Replacement Strategies to Maintain Useful Diversity in Steady-State Genetic Algorithms

Manuel Lozano[1], Francisco Herrera[1] and José Ramón Cano[2]

[1] Dept. of Computer Science and Artificial Intelligence, University of Granada,
18071-Granada, Spain
lozano@decsai.ugr.es, herrera@decsai.ugr.es

[2] Dept. of Electronic Engineering, Computer Systems and Automatics, Escuela Superior de
La Rábida, University of Huelva, 21819-Huelva, Spain
jose.cano@diesia.uhu.es

Summary. In this paper, we propose a replacement strategy for steady-state genetic algorithms that takes into account two features of the element to be included into the population: a measure of the contribution of diversity to the population and the fitness function. In particular, the proposal attempts to replace an element in the population with worse values for these two features. In this way, the diversity of the population is increased and the quality of its solutions is improved, simultaneously, maintaining high levels of useful diversity. Experimental results show that the use of the proposed replacement strategy allows significant performance to be achieved for problems with different difficulties, which regards to other replacement strategies presented in the literature.

1 Introduction

There are two primary factors in the search carried out by an genetic algorithm (GA): population diversity and selective pressure [30]. 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 favored over non-promising regions. On the other hand, population diversity is crucial to a GA's ability to continue the fruitful exploration of the search space [16]. 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 [6, 10].

Selective pressure and population diversity are inversely related [30]. Increasing selective pressure results in a faster loss of population diversity, while maintaining population diversity offsets the effect of increasing selective pressure. These two

Manuel Lozano et al.: *Replacement Strategies to Maintain Useful Diversity in Steady-State Genetic Algorithms*, Advances in Soft Computing **1**, 85–96 (2005)
www.springerlink.com

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 [18].

In *steady-state* GAs (SSGAs) usually only one or two offspring are produced 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, since parents and offspring compete for survival. Different studies have shown that improved performance in SSGAs with regards to generational GAs (where entire population is replaced every generation by the offspring population) is due to their higher selection pressure and changes in the exploration/exploitation balance caused by using different parent selection and replacement strategies and is not due to the use of an overlapping model [4]. This justifies the study on SSGAs, and in particular, the design of effective replacement strategies having the aim of propitiating useful diversity.

In this paper, we propose a replacement strategy that takes into account two features of the element to be included into the population: a measure of the contribution of diversity to the population and the fitness function. It attempts to replace an element in the population with worse values for these two features. In this way, the diversity of the population is increased and the quality of its solutions is improved, simultaneously. The goal of this strategy is to protect those individuals that allow the highest levels of useful diversity to be maintained.

The paper is set up as follows. In Sect. 2, the SSGA model is described. In Sect. 3, different types of replacement strategies presented in the literature for promoting SSGA population diversity are explained. In Sect. 4, we propose the replacement strategy based on the contribution of diversity. In Sect. 5, different empirical experiments are carried out for studying the performance of this strategy. Finally, in Sect. 6, some conclusions are pointed out.

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 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 of the population will be replaced by the new offspring. The basic algorithm step of SSGA is the following [30]:

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.

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 chose 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. In the paper, this strategy will be called *standard replacement strategy*. In [11], it was suggested that the deletion of the worst individuals induced a high selective pressure, even when the parents were selected randomly.

3 Favouring Population Diversity Throughout Replacement Strategies

There are different replacement strategies that attempt to preserve population diversity. Most of them are instances of the *crowding* methods [5]. 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. They have been used to locate and preserve multiple local optimum in multimodal problems.

An effective crowding method is the *restricted tournament selection* (RTS) [13]. 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 w (window size) more members of the population and picks the individuals that most closely resemble A' from those w elements. A' then competes with this element, and if A' wins, it is allowed to enter the population.

Another type of crowding methods assume that the parents would be ones 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* [17], *elitist recombination* [25], and *keep-best reproduction* [32].

