The Shifting Balance Genetic Algorithm as more than just another Island Model GA

Mark Wineberg

University of Guelph
Computer and Information Science
wineberg@cis.uoguelph.ca

Jun Chen

University of Guelph
Computer and Information Science
jun@uoguelph.ca

Category: Genetic Algorithm

Abstract. The Shifting Balance Genetic Algorithm (SBGA) is an extension of the Genetic Algorithm (GA) that was created to promote guided diversity to improve performance in highly multimodal environments. In this paper a new behavioral model for the SBGA is presented. Based on the model, various modifications of the SBGA are proposed: these include a mechanism for managing dynamic population sizes along with population restarts. The various mechanisms that make up the SBGA are compared and contrasted against each other and against other Island Model GA systems. It was found that the mechanisms that characterize the SBGA, such as a repulsive central force from one population on the others, could improve the behavior of multi-populational systems.

1 Introduction

While the GA has proven to be very effective in various problems, the GA still can get trapped at local optima, which is known in the GA literature as premature convergence. To a great degree overcoming local optima and lessening premature convergence can be seen as the problem of diversity maintenance. Many mechanisms are developed to help the GA to maintain diversity in the population such as Restricting mating practice [1][2], adaptive mutation rate or selection pressure [3], random immigrant [4], restart / re-initialization mechanism [5] and multiple population solution [6] [7].

It would be useful if the diversity can be added in a guided fashion; less blindly. Blindly increasing the diversity can cause the GA to experience random drift in the search area and consequently lose its search efficiency. An algorithm has already been designed with these issues in mind: the Shifting Balance Genetic Algorithm (SBGA) [16]. The SBGA is a multi-populational GA (similar to many parallel GAs), but with mechanisms introduced to promote directed diversity.

The SBGA when introduced was designed with user chosen parameters for both the number and size of the populations. These were chosen based on an initial model of how the SBGA behaved and was never investigated further. In this paper we propose a new analysis of the behavior of the SBGA and show that modifications inspired by the new analysis improve the SBGA's ability to handle multimodal environments. We then examine the various mechanisms in the SBGA, such as the repulsive central force from the core population that act on the colony populations, and see if they are effective when added to a general multi-populational system such as an Island Model GA with a bi-directional ring topology.

2 Diversity and Adaptive Population Size

Maintaining Diversity

There are different ways measuring diversity, e.g. *pair-wise distances diversity, informational entropy diversity*. Maintaining diversity usually can be done in two ways: either by decelerating the process of gene fixation or by reintroducing the diversity after it has been lost. Common diversity maintenance or enhancing techniques are increasing mutation, preventing inbreeding, decreasing selection pressure, restarting the populations, and the use of multiple populations.

One technique for introducing diversity is simply increasing mutation. Grefenstette [4] introduced the idea of *random immigrants* into the population. Here randomly generated individuals are inserted into the population to be a source of random crossover segments.

Inbreeding promotes diversity loss. Consequently, some restricted mating practices such as using species identity tag and inbreeding prohibitions are used [2]. These techniques can monitor individual gene makeup, fitness, or mating history.

Decreasing the selection pressure is also another way to help maintain diversity. There are many different ways to do this, the simplest one is the reduction of the slope used in linear rank selection.

Re-initialization of the population, in whole or in part according to some criteria or policy can reintroduce diversity after it has been lost. In practice the criteria used can include: cost bounds, time bounds, improvement probability bounds and convergence bounds, along with various combinations [5].

A multi-populational GA (also called the Island Model) localizes members into sub-populations, called islands, where each island can potentially follow a different search trajectory through the search space and develop in more or less isolated niches [9].

Adaptive Population Size

There has been very little work on adaptive population size schemes, despite the fact that the population size is a critical parameter in a GA and it largely determines the quality of the solution [10]. The limited amount of work done can be broken down into two groups: direct evolution of the population size parameter [14]; and mechanisms that indirectly affect the population size, which include GAVaPS [11], SOS [13] and BGA [12]. We will be using ideas from the last approach so we will go into further detail for that system.

