Self-Adaptation in Real-Parameter Genetic Algorithms with Simulated Binary Crossover

Kalyanmoy Deb
Kanpur Genetic Algorithms Laboratory (KanGAL)
Department of Mechanical Engineering
Indian Institute of Technology Kanpur
PIN 208 016, India
deb@iitk.ac.in

Hans-Georg Beyer
Systems Analysis Group
Department of Computer Science/XI
University of Dortmund
44221 Dortmund, Germany
beyer@ls11.informatik.uni-dortmund.de

Abstract

In the context of function optimization, self-adaptation features of evolutionary search algorithms have been explored only with evolution strategy (ES) and evolutionary programming (EP). In this paper, we demonstrate the self-adaptive feature of real-parameter genetic algorithms (GAs) using the simulated binary crossover (SBX) operator. The connection between the working of self-adaptive ESs and real-parameter GAs with SBX operator is also discussed. The self-adaptive behavior of real-parameter GAs is demonstrated on a number of test problems commonly-used in the ES literature. The remarkable similarity in the working of real-parameter GAs and self-adaptive ESs shown in this study suggests the need of emphasizing further studies on self-adaptive GAs.

1 Introduction

Self-adaptation is a phenomenon which makes evolutionary search algorithms flexible and closer to natural evolution. Among the evolutionary methods, self-adaptation properties are explored with evolution strategies (ESs) (Bäck, 1997; Beyer, 1996; Hansen and Ostermeier, 1996; Rechenberg, 1973; Saravanan, Fogel, and Nelson, 1995; Schwefel, 1987) and evolutionary programming (EP) (Fogel, Angeline, and Fogel 1995), although there exist some studies of self-adaptation in genetic algorithms (GAs) with mutation operator (Bäck, 1992). Despite such studies, there exists no formal definition of self-adaptation, nor there exists any discussion on properties which will qualify an algorithm to be a self-adaptive algorithm. However, in the context of function optimization, we would like to have self-adaptation in a search algorithm for the following reasons:

1. Knowledge of lower and upper bounds for the optimal solution may not be known a priori,
2. It may be desired to know the optimal solution with arbitrary precision,
3. The objective function and the optimal solution may change with time.

Binary GAs require the knowledge of bounds on parameter values bracketing the true optimal solution. Real-parameter GAs with flexible crossover and mutation operators does not demand this knowledge and can be used in problems where such information is not available. With fixed-length coding, there exists a fixed amount of precision which an algorithm can be hoped to achieve. Although, the precision can be increased by increasing the string length, it has been shown elsewhere (Goldberg, Deb, and Clark, 1992) that even for simple problems the required population size is of the order of the string length. One other approach to achieve more precision is to use a variable-length coding or a coarse-to-fine grained coding, both of which make an algorithm more complex and subjective to the way decisions are made in switching from coarse to fine grained coding (Shaefer, 1987; Schraudolph and Belew, 1990). Once again, real-parameter GAs with direct use of problem variables can practically achieve any precision in the problem variables, simply because the real numbers are used directly.

One of the challenging and commonly-found problems in real-world search and optimization is a problem which changes with time. In such problems, function landscapes and consequently the optimal solution change with time. When such problems are to be solved for optimality, the search procedure needs to be flexible enough to adapt to the new function landscape as quickly as it changes. In general, the population-based search methods may loose diversity while optimizing the current problem. When the problem changes, there may not be enough diversity left to adapt to the new problem. The invent of self-adaptation with both ES and EP allowed such problems to be solved with an addition of extra strategy parameters which control the degree of search power in their major mutation-based search operators (Bäck, 1997). Although not obvious, such self-adaptive behavior is also possible to achieve with real-parameter GAs with a specialized crossover operator and without using any additional endogenous strategy parameter.

In this paper, we show the self-adaptive behavior of real-parameter GAs with simulated binary crossover (SBX) on a number of commonly-used fitness landscapes. By discussing different variants of self-adaptive ESs, we argue that there is a similarity in working of real-parameter GAs with SBX and one variant of self-adaptive ES.

2 Genetic Algorithms with Simulated Binary Crossover (SBX)

In the year 1995, the first author and his students developed the simulated binary crossover (SBX), which creates children solutions in proportion to the difference in parent solutions (Deb and Agrawal, 1995; Deb and Goyal, 1998). The procedure of computing the children solutions $x_i^{(1,t+1)}$ and $x_i^{(2,t+1)}$ from parent solutions $x_i^{(1,t)}$ and $x_i^{(2,t)}$ is described as follows. First, a random number u between 0 and 1 is created. Thereafter, from a specified probability distribution function defined over a non-dimensionalized parameter $\beta = |(x_i^{(2,t+1)} - x_i^{(1,t+1)})/(x_i^{(2,t)} - x_i^{(1,t)})|$:

$$P(\beta) = \begin{cases} 0.5(\eta+1)\beta^{\eta}, & \text{if } \beta \le 1; \\ 0.5(\eta+1)/\beta^{\eta+2}, & \text{otherwise,} \end{cases} \quad (1)$$

the ordinate β_q is found so that the area under the probability curve from 0 to β_q is equal to the chosen random number u:

$$\beta_q = \begin{cases} (2u)^{\frac{1}{\eta+1}}, & \text{if } u \le 0.5; \\ (1/(2(1-u)))^{\frac{1}{\eta+1}}, & \text{otherwise.} \end{cases} \quad (2)$$

