replacement strategy.

5.5. Computational time

In this section, we undertake the analysis of the computational time associated with the CD/RW replacement. In addition, we comment on various implementation details we have followed in order to quicken its application.

The most time-consuming process in the CD/RW replacement concerns the calculation of the measure of contribution of diversity presented in Section 4.1, because it requires the determination of the nearest neighbour of a chromosome in the population. Thus, it would be interesting to find out the number of times this measure is computed during the operation of the replacement strategy. In particular, after the generation of the offspring, the following should be obtained:

(1) The contribution of diversity of the individuals in the population with poorer fitness values than the offspring (Step 1 in Fig. 3).
(2) The contribution of diversity of the offspring (Step 2 in Fig. 3). This step needs N distance calculations (N is the population size).

The first point becomes the main aspect for the study of the computational time of our proposal. We have implemented carefully this procedure, with the aim of relieving, as far as possible, its cost. In order to do this, we have taken advantage of the dynamic of the SSGA: its population undergoes little changes during the evolutionary process. For every individual, c_i, in the population, we store a pointer to its current nearest neighbour and their Euclidean distance (i.e., $CD(c_i, P)$). This information should be updated only when a new individual is introduced in the population:

- First, we find the individuals whose nearest neighbour is the individual that should leave the population and compute their new nearest neighbour after the inclusion of the offspring. In this case, $N - 1$ distance calculations are required for each one of these individuals.
- Then, we should check whether the offspring becomes the new nearest neighbour of the other individuals in the population. This may be made at the same time as the calculation of the contribution of diversity of the offspring.

Table 7
Results using other measures of contribution of diversity

Alg.	f_{sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}	P_{Cheb}	P_{fms}	+	$\sim$	−
CD_B/RW	4.2e−16 $\sim$	2.57e1 +	1.4e1 +	1.3 +	4.0e−2 +	3.9e2 +	4.0e3 +	1.2e1 +	7	1	0
CD_Σ/RW	2.6e−50 +	1.84e1 −	5.3 +	1.0e +	5.1e−2 +	1.1e2 +	1.2e3 +	5.5 $\sim$	6	1	1
CD/RW	5.5e−51	2.05e1	1.0e−1	3.1e−1	7.6e−3	3.6e1	2.3e2	3.8			

Table 8
Average number of chromosomes that updated their nearest neighbours

f_{sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}	P_{Cheb}	P_{fms}
1.06	1.29	1.33	1.13	1.07	1.30	1.31	1.29

Since we know exactly the time spent by the second operation, the most interesting aspect, at this point, is to analyse the average number of chromosomes in the population that updated their corresponding nearest neighbours after the inclusion of an offspring. Table 8 has this information for each one of the test problems considered in our experiments.

In general, we may see that very few individuals needed to change their pointers to the nearest neighbour (close to one). To sum up, we may conclude that our replacement strategy involves approximately $(N-1)+N$ distance calculations after the admission of a new individual in the population (necessary for (1) and (2)). We should also point out that the last occurs with a low frequency (see the variability of the CD/RW algorithm in Table 3).

We should admit that although it is true that our proposal does not take up an excessive computational time, this one is superior to the time spent by the other replacement strategies reviewed in Section 3. However, its excellent performance (Section 5.1) justifies this extra computational effort.

6. Conclusions

This paper presented a replacement strategy for SSGAs that promotes useful diversity by considering the merit of the offspring as solution and its contribution of diversity to the population. It attempts to replace an individual, which performs poorly compared to the offspring, with regard to these two factors. In this way, two benefits are obtained simultaneously; i.e., an increase in population diversity, and an improvement on the quality of the solution. In addition, when this is not possible, it performs the RW strategy to introduce the offspring into the population. In this way, the proposal may be seen as a hybrid replacement procedure, which is composed of two complementary components: a replacement that favours high population diversity levels and another that shows high selective pressure degrees.