- In deterministic crowding, 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 $O_j$.
```

  where $f(\cdot)$ is the fitness function and $d(\cdot,\cdot)$ is a distance measure between two chromosomes.
- In elitist recombination, the best two of these four individuals (parents and offspring) go to the next generation.

- Finally, keep-best keeps the best parent and the best offspring in order to introduce good new genetic material into the population.

In addition, two replacement strategies for SSGAs have been designed for encouraging diversity in the population: the *first-in-first-out* (FIFO) strategy [9, 28] and the *conservative* strategy [29].

- In the FIFO strategy, the element to be replaced is the oldest in the population. In its elitist form, the replacement of the current best element is not allowed.
- The *conservative* strategy combines a FIFO replacement strategy with a modified deterministic binary tournament selection operator. In this case, a tournament is carried out between the oldest element in the population and another member picked at random. The worse is replaced. In this way, elitism is implicitly assured.

4 The Proposed Replacement Strategy

The population diversity versus selective pressure problem may be consider as a biobjective problem. The GA may be used for searching simultaneously the best chromosomes together with the ones that provide more diversity to the GA population, i.e., promoters of useful diversity.

In fact, there are some generational GA approaches that use this idea for avoiding premature convergence and improving GA behavior [15, 19, 23]. During the selection phase, they use the fitness function and some type of measures for evaluating the diversity introduced by the candidate chromosomes with the aim of favoring those individuals with both high fitness function values and high diversity contributions.

In this paper, we propose to solve the diversity versus selective pressure problem in the replacement phase of a SSGA. In particular, we present a replacement strategy based on the contribution of diversity of the offspring to the population where it will be included. The idea is to replace an element of the population with a worse fitness function value than the one of the offspring, and with a lower contribution of diversity than the one provided by the offspring. In this way, we deal with two underlying objectives simultaneously: to optimize the fitness function and to enhance population diversity, i.e., to promote useful diversity.

In Subsect. 4.1, we present measures of the contribution of diversity of an element to an SSGA population, and in Subsect. 4.2, we use these measures for defining a replacement strategy that tackles the biobjective diversity-convergence.

4.1 Contribution of Diversity

Next, we present two different types of measures of the contribution of diversity of a chromosome x to a population P ($x \notin P$), $ConDiv(x, P)$. They quantify the importance of a chromosome as promoter of diversity in the population.

- Measures based on distance. They are defined as the distance of x to a particular member of P. We have chosen two possibilities: 1) this member is the best one, and 2) it is the nearest neighbor of x in P.

  These two measures are the ones used in [23] and [3], respectively, and they will be denoted as *ConDiv-B* and *ConDiv-NN*, respectively.

  Under these definitions, a chromosome will provide a great contribution of diversity when it is very different from all the other chromosomes (*ConDiv-NN*) or from the best one (*ConDiv-B*).
- Measures based on population diversity. Now, we define the contribution of diversity as follows:

$$ConDiv(x,P) = Div(P \cup \{x\}) - Div(P) ,$$

  where $Div(P)$ is a measure of the diversity in the population P. Measures of this type are to be found in [2, 18].

  In this case, the contribution of diversity of a chromosome will be high when its inclusion in the population allows a significant increasing of diversity to be achieved. In this paper, we employ the following diversity measure for a population with real-coded chromosomes, $C_j = (c_1^j, \cdots c_n^j), j = 1, \cdots N$, which is called *average interval usage*:

$$AIU(P) = \frac{1}{n} \sum_{i=1}^{n} \frac{Max_i - Min_i}{b_i - a_i}$$

  where $Max_i = \max_{j=1,\ldots N}\{c_i^j\}, Min_i = \min_{j=1,\ldots N}\{c_i^j\}$, and a_i and b_i ($a_i < b_i$) are the extremes of the action interval of the gene i ($i = 1, \cdots, n$). The measure of the contribution of diversity derived from AIU will be called *ConDiv-AIU*.

4.2 The Proposed Replacement Strategy

Now, let us assume that an offspring o is returned from the recombination phase of an SSGA, then we propose to include o in the population by means of the following strategy:

1. Find the element z in P that fulfils:
   (i) $f(o)$ is better than $f(z), f$ being the fitness function.
   (ii) $\Delta_z > \Delta_w > 0$, for all w in P, where for any $t \in P$:

$$\Delta_t = ConDiv(o, P - \{t\}) - ConDiv(t, P - \{t\}) .$$

2. If z does not exist, then apply the standard replacement strategy (Sect. 2).

First, this strategy attempts to find an element in the population with worse fitness than the offspring and whose contribution of diversity is the lowest with regards to the one provided by the offspring in a population where this element have been removed. This element will be replaced by the offspring. In this way, we introduce into the population an element with better fitness function and higher contribution of diversity, increasing the population diversity and improving the quality of its solutions.

If this individual does not exist, then the standard replacement strategy is applied. In this way, we ensure that the best elements found during the run are included in the population (elitist strategy).

5 Experiments

Minimization experiments on the test suite described in Sect. 5.1 were carried out in order to study the behavior of the replacement strategy presented in the previous section.

In Sect. 5.2, we propose different SSGA instances based on this strategy and compare its results with the ones of a standard SSGA. Finally, in Sect. 5.3, we compare the best SSGA of Sect. 5.2 with different SSGAs based on the replacement strategies described in Sect. 3.

5.1 Test Suite

The test suite that we have used for the experiments consists of six test functions and three real-world problems. The test functions are the following: *Sphere* model (f_{Sph}) [5, 22], *Generalized Rosenbrock's* function (f_{Ros}) [5], *Schwefel's Problem 1.2* (f_{Sch}) [22], *Generalized Rastringin's* function (f_{Ras}) [1, 26], *Griewangk's* function (f_{Gri}) [12], and *Expansion of* f_{10} (ef_{10}) [31]. The dimension of the search space is 10 for ef_{10} and 25 for the remaining test functions. The three real-world problems are the following: *Systems of Linear Equations* [8], *Frequency Modulation Sounds Parameter Identification Problem* [27], and *Polynomial Fitting Problem* [24]. For all the problems, the global optimum has a fitness value of zero.

5.2 SSGAs Based on the Proposed Replacement Strategy

We have implemented three instances of SSGA based on the replacement strategy proposed in Subsect. 4.2. They are called SGA-ConDiv-B, SGA-ConDiv-NN, and SGA-ConDiv-AIU, and use the *ConDiv-B*, *ConDiv-NN*, and *ConDiv-AIU* measures of the contribution of diversity defined in Subsect. 4.1, respectively. We compare these algorithms against a SSGA that uses the standard replacement strategy (see Sect. 2), which is called SGA-Standard.

All the algorithms use real coding [14] and apply the BGA mutation operator [20] and the BLX-α operator ($\alpha = 0.5$) [7]. The mutation probability is $\frac{1}{n}$ and the population size is 60 chromosomes. Parents are selected at random. They were executed 50 times, each one with a maximum of 100,000 evaluations. Table 1 shows the results obtained. The performance measures utilized are the following:

- **A** performance: average of the best fitness function found at the end of each run.
- **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.

Table 1. Results for SSGAs based on the proposed replacement strategy

	f_{sph}		f_{Ros}		f_{Sch}	
Algorithm	A	B	A	B	A	B
SGA-ConDiv-AIU	1.2e−15 ∼	2.9e−54	2.4e+01 ∼	1.0e+01	3.4e−01 −	2.9e−02
SGA-ConDiv-NN	4.3e−50 ∼	7.5e−52	2.0e+01 ∼	1.9e+01	5.9e−02 −	6.9e−03
SGA-ConDiv-B	1.1e−32 ∼	1.0e−41	2.7e+01 ∼	1.6e+01	4.3e+00 ∼	3.0e−01