In the above expressions, the distribution index η is any nonnegative real number. A large value of η allows a large probability for creating near parent solutions and a small value of η allows distant points to be created as children solutions. After obtaining β_q from the above probability distribution, the children solutions are calculated as follows:

$$x_i^{(1,t+1)} = 0.5\left[(1+\beta_q)x_i^{(1,t)} + (1-\beta_q)x_i^{(2,t)}\right], \quad (3)$$

$$x_i^{(2,t+1)} = 0.5\left[(1-\beta_q)x_i^{(1,t)} + (1+\beta_q)x_i^{(2,t)}\right]. \quad (4)$$

For a fixed β_q, the difference in children solutions is proportional to that of parent solutions:

$$x_i^{(2,t+1)} - x_i^{(1,t+1)} = \beta_q \left(x_i^{(2,t)} - x_i^{(1,t)}\right). \quad (5)$$

This has an important implication. Let us consider two scenarios: (i) Two parents are far away from each other, and (ii) two parents are closer to each other. For illustration, both these cases (with parent solutions $x_i^{(1,t)} = 2.0$ and $x_i^{(2,t)} = 5.0$ in the first case and with parent solutions $x_i^{(1,t)} = 2.0$ and $x_i^{(2,t)} = 2.5$ in the second case)

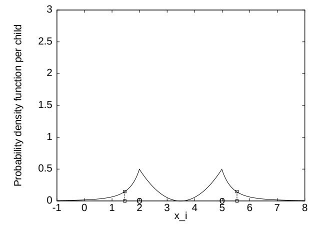

Figure 1: Probability distribution of children solutions with distant parents.

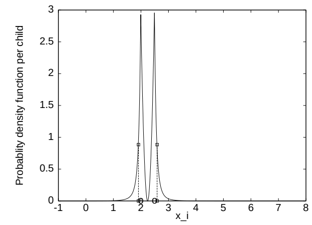

Figure 2: Probability distribution of children solutions with closely spaced parents.

and the corresponding probability distributions with $\eta = 2$ are shown in Figures 1 and 2, respectively. The figures also show the corresponding children solutions (marked with a box) for $u = 0.8$ or $\beta_q = 2.5^{1/3}$. The figures clearly show that if the parent values are far from each other (the first case), solutions away from parents are possible to be created. But if the parent values are close by (the second case), distant children solutions are not likely. Essentially, there are two properties which give the SBX operator its search power:

1. The extent of children solutions is in proportion to the parent solutions, and
2. Near parent solutions are more likely to be chosen as children solutions than solutions distant from parents.

Both the above properties may be regarded as essential for a crossover operator to exhibit self-adaptive behavior in GAs. This is because with these properties the diversity in children solutions is directly controlled by the diversity in parent solutions. Since the movement of the parent population in the search space is dictated by the fitness function through the selection operator and such a crossover operator allows GAs to search a region near (in a proportional sense) the parent population, the self-adaptation is likely. We shall discuss the connection of these properties of a crossover operator with self-adaptive ES in Section 4.

There exists a number of other real-parameter GA implementations, where crossover and mutation operators are applied directly on real parameter values. Among them, Eshelman and Schaffer's (1993) blend crossover (BLX) is of importance here. For two parent points $x_i^{(1,t)}$ and $x_i^{(2,t)}$ (assuming $x_i^{(1,t)} < x_i^{(2,t)}$), the BLX-$\alpha$ randomly picks a point in the range $[x_i^{(1,t)} - \alpha(x_i^{(2,t)} - x_i^{(1,t)}), x_i^{(2,t)} + \alpha(x_i^{(2,t)} - x_i^{(1,t)})]$. Thus, if u is a random number between 0 and 1, the following is a child solution:

$$x_i^{(1,t+1)} = (1-\gamma)x_i^{(1,t)} + \gamma x_i^{(2,t)}, \quad (6)$$

where $\gamma = (1+2\alpha)u - \alpha$. Thus, this factor γ is uniformly distributed in $[-\alpha, 1+\alpha]$. Rearranging equation 6, we observe that BLX-α also makes the difference in a child and one of its parent solutions in proportion to the difference between both parent solutions. However, the second property of SBX operator mentioned above is not truly present in the BLX-α operator. Although near parent solutions are always chosen, BLX-α does not assign a monotonically decreasing probability for choosing a child solution away from parents. As we shall see in Section 5, BLX-0.5 does not provide enough flexibility for the GAs to show sustained self-adaptation.