The principal conclusions derived from the results of the experiments carried out are the following:

(1) The proposal is a suitable replacement strategy for SSGAs. Its effects on SSGA performance (maintenance of intermediate population diversity levels and production of high variability values) are more decisive than the ones derived from the use of other replacement strategies presented in the GA literature.
(2) The synergy caused by combining the CD replacement and the RW replacement allows the proposal to have its positive influence on SSGA performance. Thus, hybrid replacement strategies become attractive, because they may bring together properties that are needed in an effective replacement policy, which is difficult to achieve from the use of a single replacement strategy.
(3) The measure of the contribution of diversity defined as the distance of the offspring to its nearest neighbour is the best choice for designing the proposed strategy.
(4) The replacement strategy is a determinant issue to take into account for designing effective SSGAs.

Finally, we should point out that extensions of the proposed replacement strategy may be followed in four ways: (1) consider additional features of the chromosomes for deciding which individual should be replaced, such as their age, average distance to their *k* nearest neighbours, distance to a neighbouring subpopulation (in a distributed SSGA framework), etc., (2) design similar replacement strategies for working under other types of coding (such as binary-coding, order-based coding, coding used for the genetic programming, etc.), (3) study the effects of the application of this strategy on non-stationary problems, and (4) implement other instances of hybrid replacement strategies.

Acknowledgement

This research has been supported by the project TIC2002-04036-C05-01.

Appendix A. Test suite

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

A.1. Test functions

We have considered six frequently used test functions: *Sphere* model (f_{Sph}) [10,36], *Generalized Rosenbrock's* function (f_{Ros}) [10], *Schwefel's Problem 1.2* (f_{Sch}) [36], *Generalized Rastringin's* function (f_{Ras}) [1,45], and *Griewangk's* function (f_{Gri}) [19]. Fig. A.1 shows their formulation. The dimension of the search space is 25.

$$\boldsymbol{f_{sph}}$$

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

$$-5.12 \le x_i \le 5.12$$

$$f_{sph}^*(x^*) = 0$$

$$\boldsymbol{f_{Ros}}$$

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

$$-5.12 \le x_i \le 5.12$$

$$f_{Ros}^*(x^*) = 0$$

$$\boldsymbol{f_{Sch}}$$

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

$$-65.536 \le x_i \le 65.536$$

$$f_{Sch}^*(x^*) = 0$$

$$\boldsymbol{f_{Ras}}$$

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

$$a = 10,\ \omega = 2\pi$$

$$-5.12 \le x_i \le 5.12$$

$$f_{Ras}^*(x^*) = 0$$

$$\boldsymbol{f_{Gri}}$$

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

$$d = 4000$$

$$-600.0 \le x_i \le 600.0$$

$$f_{Gri}^*(x^*) = 0$$

Fig. A.1. 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 because 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 [35]. A great part of its difficulty lies in the fact that there are nonlinear interactions between the variables, i.e., it is *non-separable* [50].
- 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, because the gradient of f_{Sch} is not oriented along the axes. It presents similar difficulties to f_{Ros}, but its valley is much narrower.
- f_{Ras} is a scalable, continuous, and multimodal function, which is made from f_{Sph} by modulating it with $a \cdot cos(\omega \cdot x_i)$.
- f_{Gri} is a continuous and multimodal function. This function is difficult to optimize because it is non-separable [30] 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 [50].

A.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* [15], *frequency modulation sounds parameter identification problem* [46], and *polynomial fitting problem* [41]. They are described below.

A.2.1. 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:

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

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

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

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

A.2.2. Frequency modulation sounds parameter identification problem

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

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

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

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

where the model data are given by the following equation:

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

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

A.2.3. 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

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

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

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

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

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

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

```
    **If** $(P_C(1.2) - T_8(1.2) < 0)$ **then** $R \leftarrow R + (P_C(1.2) - T_8(1.2))^2$;
    **If** $(P_C(-1.2) - T_8(-1.2) < 0)$ **then** $R \leftarrow R + (P_C(-1.2) - T_8(-1.2))^2$;
    **Return** $R$;
```

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

Appendix B. Real-parameter genetic operators

In this appendix, we describe the BLX-α crossover operator [14] and the BGA mutation operator [31]. Both are applied to real-coded GAs [21].