SGA-Standard	2.2e−04	1.1e−55	2.3e+01	1.3e+01	2.9e+00	1.5e−01

	f_{Ras}		f_{Gri}		ef_{10}	
Algorithm	A	B	A	B	A	B
SGA-ConDiv-AIU	6.8e−01 −	52.0 %	1.4e−03 −	62.0 %	6.8e−03 ∼	6.2e−35
SGA-ConDiv-NN	4.4e−01 −	56.0 %	3.5e−04 −	86.0 %	4.8e−26 −	7.4e−27
SGA-ConDiv-B	1.1e+00 ∼	30.0 %	9.6e−03 +	38.0 %	5.3e−03 ∼	1.8e−27
SGA-Standard	1.2e+00	32.0 %	4.3e−03	4.0 %	6.3e−03	8.5e−37

	P_{fms}		P_{sle}		P_{Cheb}	
Algorithm	A	B	A	B	A	B
SGA-ConDiv-AIU	7.3e+02 ∼	1.3e+02	1.4e+02 ∼	1.8e+01	1.5e+01 ∼	6.5e−07
SGA-ConDiv-NN	2.9e+02 −	2.8e+00	3.1e+01 −	5.7e−01	1.5e+00 −	2.0e−26
SGA-ConDiv-B	2.9e+03 +	1.3e+02	3.1e+02 +	5.1e+01	1.3e+01 ∼	2.2e−10
SGA-Standard	7.8e+02	1.2e+02	1.2e+02	1.2e+01	1.4e+01	3.2e−10

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 *A* performance for the standard SSGA are significant when compared against the ones for the SSGAs based on the proposed replacement strategy. The direction of any significant differences is denoted either by:

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

Taking into consideration these results, we would make the following comments:

- SGA-ConDiv-NN returns better *A* results than SGA-Standard (see t-test results) on all test functions except on the unimodal f_{sph} and f_{Ros}, where they are similar. Moreover, SGA-ConDiv-NN achieves a better *B* measure than SGA-Standard for the more complex functions: f_{Sch}, P_{Cheb},P_{fms}, and P_{sle}.
  SGA-ConDiv-AIU provides better *A* results than SGA-Standard on the complex f_{Sch} and on the multimodal f_{Ras} and f_{Gri}. For the remaining function, their performance is similar.

These facts show that using the proposed replacement strategy, the results may be improved with regards to a standard SSGA. This means that its formulation (which has the aim of propitiate useful diversity) allows a profitable operation to be achieved. Furthermore, the measure of contribution of diversity *ConDiv-NN* arises as a very adequate measure for designing this strategy.

- On the other hand, SGA-ConDiv-B losses performance with regards to SGA-Standard on f_{Gri},P_{sle}, and P_{fms}. Therefore, we may claim that the measure ConDiv-B is not suitable for building the proposed replacement strategy. The idea of favoring those elements in the population that are located far away 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.

5.3 Comparison with Other Replacement Strategies

In this section, we compare the best algorithm based on the proposed replacement strategy, SGA-ConDiv-NN, against different SSGAs that use the replacement strategies described in Sect. 3. Table 2, shows these algorithms along with their main features.

Table 2. Algorithms based on other replacement strategies

Algorithm	Replacement Strategy
SGA-RTS	Restricted tournament selection ($\omega = 5$)
SGA-DC	Deterministic crowding
SGA-ER	Elitist recombination
SGA-KBR	Keep-best reproduction
SGA-FIFO	First-in-first-out
SGA-Conse	Conservative strategy

Table 3 has the results. A *t*-test was applied in order to ascertain if differences in the *A* performance for SGA-ConDiv-NN are significant when compared against the one for the other algorithms.

Table 4 was introduced in order to analyse the results. It shows the number of improvements, reductions, and non differences (according to the t-test results in Table 3) in the *A* performance for SGA-ConDiv-NN with regards to the one for the other algorithms.

The table shows that SGA-ConDiv-NN outperforms all the other algorithms, because it improves their results on a great number of functions and never achieves a worse *A* performance measure. The source of diversity for these algorithms is the same than for SGA-ConDiv-NN since they use the same mating and recombination strategies. The difference between them is the way in which diversity is retained by the replacement strategies. The promising results of the proposal reveal that it

Table 3. Results for the comparison

	f_{sph}		f_{Ros}		f_{Sch}	
Algorithm	A	B	A	B	A	B
SGA-RTS	9.2e−17 +	5.7e−20	2.1e+01 ∼	1.5e+01	1.6e+01 +	2.1e+00
SGA-DC	9.4e−32 +	6.5e−33	2.1e+01 +	2.0e+01	1.2e+01 +	3.8e+00
SGA-ER	8.0e−46 +	2.3e−47	2.1e+01 ∼	2.0e+01	3.8e−01 +	7.6e−02
SGA-KBR	2.8e−12 ∼	6.2e−14	2.3e+01 ∼	1.5e+01	1.4e−01 +	1.9e−02