Ono and Kobayashi (1997) suggested a unimodal normally distributed crossover (UNDX) operator, where three parent solutions are used to create two children solutions. Children solutions are created from an ellipsoidal probability distribution with one axis is formed along the line joining two of the three parent solutions and the extent of the orthogonal direction is decided by the perpendicular distance of the third parent from the axis. This operator also creates a child solution from at a distance from parent proportional to the difference in parent solutions, thereby making the operator a potential candidate to give GAs its self-adaptive power.

3 Self-Adaptive Evolution Strategies

A simple adaptive evolution strategy was suggested for the (1+1)-ES by Rechenberg (1973). Depending on the success or failure of mutations in past few mutations, the mutation strength is decreased or increased by the simple 1/5-th rule. Among the current studies, there are three different ways self-adaptation is used in ES—(i) a hierarchically organized population-based meta-ES (Herdy, 1992), (ii) adaptation of covariance matrix (CMA) determining the probability distribution for mutation (Hansen and Ostermeier, 1997), and (iii) explicit use of self-adaptive control parameters (Rechenberg, 1973; Schwefel, 1987). The meta-ES method of self-adaptation uses two levels of ESs—the top level optimizes the strategy parameters (such as mutation strengths), a solution of which is used to optimize the true objective function in the lower level ES. Although the idea is simple, it involves a number of additional strategy parameters and the needed number of overall function evaluations may not make it suitable for real-world applications. The CMA method records the population history for some number of iterations before executing expensive numerical computation of finding covariance and variance information among object variables. Although application to a number of test problems shows promising results, the algorithm is difficult to implement and clearly the algorithm lacks any motivation to believe whether such complicated computations resembles any event of natural evolution. Nevertheless, the CMA approach seems interesting and readers may refer to the literature for more details (Hansen and Ostermeier, 1996; 1997).

We now discuss the third type of self-adaptive ES where the strategy parameters are explicitly coded and updated in each generation. Although there exists other ways to update (Rechenberg, 1994), we discuss here the lognormal update rules. A recent study by the second author reveals that there is a relationship between the lognormal update rule with other learning rules (Beyer, 1996). There are basically three different implementations which are in use.

3.1 Isotropic self-adaptation

In this self-adaptive ES, a single mutation strength σ is used for all variables. In addition to N object variables, the strategy parameter σ is also used in a population member. Here are the update rules:

$$\sigma^{(t+1)} = \sigma^{(t)} \exp(\tau_0 \mathrm{N}(0,1)), \tag{7}$$

$$x_i^{(t+1)} = x_i^{(t)} + \sigma^{(t+1)} \mathrm{N}_i(0,1), \tag{8}$$

where $\mathrm{N}(0,1)$ and $\mathrm{N}_i(0,1)$ are realizations of an one-dimensional normally distributed random variable with mean zero and standard deviation one. The parameter τ_0 is the learning parameter which is $\tau_0 \propto N^{-1/2}$, where N is the dimension of the variable vector (Schwefel, 1987). Beyer (1996) has shown that the optimal learning parameter for (1,λ)-ES is $\tau_0 = c_{1,\lambda}/\sqrt{N}$, where $c_{1,\lambda}$ is the progress coefficient. For multi-parent ESs, we use $c_{\mu,\lambda}$ or $c_{\mu/\mu,\lambda}$ as constant of proportionality in the corresponding ES, although they may not be optimal. The above update rule for σ requires an initial value. In all simulations here, we choose a $\sigma^{(0)} = (x^u - x^l)/\sqrt{12}$ which assumes a uniform distribution of solutions within the specified range of x_i values (assuming $x^u = x_i^u$ and $x^l = x_i^l$ for all i).

3.2 Non-isotropic self-adaptation

Here, a different mutation strength σ_i is used for each variable. This ES is capable of learning to self-adapt to problems where variables are unequally scaled in the objective function. In addition to N object variables, there are N other strategy parameters. The update rules are as follows:

$$\sigma_i^{(t+1)} = \sigma_i^{(t)} \exp\left(\tau' \mathrm{N}(0,1) + \tau \mathrm{N}_i(0,1)\right), \tag{9}$$

$$x_i^{(t+1)} = x_i^{(t)} + \sigma_i^{(t+1)} \mathrm{N}_i(0,1), \tag{10}$$

where $\tau' \propto (2n)^{-1/2}$ and $\tau \propto (2n^{1/2})^{-1/2}$. Due to lack of any theoretical results on this self-adaptive ES, we use the progress coefficient of the (μ,λ)-ES or $(\mu/\mu,\lambda)$-ESs as the constant of proportionality of both τ' and τ. Similar initial values for $\sigma_i^{(0)}$ as discussed for isotropic self-adaptive ESs are used here.