Schilerkamp-Voose and Mühlenbein [12] use subpopulations competition that changes the sizes of subpopulation in their breeder genetic algorithm (BGA). There the total number of individuals is fixed, whereas the size of a single group varies. In every generation the subpopulations compete based on the fitness of the best of individual of the group. At regular generation intervals a quality criterion is used on each subpopulation; the group with best quality increases the size and all other groups are decreased. A lower limit is used for the size of each group, so no population is lost. At a later generation within the interval, the best individual from each subpopulation emigrates to all populations and the best fitness of the groups becomes equal. We will borrow these ideas for the SBGA when we introduce subpopulation competition among colonies (a colony is a specialized subpopulation), although the mechanism will be slightly modified.

3 Shifting Balance Genetic Algorithm

Introduction

The Shifting Balance Genetic Algorithm was originally designed to use multiple populations to solve moving peak problems in dynamic environment; but it also can help GA as a function optimizer to overcome local optima or lessen the premature convergence in multimodal environments. Since this paper re-examines the behavior of the SBGA and improves on the algorithm, a brief overview of the system is presented below. For a more detailed explanation of the SBGA, see [15] [16].

Core group and Colonies

SBGA is a multiple population GA similar to the island model. But unlike other island model GAs it does not divide the population to several demes with the same size. In SBGA the populations are classified into two categories: A large central population, called the core and a series of smaller, helping population called colonies.

The core is responsible for exploring the area of the search space that seems most promising and performing exploitation through receiving the elite immigrants from the colonies, while the colonies explore the areas of the search space where the core population does not yet have a presence. The core has a large population size so that it has an increased ability to do search or perform hill climbing since the more members the GA has, the more children it can produce and hence generate more trials in that area.

The colony is set to have a smaller population size in order for each colony's gene make-up to more easily change, however this also can cause a reduced search power.

Selecting for Distance from the Core

For the SBGA to perform as desired, a mechanism is needed that allows a population to relocate to other places in the search space, thus escaping from a local maximum. The SBGA has a mechanism for determining whether an individual from the colony is searching in the same area as the core, which is implemented as a distance to the core: it is the average Hamming distance between a colony member and each member of the core. The distance is used as a fitness function for the members in the colony when the colony becomes too close to the core. The members will then be selected for reproduction, not only according to their objective function value but also according to their distance from the core. They will *evolve* into a new area of the search space.

Since we intend the colony to follow the objective function's landscape, even as it moves away from the core, we are faced with a bi-objective optimization task. We handle this bi-objective optimization using a modification of the VEGA multi-objective optimization system along with dynamically controlled partitioning:

The population for the next generation is split into two sections. The first section is selected under the objective function, and the second under the distance to the core. When filling a subsection, the parents are selected from the previous generation's entire population. However, the fitness function used during the selection is the one associated with the sub-population being filled.

In order to implement the above bi-objective approach to moving the colony away from the core, we need to determine the portion (κ) of the population governed by the distant to the core versus the objective function. This is dynamically calculated (instead of using fixed percentages as withVEGA) using a formula called the *percentage similarity*, see [16] for details.

Migration from Colony to Core

In the SBGA, the colony sends members, called *migrants*, back to the core. During migration the colony may send all of its members to the core or only some portion thereof. The migrants are chosen from the elite members of the colony (25% of the colony has been used).

Since migration of the colony members disrupts the core group, time is given for the colony to evolve potentially useful members. The number of generations between migrations is called the *migration interval*. To reduce the pressure on the core even more, only a few colonies are allowed to send migrants in the same generation.

Just like all multiple-population based GAs, the SBGA needs a method to integrate the migrants arriving from the colony into the core's population. The host population is temporarily enlarged by the migrants and then pared down again to its normal size after reproduction. In other words, selection for reproduction is performed on the extended population, host members plus immigrants, but the new population will only be the size of the original population.