SGA-FIFO	1.9e−02 +	9.3e−12	2.5e+01 +	2.1e+00	5.5e+01 +	3.6e−01
SGA-Conse	7.0e−03 ∼	2.2e−12	3.2e+01 +	4.5e+00	1.2e+02 +	7.8e−01
SGA-ConDiv-NN	4.3e−50	7.5e−52	2.0e+01	1.9e+01	5.9e−02	6.9e−03

	f_{Ras}		f_{Gri}		ef_{10}	
Algorithm	A	B	A	B	A	B
SGA-RTS	6.8e−01 ∼	3.3e−10	6.6e−03 +	1.1e−18	1.4e−02 +	1.3e−09
SGA-DC	2.2e+01 +	3.2e+00	4.3e−08 ∼	90.0 %	9.6e−15 +	3.0e−15
SGA-ER	2.6e+00 +	8.0 %	8.9e−04 ∼	84.0 %	4.7e−25 +	1.3e−25
SGA-KBR	5.4e−01 ∼	1.9e−10	2.0e−03 ∼	6.6e−11	8.8e−03 +	1.8e−03
SGA-FIFO	1.0e+00 +	1.2e−08	3.7e−02 +	8.7e−09	4.5e−01 +	4.0e−06
SGA-Conse	1.2e+00 +	2.8e−04	4.3e−02 +	1.5e−08	3.7e−01 +	3.1e−11
SGA-ConDiv-NN	4.4e−01	56.0 %	3.5e−04	86.0 %	4.8e−26	7.4e−27

	P_{fms}		P_{sle}		P_{Cheb}	
Algorithm	A	B	A	B	A	B
SGA-RTS	5.7e+02 +	6.1e+01	7.0e+01 +	6.7e+00	4.8e+00 +	4.6e−15
SGA-DC	2.8e+02 ∼	3.4e+01	2.4e+01 ∼	2.5e+00	3.9e+00 +	2.4e−05
SGA-ER	3.2e+02 ∼	2.1e+01	5.1e+01 +	1.3e+00	3.4e+00 ∼	3.0e−17
SGA-KBR	3.3e+02 ∼	5.5e+01	6.9e+01 +	3.0e+00	7.9e+00 +	1.1e−15
SGA-FIFO	1.6e+03 +	3.6e+02	2.2e+02 +	4.1e+01	1.7e+01 +	5.8e−07
SGA-Conse	1.5e+03 +	1.5e+02	2.5e+02 +	2.6e+01	1.8e+01 +	4.0e−04
SGA-ConDiv-NN	2.9e+02	2.8e+00	3.1e+01	5.7e−01	1.5e+00	2.0e−26

Table 4. SGA-ConDiv vs. the other algorithms

Algorithm	Num. Impr. (+)	Num. Red. (−)	Num. Non Diff. (∼)
SGA-RTS	7	0	2
SGA-DC	6	0	3
SGA-ER	5	0	4
SGA-KBR	5	0	4
SGA-FIFO	9	0	0
SGA-Conse	8	0	1

conserves those elements (the more diverse and with the higher quality) that allow the search process to be driven towards better regions than by means of the other strategies. Thus, we conclude that it is an adequate replacement strategy for SSGAs.

6 Conclusions

This paper presented a replacement strategy for SSGAs that attempts to promote 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 that is worse than the offspring with regards to these two factors, obtaining two benefits simultaneously: an increase of the population diversity and an improvement on the quality of its solutions. The principal conclusions derived from the results of the experiments carried out are the following:

1. The ConDiv-NN measure of the contribution of diversity is the best choice for designing the proposed strategy. However, other measures (such as ConDiv-AIU) allow the effectiveness of the strategy to be exhibited as well, with regards to a standard SSGA.
2. The proposal is a suitable replacement strategy for SSGAs. Its effects on SSGA performance are more decisive than the ones derived from the use of other replacement strategies presented in the GA literature.
3. These results confirm that 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 codings, codings used for the genetic programming, etc.), and 3) study the effects of the application of this strategy on nonstationary problems.

Acknowledgments

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

References

1. Bäck T (1992) Self-adaptation in genetic algorithms. In: F.J. Varela, P. Bourgine, (eds) Proc. of the First European Conference on Artificial Life. The MIT Press, Cambridge, MA, 263–271.
2. Bedau MA, Ronneburg F, Zwick M (1992) Dynamic of diversity in an evolving population. In: R. Männer, B. Manderick, (eds) Parallel Problem Solving from Nature 2. Elsevier Science Publichers, Amsterdam, 95–104.
3. Bonham CR, Parmee IC (1999) An investigation of exploration and exploitation within cluster oriented genetic algorithms (COGAs). In: Proc. of the Genetic and Evolutionary Computation Conference 1999. Morgan Kaufmann, San Francisco, California, 1491–1497.

4. De Jong KA, Sarma J (1993) Generation gaps revisited. In: L.D. Whitley (ed) Foundations of Genetic Algorithms 2. Morgan Kaufmann, San Mateo, CA, 19–28.
5. De Jong KA (1975) An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Ph. dissertation, Univ. of Michigan, Ann Arbor, 1975.
6. Eshelman LJ, Schaffer JD (1991) Preventing premature convergence in genetic algorithms by preventing incest. In: R. Belew, L.B. Booker, (eds) Proc. of the Fourth Int. Conf. on Genetic Algorithms. Morgan Kaufmann, San Mateo, CA, 115–122.
7. Eshelman LJ, Schaffer JD (1993) Real-coded genetic algorithms and interval-schemata. In: L. Darrell Whitley (ed) Foundation of Genetic Algorithms 2. Morgan Kaufmann Publishers, San Mateo, 187–202.