3.3 Correlated self-adaptation

Here, different mutation strengths σ_i and rotation angles α_i are used to represent the covariances for pair-wise interactions among variables. Thus, in addition to N object variables there are a total of N mutation strengths and

$N(N-1)/2$ rotation angles used explicitly in each population member. The update rules are as follows:

$$\sigma_i^{(t+1)} = \sigma_i^{(t)} \exp\left(\tau' \mathrm{N}(0,1) + \tau \mathrm{N}_i(0,1)\right), \quad (11)$$

$$\alpha_i^{(t+1)} = \alpha_i^{(t)} + \beta \mathrm{N}_i(0,1), \quad (12)$$

$$\vec{x}^{(t+1)} = \vec{x}^{(t)} + \vec{\mathrm{N}}\left(\vec{0}, \mathrm{C}(\vec{\sigma}^{(t+1)}, \vec{\alpha}^{(t+1)})\right), \quad (13)$$

where $\vec{\mathrm{N}}\left(\vec{0}, \mathrm{C}(\vec{\sigma}^{(t+1)}, \vec{\alpha}^{(t+1)})\right)$ is a realization of correlated mutation vector with a zero mean vector and covariance matrix C. The parameter β is fixed as 0.0873 (or 5^o) (Schwefel, 1977). The parameters τ' and τ are used the same as before. We initialize the rotation angles within zero and 360 degrees at random.

4 Self-Adaptive Power of GAs with SBX

Under isotropic self-adaptive ES, the difference between the child and its parent (say, Δ) can be written from equations 7 and 8, as follows:

$$\Delta = \left(\sigma^{(t)} \exp(\tau_0 \mathrm{N}(0,1))\right) \mathrm{N}(0,1). \quad (14)$$

Thus, an instantiation of Δ is a normal distribution with zero mean and a variance which depends on $\sigma^{(t)}$, τ_0, and the realization of the lognormal distribution. For our discussion, it is important to note that the variance of Δ is proportional to the mutation strength $\sigma^{(t)}$, which signifies, in some sense, the population diversity.

Under the SBX operator, we write the term Δ using equations 3 and 4 as follows:

$$\Delta = \frac{\delta_p}{2}(\beta_q - 1), \quad (15)$$

where δ_p is the absolute difference in two parent solutions. The above equation suggests that an instantiation of Δ under SBX operator depends on the distribution of $(\beta_q - 1)$ for a particular pair of parents. The distribution of β_q has its mode at $\beta_q = 1$ (equation 1), thus, the distribution of $(\beta_q - 1)$ will have its mode at zero. Although, we have not used a normal distribution for $(\beta_q - 1)$ here, Figure 1 or 2 suggest that a child with a small Δ has a higher probability to be created than a child with a large Δ. However, the variance of the distribution of Δ depends on δ_p, which signifies the population diversity. Thus, there is a remarkable similarity in the way children solutions are assigned in both isotropic self-adaptive ES and in GAs with SBX. In both cases, the children solutions closer to parent solutions are assigned more probability to be created than solutions away from parents and the variance of this probability distribution depends on the current population diversity.

The self-adaptation power of self-adaptive ESs comes from the fact that the mutation strength gets continuously updated depending on the fitness landscape. For example, if the fitness landscape is such that the population needs to concentrate in a narrow region in the search space for improvement in the fitness, a self-adaptive ES usually evolves mutation strengths to become smaller and smaller, so that search concentrates near the parents rather than away from parents. This is precisely how a self-adaptive ES works on sphere model to achieve continuously improving performance. The outcome of continuously reducing mutation strength is that most population members come closer and closer. When population members come closer in a real-parameter GA, the effective variance of probability distribution under SBX operator also reduces. This, in turn, creates children solutions which are also not far away from each other. This helps to produce continuously closer population members, thereby producing the effect of increased precision like that in the self-adaptive ES. A similar phenomenon is expected to occur when a fitness function demands the population to diverge to get to the optimal region or demands other kind of variations in the search process.

5 Simulation Results

In this section, we present simulation results of real-parameter GAs with SBX operator on a number of different test problems borrowed from the ES literature. In all methods, no special effort is spent to find the best parameter settings, instead a reasonable set of parameter values are used.

5.1 Sphere Model

We consider several variants of this function in the following subsections.

5.1.1 Function F1-1: Quadratic function

First, we consider the sphere model, where the objective is to minimize the following N-variable function:

$$\text{F1-1:} \quad \text{Minimize} \sum_{i=1}^{N} (x_i - x_i^*)^2, \quad (16)$$

where x_i^* is the optimal value of the i-th variable. We use $N = 30$ and $x_i^* = 0.0$ and initialize populations in $x_i \in [-1.0, 1.0]$. For real-parameter GAs with SBX, we use binary tournament selection and SBX with $\eta = 1$. No mutation operator is used. A population size of 100 is used. The Euclidean distance $R = \sqrt{\sum_{i=1}^{N} x_i^2}$ of the best solution in a population from the minimum is plotted with generation number in Figure 3. The ordinate axis is drawn in logarithmic scale, thus the figure shows that real-parameter GAs with SBX (solid line) are able to maintain exponentially increased precision with generation number. A comma ES with no self-adaptation and with $\mu = 1$ and $\lambda = 100$ are shown next. We have used a fixed mutation

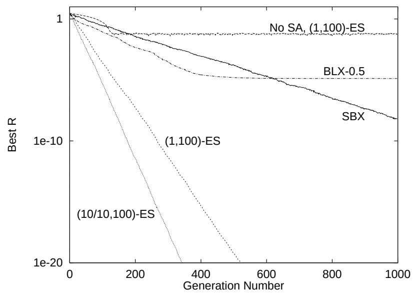

Figure 3: Population-best distance R on test function F1-1.