4 Analysis of the Behavior of SBGA

Preliminary Experiments and Questions

In our description of the SBGA we assumed that the core population would behave as a regular GA enhanced by novel solutions found by the colonies. The large population of the core was assumed to be able to do the exploitation of an area with the small, more easily changed colonies providing the exploration. But whether the SBGA does search in this way was never really proved. One of the ways to help us to understand the SBGA's behavior is to monitor each colony's portion governed by the distant to the core, i.e. its κ value.

We should observe the following: at first the core should repel all the colonies away, driving most of their κ value to nearly 0. Some time later if the current best colony is even better than core, immigrants entering the core will force the core to overlap with the colony, thereby forcing the colony's κ higher. Shortly after the repulsive force acted on the colony by the core will cause the colony to move away from the core and the κ value will drop. Maybe several generations later another colony becomes the best and the process will repeat.

Experiments were performed to see if this behavior actually happened. The κ values of all colonies were recorded during various runs of the SBGA using the same fitness function used to test the original SBGA system. Furthermore, a modified SBGA without the migration from the colonies to the core was used as a comparative system to the SBGA. The fitness function is the 6th dimensional F8F2 minimization problem (see [17]). The definition of F8F2 function is

$$\mathrm{F8F2}(\mathbf{x}) = \sum_{i=0}^{n-1} \left(1 + \frac{\left(100\left(x_i^2 - x_{i+1}\right) + \left(1 - x_i\right)^2\right)^2}{4000} - \cos\left(100\left(x_i^2 - x_{i+1}\right) + \left(1 - x_i\right)^2\right) \right) \cdot \qquad \mathbf{(1)}$$

The parameter settings used for the experiment is summarized in **Table 1**. Each experiment lasted 300 generations and was repeated 60 times.

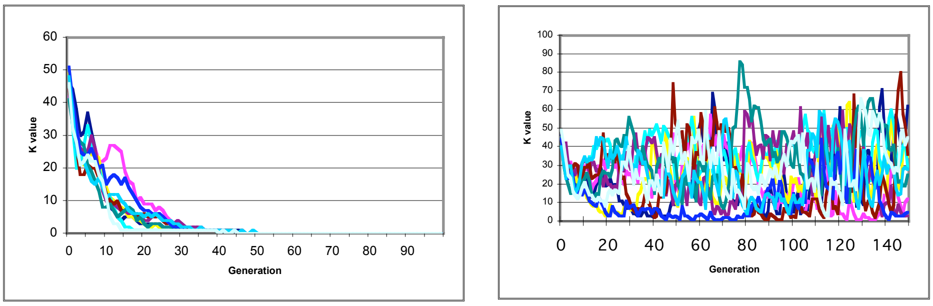
In these 60 experiments we find the trend of the κ value is very similar. **Fig. 1** shows a typical result.

Table 1. SBGA parameter setting

Population Size	**Selection**
Core 1000	Linear rank selection
Colony 100	Low=0.0 (high=2.0) Elitism
Core	**Colonies**
Mutation rate 0.006 bits/locus	Mutation rate 0.01 bits/locus
Prob. of crossover 0.7	Prob. of crossover 0.9
One point crossover for both Core and Colony	Migration size 25% colony size (elite)
	Migration interval to core 6
	Number of Colonies 10

In **Fig. 1a** where no immigration occurs, the colonies will soon be repelled away to other areas of the search space and no longer overlap the core. After 15-40 generations all the k values for colonies are almost 0, which continued throughout the rest of the run (not shown). This means the genetic composition of each colony becomes completely dissimilar to that of the core. Now look at **Fig. 1b**: when the colonies send immigrants to core, the κ values remain high over the entire course of the experiment for most of the colonies. We do not see individual colonies rise and decay sequentially as the core becomes interested in that colony's area. Rather the core seems to overlap with almost all the colonies simultaneously (this occur throughout the run – only the first half of the run is shown). This seems to mean that the core is trying to repel all of the colonies away at the same time, although the repulsive force is not constant for each of them.