8. Eshelman LJ, Mathias KE, Schaffer JD (1997) Convergence controlled variation. In: R. Belew, M. Vose, (eds) Foundations of Genetic Algorithms 4. Morgan Kaufmann, San Mateo, CA, 203–224.
9. Ghosh A, Tsutsui S, Tanaka H (1998) Function optimization in nonstationary environment using steady state genetic algorithms with aging of individuals. In: Proc. of the 1998 IEEE International Conference on Evolutionay Computation. IEEE Press, New York, 666–671.
10. Goldberg DE (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA.
11. Goldberg DE, Deb K (1991) 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, 69–93.
12. Griewangk AO (1981) Generalized descent of global optimization. Journal of Optimization Theory and Applications 34: 11–39.
13. Harik G (1995) Finding multimodal solutions using restricted tournament selection. In: Proc. 6th Int. Conf. Genetic Algorithms, L. J. Eshelman (Ed.), (Morgan Kaufmann, San Mateo, CA, 1995) 24–31.
14. Herrera F, Lozano M, Verdegay JL (1998) Tackling real-coded genetic algorithms: operators and tools for behavioural analysis. Artificial Intelligence Review 12(4): 265–319.
15. Lee C-Y (2003) Entropy-boltzmann selection in the genetic algorithms. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 33(1): 138–149.
16. Li T-H, Lucasius CB, Kateman G (1992) Optimization of calibration data with the dynamic genetic algorithm. Analytica Chimica Acta 2768: 123–1342.
17. Mahfoud SW (1992), Crowding and preselection revised. In: R. Männer, B. Manderick, (eds) Parallel Problem Solving from Nature 2. (Elsevier, Amsterdam, 27–36.
18. Mahfoud SW (1995) Niching methods for genetic algorithms. IlliGAL report 95001, University of Illinois at Urbana-Champaign, IL, Illinois Genetic Algorithms Laboratory.
19. Mori N, Yoshida J, Tamaki H, Kita H, Nichikawa Y (1995) A thermodynamical selection rule for genetic algorithms. In: Proc. of the 2nd IEEE Conference on Evolutionary Computation (ICEC'95). IEEE Press, New York, 188–192.
20. Mühlenbein H, Schlierkamp-Voosen D (1993) Predictive models for the breeder genetic algorithm I. continuous parameter optimization. Evolutionary Computation 1: 25–49.
21. Schlierkamp-Voosen D, Mühlenbein H (1994) Strategy adaptation by competing subpopulations. In: Y. Davidor, H.-P. Schwefel, R. Männer (eds) Parallel Problem Solving from Nature 3. Springer-Verlag, Berlin, 199–208.
22. Schwefel H-P (1981) Numerical optimization of computer models. Wiley, Chichester.
23. Shimodaira H (1999) A diversity control oriented genetic algorithm (DCGA): development and experimental results. In: Proc. of the Genetic and Evolutionary Computation Conference 1999. Morgan Kaufmann, San Francisco, California, 603–611.

24. Storn R, Price K (1995) 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.
25. Thierens D (1997) Selection schemes, elitist recombination, and selection intensity. In: T. Bäck, (ed) Proc. of the Seventh Int. Conf. on Genetic Algorithms. Morgan Kaufmann Publishers, San Mateo, 152–159.
26. Törn A, Antanas Ž (1989) Global optimization. Lecture Notes in Computer Science 350, Springer, Berlin.
27. Tsutsui S, Fujimoto Y (1993) Forking genetic algorithm with blocking and shrinking modes. In: S. Forrest (ed) Proc. of the Fifth Int. Conf. on Genetic Algorithms. Morgan Kaufmmann, San Mateo, CA, 206–213.
28. Vavak F, Fogarty TC (1996) Comparison of steady state and generational genetic algorithms for use in nonstationary environments. In: Proc. Of the 1996 IEEE International Conference on Evolutionary Computation. IEEE Press, New York, 192–195.
29. Vavak F (1997) Genetic algorithm based self-adaptive techniques for direct load balancing in nonstationary environments. PhD thesis, University of the West of England.
30. Whitley D. (1989), The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best. In: J. David Schaffer (ed) Proc. of the Third Int. Conf. on Genetic Algorithms. Morgan Kaufmann Publishers, San Mateo, 116–121.
31. Whitley D, Beveridge R, Graves C, Mathias K (1995) Test driving three 1995 genetic algorithms: new test functions and geometric matching. Journal of Heuristics 1: 77–104.
32. Wiese K, Goodwin SD (1999) Convergence characteristics of keep-best reproduction. In: Selected Areas in Cryptography, 312–318.