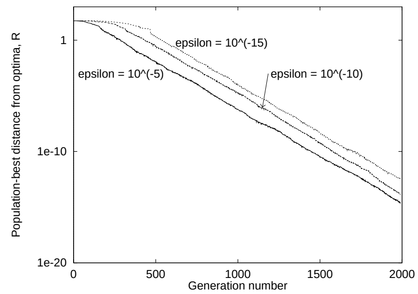

Figure 4: Population-best distance from optimum (R) for F1-2.

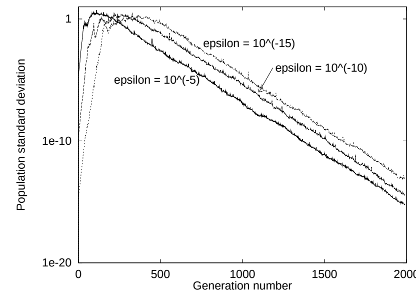

Figure 5: Population standard deviation with generation number for F1-2.

strength of 0.01 here. The children population size $\lambda = 100$ is used to keep the number of function evaluations same as that in the real-parameter GAs. The figure re-confirms an already established fact that an ES without self-adaptation cannot find continuously-increasing precision.

Next, we apply isotropic self-adaptive ESs ((1,100)-ES and (10/10,100)-ES). In the latter ES, dominant crossover on problem variables and intermediate crossover on the mutation strength are used. The figure also shows an already established fact (Beyer, 1996) that with proper learning parameter update, self-adaptive ESs can find continuously improving precision. There are two aspects to notice in these plots. First, the introduction of crossover enhances the performance of self-adaptive ES in this problem (Beyer, 1995). Second, the performance of the self-adaptive ES is much better than that of real-parameter GAs with the SBX operator. This test function is unimodal and isotropic ES uses this problem knowledge by using only one mutation strength parameter for all variables. On the other hand, real-parameter GAs with SBX does not use any such information and hence the progress rate comparison between the two algorithms is not proper. Moreover, the effective selective pressure in both cases are not equivalent. However, what is important here to note that real-parameter GAs with SBX operator is able to maintain exponentially increasing precision with generation number.

Before we leave this test problem, we would like to mention that when real-parameter GAs with BLX-0.5 is used on this problem, similar self-adaptive behavior is not observed (Figure 3). GAs get stuck at solutions away (at a distance $R = 1.276(10^{-5})$) from the optimum. Although it has some capabilities of self-adaptation compared to (1,100)-ES without self-adaptation, clearly the self-adaptive power is not adequate.

5.1.2 Function F1-2: Biased population

To avoid any bias of a symmetrically chosen initial population (Fogel and Beyer, 1996), we use the same sphere model as that used in the previous subsection, but here we initialize the population far away from the optimum and in a narrow range $[10 - \epsilon, 10 + \epsilon]$, where ϵ is a small positive number. However, the minimum solution is still kept at $x_i^* = 0.0$. We choose three different values of $\epsilon = 10^{-5}$, 10^{-10}, and 10^{-15}. Figures 4 and 5 show R and the population standard deviation (averaged over all 30 variables). Identical GA parameter settings as before are used here. Notice, how GAs require more generations to bring the population standard deviation to a reasonable limit with smaller ϵ values. Once the population has the correct population variance and it is near the optimum, the rate of convergence to the optimum with increasing precision is independent of how the population was initialized. These results show that although 100 members in the initial population was confined to a tiny region, GAs with SBX operator can come out of there mainly with function value information alone and converge to the correct optimum. Similar performance is observed with self-adaptive ESs (Bäck, 1997).

5.1.3 Function F1-3: Time-varying function

In order to investigate the performance of real-parameter GAs with SBX on time-varying functions, we have used the same function as F1-1, but x_i^* now varies with generation number in the range $[-1.0, 1.0]$ at random. The optimum is changed after every 1,000 generations so that at the time of change, the population diversity has reduced substantially. The best function value and average population standard deviation in all variables are plotted versus generation number in Figure 6. GA parameter settings are the same as that in F1-1. The figure shows that even though all population members are all within a small range (in the order of 10^{-10}) at the end of 999 generations, the population in GAs with the SBX operator can diverge and gets adapted to a changed optimum. This happens not only once, but as many times as there is a change in the function.