Based on the result of the preliminary experiments, it seems the SBGA does not always work as originally expected. Perhaps there is a better explanation search behavior of the SBGA that allows it to better overcome local optima. In the following section we present a new hypothesis of the core's role in search.


a) No migration from colony to core

b) With migration from colony to core

Fig. 1. Overlap between the Core and Colonies (dynamic κ)

New Analysis of the Core's Role for Search

In the above experiment the core overlaps with all of them throughout the experiment. One possible explanation is that, because of the core's large population size and the fact that the immigrants are the elite members; the immigrants from all colonies are entering the core and staying around long enough to create multiple niches in the core. This causes the core to become disorganized, which fragments core's gene pool and confuses the core on where in gene space to go. If this is the case, the core can be seen as a gathering center for the different immigrants from the colonies. If most of colonies continue to find better solutions and contributes them to the core, the core actually will not be stable and cannot focus on searching in just one area of the gene space.

However, the original experiments SBGA really do work better than GA when test on some multimodal environments [15][16]. What makes it better? One reason probably comes from simply using multi-populations. Another reason may be that the main power does not come from the core moving to where some colony is lying in the search space and just repelling it away and then exploring the area by itself; rather the mechanism kicks in at the beginning when the core tries to repel all the colonies away. This kind of repulsive force can make the colonies more active and search more areas, thus increasing the probability that one of them finds a better local optimum or even the global optimum.

From this point of view, the key role of the core is not seen as that of an exploiter who mines the area found by the current best colony, but rather more like the center of a repulsive force or possibly even a memory center who receives immigrants from all the colonies and tell them "you have already looked here, you can search some other place". The core may even be better than a simple memory since it improves on what was stored by evolving it.

As a natural consequence of this new explanation, it becomes feasible to redesign the SBGA and improve its search ability and robustness. In next section we will consider how to redesign the SBGA in light of our new understanding.

5 Redesigning the SBGA

Introduction

Through the analysis from the previous chapter we questioned the explanation of the behavior of the original SBGA and supplied some alternative explanations on why the SBGA is better than simple GA, especially when facing multimodal fitness landscapes. Based on this analysis, we redesign the SBGA along three lines, which are presented below.

Small Core or Big Core

According to the analysis in section 4 the core cannot be easily shifted into the area of the best colony; furthermore if it did manage to cover the new territory, it would not efficiently search it. Therefore, a big core size is probably a waste of resource. As a result in the modified SBGA we decided to shrink the core size and give more search labor to do exploration by increasing the colony number.

However, we must not choose a core size that is too small, for it should be large enough to repel all of the colonies simultaneously. Consequently we decided to reduce the core from 1000 members to 200 members.

Dynamically modifying the Colonies' population sizes

If the core's population size is decreased, then the colonies population must increase. The new analysis of the SBGA's behavior leads us to the conclusion that it is the colonies and not the core that is responsible for exploiting a region in gene space. This suggests that we give the colony with the current highest performance a larger population size, thus giving it the greater search power. Consequently, we introduce competition among the colonies directly: a better colony will be given more members so the population size of the colony is adapted according to their performance.

In section 2 we saw that Schilerkamp-Voose and Mühlenbein added subpopulation competition for population size to their breeder genetic algorithm [12]. We will add that mechanism, with modifications, to the SBGA to determine which colony should have the greater search power.

Each colony will be evaluated over some generation interval (e.g. 5 generations). The colony with the best performance will have their population size increased, and likewise all other colonies will have their population size decreased.

The first step is the choice of which population is to have its size increased. This part of the algorithm is where we differ from the BGA the most. We removed their complex mechanism for "averaging" the sub-population's performance over the generations. We also removed the evening-out of the best of a population through broadcasting each population's elite member to all other populations. In their place, two criteria are used to select which population should have its size increased:

1. *Rank value*
    - Colonies with a higher fitness of its best member get a higher priority of consideration
2. *Colony stagnation*
    - We consider a colony stagnant if there is no improvement in last 10 generations

Expanding on the second criterion, the chosen colony needs to show it is not stagnant before it can be chosen. This means each colony will be monitored to see when the colony last improved. The improvement measure is based on the best fitness of individual in the population, not the average fitness of the population. So the generation gap between the current generation and the generation when the best fitness was found is calculated, and is called the *stagnation interval*. If this gap is bigger than a user defined parameter or threshold (we used the value of 10 to match the BGA *quality criterion*, which our mechanism replaces) the colony will not be chosen and the algorithm will automatically turn to next one on the fitness list. If all the colonies are not eligible, the algorithm will randomly choose one. If there is a tie in the ranks for both criteria, a colony is randomly chosen from among the ties.