B.1. BLX-α operator

Let us assume that $X = (x_1 \ldots x_n)$ and $Y = (y_1 \ldots y_n)$ $(x_i, y_i \in [a_i, b_i] \subset \Re,\ i = 1 \ldots n)$ are two real-coded chromosomes that have been selected to apply the crossover operator to them. BLX-α generates an offspring, $Z = (z_1 \ldots z_n)$, 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 $I = \max_i - \min_i$.

B.2. BGA mutation operator

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

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

where $rang_i$ defines the mutation range and it is normally set to $0.1 \cdot (b_i - a_i)$. The + or $-$ sign is chosen with a probability of 0.5 and $\alpha_i \in \{0, 1\}$ is randomly generated with $p(\alpha_i = 1) = \frac{1}{16}$.

Values in the interval $[c_i - rang_i, c_i + rang_i]$ are generated using this operator, with the probability of generating a neighbourhood of c_i being very high. The minimum possible proximity is produced with a precision of $rang_i \cdot 2^{-15}$.

References

[1] T. Bäck, Self-adaptation in genetic algorithms, in: F.J. Varela, P. Bourgine (Eds.), Proceedings of the First European Conference on Artificial Life, The MIT Press, Cambridge, MA, 1992, pp. 263–271.

[2] J. Branke, M. Cutaia, H. Dold, Reducing genetic drift in steady state evolutionary algorithms, in: W. Banzhaf et al. (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann, San Francisco, CA, 1999, pp. 68–74.

[3] C.R. Bonham, I.C. Parmee, An investigation of exploration and exploitation within cluster oriented genetic algorithms (COGAs), in: Proceedings of the Genetic and Evolutionary Computation Conference 1999, Morgan Kaufmann, San Francisco, California, 1999, pp. 1491–1497.

[4] E. Cantú-Paz, Selection intensity in genetic algorithms with generation gaps, in: D. Whitley, D.E. Goldberg, E. Cantú-Paz, L. Spector, I. Parmee, H.-G. Beyer (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference 2000, Morgan Kaufmann, San Francisco, California, 2000, pp. 911–918.

[5] W. Cedeño, V. Vemuri, T. Slezak, Multi-niche crowding in genetic algorithms and its application to the assembly of DNA restriction-fragments, Evolutionary Computation 2 (4) (1995) 321–345.

[6] W. Cedeño, V. Vemuri, Analysis of speciation and niching in the multi-niche crowding GA, Theoretical Computer Science 229 (1999) 177–197.

[7] K. Chellapilla, Combining mutation operators in evolutionary programming, IEEE Transactions on Evolutionary Computation 2 (3) (1998) 91–96.

[8] K.A. De Jong, J. Sarma, Generation gaps revisited, in: L.D. Whitley (Ed.), Foundations of Genetic Algorithms 2, Morgan Kaufmann, San Mateo, CA, 1993, pp. 19–28.

[9] E.D. De Jong, R.A. Watson, J.B. Pollack. Reducing bloat and promoting diversity using multi-objective methods, in: L. Spector et al., (Ed.), Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco, CA, Morgan Kaufmann, 2001.

[10] K.A. De Jong, An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Ph.D. dissertation, University of Michigan, Ann Arbor, 1975.

[11] G. Dozier, Steady-state evolutionary path planning, adaptive replacement and hyper-diversity, in: M. Schoenauer et al. (Eds.), Parallel Problem Solving from Nature VI, Springer-Verlag, Berlin, 2000, pp. 561–570.

[12] L.J. Eshelman, The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination, in: G.J.E. Rawlin (Ed.), Foundations of Genetic Algorithms 1, Morgan Kaufmann, San Mateo, CA, 1991, pp. 265–283.