5.1.4 Function F1-4: Multi-Modal function

We now choose one test problem which is not quadratic in the search space. Moreover, in the range $[-1.0, 1.0]$, where the population is initialized the function is unimodal, but

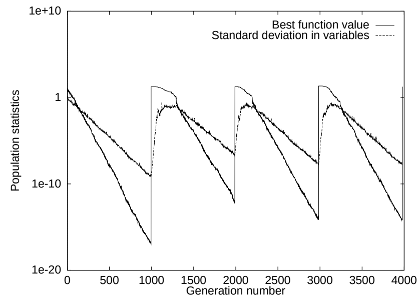

Figure 6: Population-best function value and average of population standard deviation for F1-3.

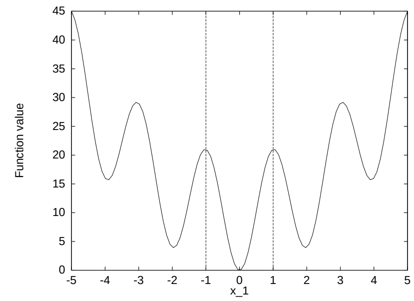

Figure 7: An one-dimensional version of F1-4.

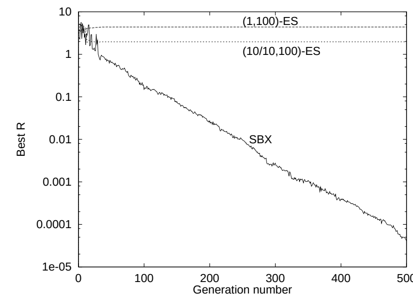

Figure 8: Population-best distance R from optimum for F1-4.

just outside this range the function has other local attractors. We simply add a non-linear term to the sphere model:

$$\text{F1-4:} \quad \text{Minimize} \sum_{i=1}^{N} \left(x_i^2 + 10(1 - \cos(\pi x_i)) \right). \qquad (17)$$

Figure 7 shows one-dimensional version of this function. The range of initial population is shown by drawing two vertical dashed lines. Figure 8 shows the performance of real-parameter GAs with SBX having $\eta = 1$. The figure also shows the performance of isotropic self-adaptive (1,100)-ES and (10/10,100)-ES with identical parameter settings as that used in Function F1-1. Now neither self-adaptive ES is able to converge to the global attractor (all $x_i = 0$ having function value equal to zero), although the initial population was placed in the global basin. In self-adaptive ESs, the mutation strength for each parameter needs an adaptation time within which update of mutation strengths and corresponding fitness landscape should make a suitable agreement. If either due to improper use of learning rate or other ES parameters or due to a complex fitness landscape this agreement does not happen, the mutation strength does not get adapted properly. Since in this function, the function landscape just outside $[-1, 1]$ has a non-agreeing landscape compared to that inside the region $[-1, 1]$ for each variable, self-adaptive ES gets confused whether to increase or decrease mutation strengths. However, as suggested in Beyer (1996, page 335), if lower and upper bounds on variables are known with confidence, self-adaptative ES may be used with a small initial mutation strength. This is because in such cases the mutated solutions are not likely to be outside $[-1, 1]$, and thus self-adaptive ES will be confined in the global basin. When a mutation strength one-tenth of what used in the above runs is used, the ES converges to the correct optimum. However, such a small initial mutation strength may lead to a larger adaptation time in other functions where divergence from the initial population is necessary to get to the true optimum (such as test function F1-2, F1-3, or ridge functions).

5.2 Elliptic Model

In this function, every variable has an unequal contribution to the objective function. We consider a couple of variants of the elliptic function.

5.2.1 Function F2-1: Elliptic function

We have a function where parameters have exponentially increasing contribution:

$$\text{F2-1:} \quad \text{Minimize} \sum_{i=1}^{N} 1.5^{i-1} x_i^2. \qquad (18)$$

Figure 9 shows the objective function value of the best solution in the population. The population is initialized in $x_i \in [-1.0, 1.0]$. Once again, we use the same GA parameters as before, but use tournament size 3 to compare the performance with self-adaptive ES. The performance of non-isotropic (10/10, 100)-ES reveals that both algorithms have similar performance on this function. However, the performance of BLX-0.5 on this function shows that BLX-0.5 does not have adequate self-adaptive power. Figure 10 plots the population standard deviation in x_1, x_{15} and x_{30} in the population for the real-parameter GAs with the SBX operator. Since they are scaled as 1.0, $1.5^{14} \approx 292$, and $1.5^{29} \approx 127,834$, the 30-th variable is likely to have smaller variance than the 1st variable. The figure shows this fact clearly. Since, ideal mutation strengths in ESs for these variables are also likely to be inversely proportional to 1.5^{i-1}, we find similar ordering with non-isotropic self-adaptive ES as well (Figure 11). Thus, there is a remarkable similarity by which both real-parameter GAs with SBX and self-adaptive ES work.