After one of the colonies is selected, all other colonies give a fixed percentage of their members (Schilerkamp-Voose and Mühlenbein used the value of 1/8 for their experiments) to the best non-stagnant colony. This is called the *gain criterion.*

Restarting Dead Colonies

Since the colonies is now seen as the focus of the both exploration and exploitation, colonies that are not productive are more deleterious than previously thought and should be replaced. When this happens, these colonies will restart using members from other colonies as seeds.

Three criteria are used for choosing the colony to restart

1. Choose the lowest ranking colony.
2. Choose a colony that has seen no improvements over the last few generations.
   - Preliminary experiments have shown that 10 generations is an effective choice.
3. Choose a colony that has a diversity that is lower than some threshold.
   - Preliminary experiments have shown that 0.15 is an effective choice.

Diversity is computed using the average information entropy at a locus across the population.

After the stagnant colony is chosen, we will restart it. The whole population will be re-initialized based on a seed individual from the colony that is having its population size increased. This seed individual, which will be the best individual from the donating population, will be used to create all of the individuals in the new population by performing bit mutation with a given probability (e,g., 30%). So the new population can be seen as a hyper-sphere around the seed in genotype space, with the radius of the hyper-sphere is decided by the probability of mutation.

Table 2. Parameter setting for two modified system and simple GA

Population Size	**Selection**	
GA 2000, Core 200, Colony 100	Linear rank selection (max = 2), Elitism	
GA and Core	**Colonies**	
Mutation rate = 0.006 bits/locus	Mutation rate = 0.01 bits/locus, Prob. of xover = 0.9	
Prob. of crossover = 0.75	Number of Colonies 18	
Migration	**Special for SBGA with competition**	
Size to core = 25 colony members	Initial Colony size 100	Competition begins 30
Size to colony = 10% colony	Min Colony size 40	Lowest diversity 0.15
Migration interval: to core = 6, to colony = 20	Evaluation interval 5	Stagnation interval 10

Table 3. GA systems with ‘repulsive force’ VS systems without ‘repulsive force’

System	Lower Bound: 29% percentile	Median	Upper Bound: 71% percentile
SBGA	0.05133	0.08921	0.11012
SBGA-NoRepel	0.08926	0.11906	0.13722
SBGA-Com	0.01003	0.01378	0.03319
SBGA-Com-NoRepel	0.05220	0.08905	0.11877
SBGA-Bi	0.02093	0.04066	0.05960
SBGA-Bi-NoRepel	0.05982	0.08984	0.13231

Table 4. Bi-objective function VS Random Immigrant for Colonies

System	Lower Bound: 29% percentile	Median	Upper Bound: 71% percentile
SBGA	0.02228	0.05994	0.08895
Random-Imm	0.06009	0.11147	0.13092

6 Experimental Results and Analysis

Can the Repulsive Central Force Help an Island Model GA?

In this section, we want to test whether the repulsive central force mechanism is useful. Without this mechanism the SBGA can just be considered a rather limited variant of an Island Model GA. We therefore compare the original SBGA and an SBGA with a bidirectional ring added to it (to be more like a traditional Island Model GA) against those same systems with the repulsive central force turned off. To complete our comparisons we will also test the SBGA with the new colony competition system installed, with and without the central repulsive force.

The test function used is F8F2 on 6 dimensions. Each experiment lasted 300 generations and was repeated 60 times. The parameter setting of the original SBGA can be found in Table 1. The configuration for two new modified SBGA systems is given in Table 2.

Table 3 gives the results[1] for three systems and their counterparts[2]. From these results[3], we can see all three systems that use the bi-objective function in their colonies produce better value on average than their counterpart. According the experiment data

---

[1] We record the median instead of the mean because the results are not normally distributed.