[13] L.J. Eshelman, J.D. Schaffer, Preventing premature convergence in genetic algorithms by preventing incest, in: R. Belew, L.B. Booker (Eds.), Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, 1991, pp. 115–122.

[14] L.J. Eshelman, J.D. Schaffer, Real-coded genetic algorithms and interval-schemata, in: L. Darrell Whitley (Ed.), Foundation of Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, 1993, pp. 187–202.

[15] L.J. Eshelman, K.E. Mathias, J.D. Schaffer, Convergence controlled variation, in: R. Belew, M. Vose (Eds.), Foundations of Genetic Algorithms 4, Morgan Kaufmann, San Mateo, CA, 1997, pp. 203–224.

[16] A. Ghosh, S. Tsutsui, H. Tanaka, Function optimization in nonstationary environment using steady state genetic algorithms with aging of individuals, in: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, IEEE Press, New York, 1998, pp. 666–671.

[17] D.E. Goldberg, Genetic Algorithms in Search Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.

[18] D.E. Goldberg, K. Deb, A comparative analysis of selection schemes used in genetic algorithms, in: G.J.E. Rawlins (Ed.), Foundations of Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, 1991, pp. 69–93.

[19] A.O. Griewangk, Generalized descent of global optimization, Journal of Optimization Theory and Applications 34 (1981) 11–39.

[20] G. Harik, Finding multimodal solutions using restricted tournament selection, in: L.J. Eshelman (Ed.), Proceedings of the 6th International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, 1995, pp. 24–31.

[21] F. Herrera, M. Lozano, J.L. Verdegay, Tackling real-coded genetic algorithms: operators and tools for behavioural analysis, Artificial Intelligence Review 12 (4) (1998) 265–319.

[22] Y. Ichikawa, Y. Ishiiolland, Retainig diversity of genetic algorithms for multivariable optimization and neural network learning, in: Proceedings of IEEE International on Conference on Neural Networks, IEEE Press, San Francisco, California, 1993, pp.1110–1114.

[23] C.-Y. Lee, Entropy-Boltzmann selection in the genetic algorithms, IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 33 (1) (2003) 138–149.

[24] T.-H. Li, C.B. Lucasius, G. Kateman, Optimization of calibration data with the dynamic genetic algorithm, Analytica Chimica Acta 2768 (1992) 123–1342.

[25] S.W. Mahfoud, Crowding and preselection revised, in: R. Männer, B. Manderick (Eds.), Parallel Problem Solving from Nature 2, Elsevier, Amsterdam, 1992, pp. 27–36.

[26] S.W. Mahfoud, Niching methods for genetic algorithms. IlliGAL report 95001, University of Illinois at Urbana-Champaign, IL, Illinois Genetic Algorithms Laboratory, 1995.

[27] K. Matsui, New selection method to improve the population diversity in genetic algorithms, in: Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings, 1999 IEEE International Conference, vol. 1 , 12–15 October 1999, pp. 625–630.

[28] O.J. Mengshoel, D.E. Goldberg, Probabilistic crowding: deterministic crowding with probabilistic replacement, in: W. Banzhaf et al. (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99, Morgan Kaufmann Publishers, San Francisco, CA, 1999, pp. 409–416.

[29] N. Mori, J. Yoshida, H. Tamaki, H. Kita, Y. Nichikawa, A thermodynamical selection rule for genetic algorithms, in: Proceedings of the 2nd IEEE Conference on Evolutionary Computation (ICEC'95), IEEE Press, New York, 1995, pp. 188–192.

[30] H. Mühlenbein, M. Schomisch, J. Born, The parallel genetic algorithm as function optimizer, in: R. Belew, L.B. Booker (Eds.), Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, 1991, pp. 271–278.

[31] H. Mühlenbein, D. Schlierkamp-Voosen, Predictive models for the breeder genetic algorithm I. Continuous parameter optimization, Evolutionary Computation 1 (1993) 25–49.