5.2.2 Function F2-2: Time varying elliptic function

Like in the sphere model, we construct a test problem where the elliptic function changes its optimum solution occasionally with generation. We use the following func-

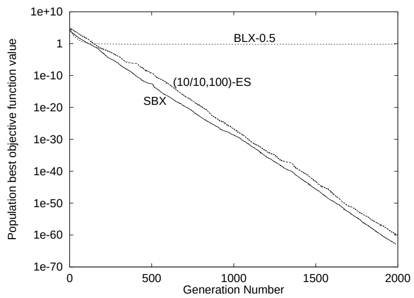

Figure 9: The best objective function value for F2-1.

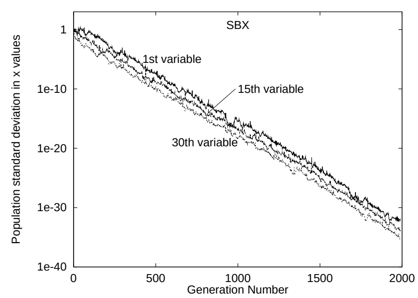

Figure 10: Population standard deviation variables x_1, x_{15}, and x_{30} obtained with GAs with SBX for F2-1.

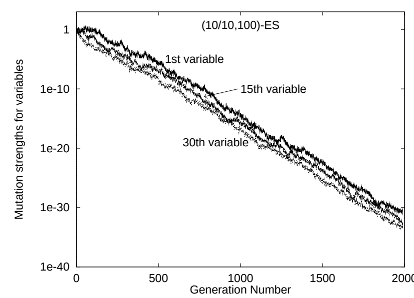

Figure 11: Mutation strength for variables x_1, x_{15}, and x_{30} obtained with (10/10,100)-ES for F2-1.

tion:

$$\text{F2-2:} \quad \text{Minimize} \sum_{i=1}^{N} r_i (x_i - x_i^*)^2. \qquad (19)$$

where r_i is an randomly shuffled array of integers between 1 to N. After every 1,000 generations, this array is changed to another permutation of integers from 1 to N. In addition, the optimum (x_i^* values) of function is also changed to a random value in the range $[-1.0, 1.0]$. The parameter setting of tournament size of 3 for real-parameter GAs and $\mu = \rho = 10$ for non-isotropic self-adaptive ES make corresponding algorithms too sluggish to adapt to the changes made at every 1,000 generations. Thus, in this experiments, we use tournament size of 2 and $\mu = \rho = 15$. Figure 12 shows the population-best objective function value and the population standard deviation in the x_{15} variable. It is clear that although the population deviations are quite small at the end of 999 generations, the population can adapt to the change in the function landscape. A similar performance plots are observed with (15/15, 100)-ES in Figure 13. In this figure, the population-best objective function value and the mutation strength for x_{15} variable are shown. The remarkable similarity in both figures suggests that for chosen parameter settings both real-parameter GAs with SBX operator and self-adaptive ES have very similar working principles.

5.3 Correlated function

Next, we consider a function where pair-wise interactions of variables exist. The Schwefel's function is chosen:

$$\text{F3-1:} \quad \text{Minimize} \sum_{i=1}^{N} \left(\sum_{j=1}^{i} x_j \right)^2. \qquad (20)$$

The population is initialized at $x_i \in [-1.0, 1.0]$. Figure 14 shows the performance of real-parameter GA with SBX ($\eta = 1$) and tournament size 3, non-isotropic (10,100)-ES, and correlated self-adaptive (10,100)-ES. Although all methods have been able to find increased precision in obtained solutions, the rate of progress for the real-parameter GAs with SBX and uncorrelated ES are much slower compared to that of the correlated self-adaptive ESs, where correlations among variables are explicitly taken care of. In the SBX operator, variable-by-variable crossover is used with a probability of 0.5. Correlations (or linkage) among the variables are not explicitly considered in this version of SBX. Although some such information comes via the population diversity in variables, it is not enough to progress faster towards the optimum in this problem. Clearly, a better mechanism to handle the linkage issue but with the concept of probability distribution to create children solutions is in order to solve such problems better. We are working on a SBX operator where a line version of SBX operator is added on pairs of variables, in addition to the regular variable-by-variable SBX operator. This should provide the needed correlations among variables in problems like the above.

6 Conclusions

In this paper, we have demonstrated that real-parameter genetic algorithms (GAs) with simulated binary crossover (SBX) exhibit self-adaptive behavior on a number of test problems. The SBX operator has two properties: (i) children solutions are created in proportion to the difference in parent solutions and (ii) children solutions closer to parent solutions are monotonically more probable. It has been observed that GAs with the SBX operator use diversity among population members in a way which is similar in concept to the explicit control of strategy parameters in self-adaptive ESs.

This study is interesting and should encourage further research in the areas of self-adaptation in genetic algorithms. In this regard, GAs with other real-parameter crossover operators (BLX-α, UNDX or others) may be tested for adequate self-adaptation. An SBX operator with a different probability distribution than that used here (may be a log-

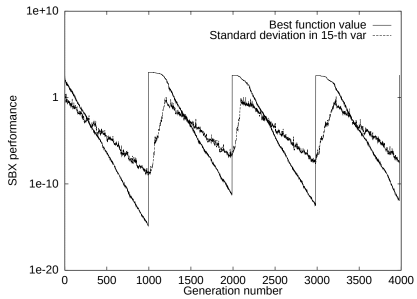

Figure 12: Population-best objective function value and the standard deviation of x_{15} for F2-2.