[2] We are testing for statistical significance of the comparison between the 3 systems. If the upper and lower bounds don’t cross, neither system is better than the other. This test is not as sensitive as using ANOVA with pair-wise Student’s T tests. These were done (with a confidence level of 95%) and they confirm the results presented here.

[3] The upper and lower percentiles are obtained from the Thompson - Savur method for finding confidence intervals around a median, using a Bonforonni correction (we performed 97 independent comparisons in all our experiments; we therefore always divide our α by 97).

Table 5. Big Core SBGA VS Small Core SBGA

System	Dim	Lower Bound: 29% percentile	Median	Upper Bound: 71% percentile
Core200	4	0.00995	0.01018	0.02060
Core1000	4	0.00990	0.01286	0.02969
Core200	6	0.02228	0.05994	0.08895
Core1000	6	0.05133	0.08925	0.11012
Core200	8	0.13113	0.15580	0.21074
Core1000	8	0.15237	0.20850	0.24143

for this problem we found that in all cases using the repelling central force improved the behavior of the system with statistical significance.

Finally we also see that the subpopulation competition is better than all of the other systems, showing that the new analysis did produce a system that improves the SBGA's behavior overall.

The Repelling Central Force vs. Random Immigrants

Perhaps the SBGA is just effective entirely because of increased diversity. We could try comparing against systems with high mutation rate, but as discussed in section 2, a better technique is to introduce diversity through a Random Immigrant system [4], where in every generation the population is partly replaced by randomly generated individuals. The Random Immigrant mechanism has the ability to constantly keep the diversity of the population high; but unfortunately it is a completely blind process and so can be very disruptive. The repelling force more gently introduces the needed diversity, but if the population is almost converged it loses its power. Comparing these two mechanisms in the same F8F2 (6 dimensions) function produces the following results in table 4.

Here we are using the original SBGA with the same configuration as in Table 1 except that the core size changed to 200 and 18 colonies are used (total population size is the same 2000) as was originally formulated. For the **Random Immigrant** system, every thing is the same but the bi-objective function of the colonies is replaced by the random immigrant mechanism (20% of population is replaced by randomly created individual in every generation). From Table 4 we see that the SBGA with bi-objective function is statistically better the SBGA with Random Immigrant.

Smaller Core with more Colonies vs. Big Core with fewer Colonies

In this experiment, we will see if various size core and colony combinations alone can have improved results on the F8F2 function in 4, 6 and 8 dimensions. The parameter setting for base SBGA is the same as in Table 2 but the core size is larger (**1000** vs. ***200***) and colony number is smaller (**10** vs. ***18***); however, the total population size is always fixed to 2000. Each experiment lasted 300 generations and was repeated 60 times.

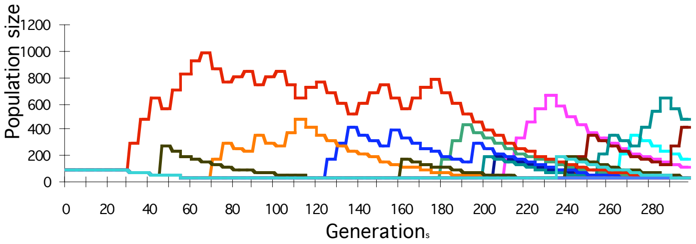


Fig. 2. Dynamical population size for each colony

The results seen in Table 5 can be summarized as follows[4]:

4 dim: Small Core SBGA | Big Core SBGA
6 dim: Small Core SBGA > Big Core SBGA
8 dim: Small Core SBGA > Big Core SBGA

We can see that on the F8F2 problem, a smaller core with more colonies works equally well or better than the SBGA with a large core and fewer colonies function. Furthermore the results get stronger as dimensionality increases. This means that as the fitness landscape becomes larger, more local optimal need to be overcome in order to get better result, and so the advantage accrued by the many colonies being spurred on by a small core becomes more obvious.