[32] A. Rogers, A. Prügel-Bennett, Modelling the dynamics of a steady state genetic algorithm, in: W. Banzhaf, C. Reeves (Eds.), Foundations of Genetic Algorithms 5, Morgan Kaufmann, San Francisco, CA, 1999, pp. 57–68.

[33] B. Sareni, L. Krähenbühl, Fitness sharing and niching methods revisited, IEEE Transactions on Evolutionary Computation 2 (3) (1998) 97–106.

[34] T.A. Sedbrook, H. Wright, R. Wright, Application of a genetic classifier for patient triage, in: R. Belew, L. Booker (Eds.), Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, 1991, pp. 334–338.

[35] D. Schlierkamp-Voosen, H. Mühlenbein, Strategy adaptation by competing subpopulations, in: Y. Davidor, H.-P. Schwefel, R. Männer (Eds.), Parallel Problem Solving from Nature 3, Springer-Verlag, Berlin, 1994, pp. 199–208.

[36] H.-P. Schwefel, Numerical Optimization of Computer Models, Wiley, Chichester, 1981.

[37] H. Shimodaira, A diversity control oriented genetic algorithm (DCGA): development and experimental results, in: Proceedings of the Genetic and Evolutionary Computation Conference 1999, Morgan Kaufmann, San Francisco, California, 1999, pp. 603–611.

[38] H. Shimodaira, A new genetic algorithm using large mutation rates and population-elitist selection (GALME), in: Proceedings of the International Conference on Tools with Artificial Intelligence, IEEE Computer Society, 1996, pp. 25–32.

[39] J. Smith, F. Vavak, Replacement strategies in steady state genetic algorithms: static environments, in: W. Banzhaf, C. Reeves (Eds.), Foundations of Genetic Algorithms, Morgan Kaufmann, San Francisco, CA, 1999, pp. 219–233.

[40] I. Stadnyk, Schema recombination in pattern recognition problems, in: J.J. Grefenstette (Ed.), Proceedings of the Second International Conference on Genetic Algorithms, Lawrence Erlbaum, Hillsdale, NJ, 1987, pp. 27–35.

[41] R. Storn, K. Price, Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA, 1995.

[42] G. Syswerda, Uniform crossover in genetic algorithms, in: J. David Schaffer (Ed.), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, 1989, pp. 2–9.

[43] D. Thierens, Selection schemes, elitist recombination, and selection intensity, in: T. Bäck (Ed.), Proceedings of the Seventh International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, 1997, pp. 152–159.

[44] A. Toffolo, E. Benini, Genetic diversity as an objective in multi-objective evolutionary algorithms, Evolutionary Computation 11 (2) (2003) 151–167.

[45] A. Törn, Ž. Antanas, Global Optimization, Lecture Notes in Computer Science 350, vol. 350, Springer, Berlin, 1989.

[46] S. Tsutsui, Y. Fujimoto, Forking genetic algorithm with blocking and shrinking modes, in: S. Forrest (Ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, 1993, pp. 206–213.

[47] F. Vavak, T.C. Fogarty, Comparison of steady state and generational genetic algorithms for use in nonstationary environments, in: Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, IEEE Press, New York, 1996, pp. 192–195.

[48] J. Wakunda, A. Zell, Median-selection for parallel steady-state evolution strategies, in: Proceedings of the Parallel Problem Solving from Nature — PPSN VI, Lecture Notes in Computer Science, vol. 1917, Springer, 2000, pp. 405–414.

[49] D. Whitley, The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best, in: J. David Schaffer (Ed.), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, 1989, pp. 116–121.

[50] D. Whitley, S. Rana, J. Dzubera, E. Mathias, Evaluating evolutionary algorithms, Artificial Intelligence 85 (1996) 245–276.

[51] K. Wiese, S.D. Goodwin, Convergence characteristics of keep-best reproduction, Selected Areas in Cryptography (1999) 312–318.

[52] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 67–82.

[53] H.-S. Yoon, B.-R. Moon, An empirical study on the synergy of multiple crossover operators, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 212–223.