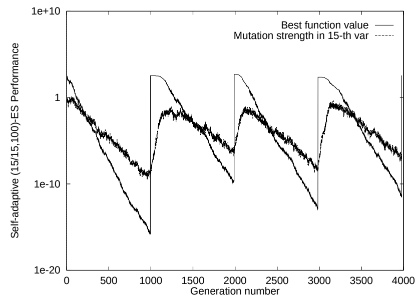

Figure 13: Population-best objective function value and the mutation strength for x_{15} on F2-2.

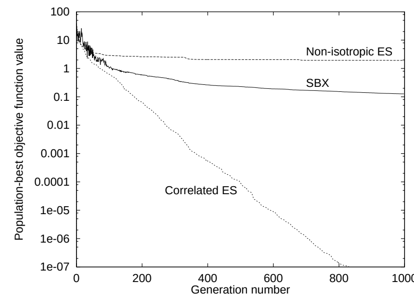

Figure 14: The best population objective function value for F3-1.

normal distribution) can be investigated.

Acknowledgments

The first author acknowledges the support provided by the Alexander von Humboldt Foundation, Germany during the course of this study. The second author acknowledges support as Heisenberg Fellow of the Deutsche Forschungsgemeinschaft under grant Be1578/4-1.

References

Bäck, T. (1997). Self-adaptation. *Handbook of Evolutionary Computation*. New York: Oxford University Press.

Bäck, T. (1992). The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm. *Parallel Problem Solving from Nature, II*, 85–94.

Beyer, H.-G. (1996). Toward a theory of evolution strategies: Self-adaptation. *Evolutionary Computation, 3*(3), 311–347.

Beyer, H.-G. (1995). Toward a theory of evolution strategies: On the benefit of sex—the $(\mu/\mu, \lambda)$-theory. *Evolutionary Computation, 3*(1), 81–111.

Deb, K. and Agrawal, R. B. (1995) Simulated binary crossover for continuous search space. *Complex Systems, 9* 115–148.

Deb, K. and Goyal, M. (1998). A robust optimization procedure for mechanical component design based on genetic adaptive search. *Transactions of the ASME: Journal of Mechanical Design*, *120*(2), 162–164.

Eshelman, L. J. and Schaffer, J. D. (1993). Real-coded genetic algorithms and interval schemata. *Foundations of Genetic Algorithms, II* (pp. 187–202).

Fogel, L. J., Angeline, P. J., and Fogel, D. B. (1995). An evolutionary programming approach to self-adaptation on finite state machines. *Proceedings of the Fourth International Conference on Evolutionary Programming*, 355–365.

Fogel, D. B. and Beyer, H.-G. (1996). A note on the empirical evaluation of intermediate recombination. *Evolutionary Computation, 3*(4). 491–495.

Goldberg, D. E., Deb, K., and Clark, J. H. (1992). Genetic algorithms, noise, and the sizing of populations. *Complex Systems, 6*, 333–362.

Hansen, N. and Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. *Proceedings of the IEEE International Conference on Evolutionary Computation*, 312–317.

Hansen, N. and Ostermeier, A. (1997). Convergence properties of evolution strategies with the derandomized covariance matrix adaptation: The $(\mu/\mu_I, \lambda)$-CMA-ES. *European Congress on Intelligent Techniques and Soft Computing*, 650–654.

Herdy, M. (1992). Reproductive isolation as strategy parameter in hierarhically organized evolution strategies. *Parallel Problem Solving from Nature, II*, 207–217.

Ono, I. and Kobayashi, S. (1997). A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover. *Proceedings of the Seventh International Conference on Genetic Algorithms*, 246–253.

Rechenberg, I. (1994). *Evolutionsstrategie'94*. Stuttgart: Frommann-Holzboog Verlag.

Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Frommann-Holzboog Verlag.

Saravanan, N., Fogel, D. B., and Nelson, K. M. (1995). A comparison of methods for self-adaptation in evolutionary algorithms. *BioSystems*, *36*, 157–166.

Schraudolph, N. N. and Belew, R. K. (1990). Dynamic parameter encoding for genetic algorithms. Technical Report No. LAUR90-2795. Los Alamos: Los Alamos National Laboratory.

Schwefel, H.-P. (1987). Collective phenomena in evolutionary systems. *Problems of Constancy and Change—the Complementarity of Systems Approaches to Complexity*. 1025–1033.

Schwefel, H.-P. (1977). *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie (Interdisciplinary Systems Research 26)* Basel: Birkhäuser.

Shaefer, C. G. (1987). The ARGOT strategy: Adaptive representation genetic optimizer technique. *Proceedings of the Second International Conference on Genetic Algorithms*, 50–58.