Dynamical population size on Graph

We have discussed the mechanism that allowed the SBGA colonies to compete for members and survival. However, that analysis was merely a thought experiment; how the system behaves in reality is still unknown. It will be very educational to see how each colony's population size changes every generation. Consequently, we choose a typical result of the 6 dimensional F8F2 experiment from the previous section and plot all colony population sizes by generation. **Fig. 2** gives the result.

The mechanism used for colony competition can be summarized as follows: at some generation interval (5 generations here) only one colony is chosen as the winner to increase the population size and all others will decrease, except for those that have reached their minimal size. But if the winner wants to keep its position, it must also show improvement.

From **Fig. 2**, we can see that at the early stage of the competition (30 to 100 generation) only 3 of the colonies really get a chance to increase their population size; the rest of them quickly lose their members and reduce to the minimum size. However after generation 120, more colonies get a chance to increase their population size, which can be seen from the fact that more peaks appear on the diagram. One possible reason for this is that, as the colony evolved, many of them gradually lose diversity and also become stagnant. At this time they are qualified for restart. After a colony is restarted using a seed from the current colony winner they have more of a chance to win members.

---

[4] Here 'A>B' means system A is better than system B with statistical significance and 'A | B' is means that system A is statistically indistinguishable from system B.

7 Conclusions

Through the experimentation and analysis of the F8F2 problem, we found that the mechanisms that characterize the SBGA, such as the core's repulsive central force improve multi-populations GAs such as the bi-directional ring Island Model GA. We also re-analyzed the dynamic behavior of the SBGA, verified various aspects of it, and proposed modifications: the addition of dynamic population sizes and population restarts. These additions to the SBGA were also seen to be effective and improved the SBGA's performance.

Reference

1. Eshelman, L.J., Schaffer J.D. (1991). Preventing Premature Convergence in Genetic Algorithms by Preventing Incest. 4th ICGA, pp.115-112.
2. Craighurst, R., Martin, W. (1995) Enhancing GA Performance through Crossover Prohibitions Based on Ancestry. 6th ICGA, pp.130-135
3. T, Bäck, (1992). The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm. PPSN 2.
4. J..J. Grefenstette. (1992) GAs for changing environments. PPSN 2, pp.137-144.
5. Eshelman, L.J., (1991) The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic recombination. FOGA pp. 265-283.
6. E. Cantu-Paz. (1998) A survey of parallel genetic algoritms. Calculateurs paralleles, Reseaux et Systems Repartis, 10(2): 141-171.
7. E. Cantu-Paz (2000). *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer.
8. D. Whitley, S. Rana and R. B. Heckendorn (1998) The Island Model Genetic Algorithm: On Separability, Population Size and Convergence, Proceedings of the AISB Workshop on Evolutionary Computation.
9. Munetomo, M., Takai, Y., & Sato, Y. (1993). An efficient migration scheme for subpopulation-based asynchronously parallel genetic algorithms. 5th ICGA. pg. 649.
10. Harik, G., Cantú-Paz, e., Goldberg, D., & Miller, B. L. (1997). The gambler's ruin problem, genetic algorithms, and the sizing of populations. In the 4th ICEC.. pp. 7-12
11. Michalewicz, Z. (1994) *Genetic Algorithms + Data Structures = Evolution Programs* (2nd edition). Springer-Verlag.
12. Dirk Schlierkamp-Voosen and Heinz Mühlenbein (1994). Strategy adaptation by competing subpopulations. PPSN 3, pages 199-208.
13. Jürgen Branke. (2002). Evolutionary Optimization In Dynamic Environments University of Karlsruhe, Germany. Kluwer Academic Publishers
14. Shisanu Tongchimaand and Prabhas Chongstitvatana. (2002). Parallel Genetic Algorithm with Parameter control.
15. M. Wineberg and F. Oppacher (2000). Enhancing the GA's Ability to Cope with Dynamic Environments. In GECCO-2000. pp. 3-10.
16. M. Wineberg (2000). Improving the behavior of the genetic algorithm in a dynamic environment. PHD Dissertation, Carleton University, Canada.
17. Whitley, D., K. Mathias, et al. (1996). "Evaluating Evolutionary Algorithms". Artificial Intelligence 85: 245-276.