On Self-Adaptive Features in Real-Parameter Evolutionary Algorithms

Hans-Georg Beyer
Systems Analysis Group
Department of Computer Science XI
University of Dortmund
D-44221 Dortmund, Germany
beyer@zappa.cs.uni-dortmund.de

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

Abstract

Due to the flexibility in adapting to different fitness landscapes, self-adaptive evolutionary algorithms (SA-EAs) have been gaining popularity in the recent past. In this paper, we postulate the properties that SA-EA operators should have for successful applications in real-valued search spaces. Specifically, population mean and variance of a number of SA-EA operators, such as various real-parameter crossover operators and self-adaptive evolution strategies, are calculated for this purpose. Simulation results are shown to verify the theoretical calculations. The postulations and population variance calculations explain why self-adaptive GAs and ESs have shown similar performance in the past and also suggest appropriate strategy parameter values which must be chosen while applying and comparing different SA-EAs.

Keywords

Self-adaptation, Test fitness landscapes, Population mean, Population variance, Evolution strategies, Genetic algorithms, Simulated binary crossover, Blend crossover operator, Fuzzy recombination operator

I. Introduction

The self-adaptive feature of evolutionary algorithms[1] (EAs), aiming at an implicit control of certain endogenous strategy parameters, is commonly regarded as a specialty of evolution strategies (ESs) [1], [2], [3] and evolutionary programming (EP) [4], [5]. Until now, the main focus of research has been in the implicit learning of mutation strengths or generally mutation distributions of EAs operating in real-valued search spaces. However, there are also attempts to use this idea to self-adapt mutation rates in GAs [6], [7] and mutation distributions in GAs [8], to learn recombination parameters in GAs [9], to self-adapt permutation operators for ordering problems [10], [11], and to self-adapt mutation rates for mixed-integer problems [12] in ESs, and others [13]. While these investigations have in general shown the usefulness of the self-adaptation (SA) concept in evolution strategies (ESs) [14], [15], [16], [17] and evolutionary programming (EP) [13], [18], only little is known about the conditions under which an algorithm may exhibit self-adaptation properties. Even worse, it is by no means clear what behavior may be regarded as a desirable "self-adaptive behavior," and furthermore, whether the standard approach to SA (i.e., the adaptation via endogenous strategy parameters [19]) is the best way to acquire adaptive behavior in EAs. To overcome these deficiencies, this paper attempts to contribute to the solution of the problems just raised for the case of real-valued function optimization:
(a) by providing postulates on desirable behavior of SA and
(b) by revealing connections between a class of real-coded GAs and the SA-ESs showing a surprising similar behavior.

Self-adaptive evolutionary algorithms (SA-EAs) use operators which are adaptive in some sense – either the strategy parameters controlling the extent of search are evolved [2], [14], [19] or are updated based on statistics gathered from past generations [20], [21]. Since these operators are tunable (i.e., they can be controlled by so-called endogenous strategy parameters), they have been shown to adapt to a variety of fitness landscapes better than EAs that do not use specific self-adaptive operators. For example, when a population is located in the basin of attraction of an optimum, SA-EAs have been shown empirically to converge to the optimum exponentially quickly [14], [17], [19]. For the special case of the $(1,\lambda)$-ES on the sphere model, the first author proved this convergence behavior theoretically [15]. Self-adaptive EAs have also been shown to adapt to changing fitness landscapes. They have been found to quickly get out of the current optimum and proceed towards the new optimum when the fitness landscape changes (see e.g. [19], [22]). If initialized away from the optimum, SA-EAs have also been able to diversify the population exponentially fast to reach near the optimum [22].

Recently, the self-adaptive features of a number of real-coded GAs have also been demonstrated on similar fitness landscapes [23], [24], [25]. Unlike evolving or calculating an endogenous strategy parameter set, these GAs use special crossover operators which create offspring statistically located in proportion to the difference of the parents in the search space. Since the offspring population (i.e. the population in the next generation) is controlled indirectly by the spread

[1] The notion "evolutionary algorithm" is used here to refer to *all* classes of algorithms inspired from Darwin's evolutionary principles, including evolutionary programming, evolution strategies, and genetic algorithms.

of parent population, convergence, divergence, or adapting to changing fitness landscapes are possible to achieve. These crossover operators create one or two children according to a probability distribution over two or more parent solutions. The most popular approach has been to use a uniform probability distribution ("blend crossover," BLX suggested in [26]) around a region bracketing the parent solutions. There exist at least three different approaches where a non-uniform probability distribution has been suggested. Of them, the simulated binary crossover (SBX) uses a bimodal probability distribution with its mode at the parent solutions [27]. This is similar to fuzzy recombination operator (FR) which also uses a bimodal distribution, but the distribution around each parent is always triangular [25]. Ono and Kobayashi [28] suggested a unimodal normally distributed crossover (UNDX) operator which uses an ellipsoidal probability distribution around three parents to create an offspring.

The authors [23] and Kita [24] have shown that SBX and UNDX operators can provide real-parameter GAs self-adaptive behavior similar to that of self-adaptive ESs on a number of problems. A previous study [22] has drawn a connection between the working principle of real-parameter GAs with SBX and isotropic SA-ESs. GAs with SBX uses a probability distribution that is controlled adaptively in proportion to the difference in parent solutions, whereas a self-adaptive ES uses a probability distribution of similar nature but is controlled by explicit mutation strength parameters, which are also evolved along with decision variables. Although this connection was not made any more rigorous, it was felt that there exists a much deeper connection between the workings of real-parameter GAs with SBX and other similar crossover operators and self-adaptive ESs. In this paper, we attempt to make that connection between their workings more rigorous by calculating the mean and variance of the population from one generation to another. Our arguments are supported by performing simulations on a number of test problems.

In the remainder of this paper, we present some postulates on the behavior of SA-EAs on a number of fitness landscapes. Thereafter, we investigate the properties of three crossover operators (BLX, SBX, and FR) and SA-ESs. Theoretical calculations of population mean and variance (mainly in flat fitness landscapes) are verified with simulations. Although the implementations of these EAs are different, the analysis shows that the underlying working principles lead to very similar dynamical behavior, if appropriate characteristic parameters are chosen for each operator.

II. Some Postulates on the Behavior of Self-Adaptive Evolutionary Algorithms in $\mathbb{R}^N$

In this section, postulates on the behavior of SA-EAs in unbounded real-valued search spaces $\mathbb{R}^N$ (N – search space dimension) are proposed. Most of these postulates are arrived at by observing the properties of a number of successful self-adaptive EA implementations.

In the discussion here, it is helpful to subdivide an evolutionary algorithm into two major components – reproduction and variation (in the usual EAs, the latter refers to recombination and mutation operators). In our arguments, the sequence of their operations is not important. Under the reproduction operation, a population of μ solutions is expected to lose some inferior solutions, only to make duplicates of some good solutions. This process of deleting some solutions and making duplicates of some other solutions can, in general, reduce the variance of the population. That is, the variance of population in the search space after the reproduction operation is, in most situations, smaller than before the reproduction operation. This is particularly true for most instances of unimodal or linear fitness landscapes. For multimodal fitness landscapes, this may not be true always, particularly when the solutions near the population mean are bad solutions. While the variance may increase in these cases after the reproduction operation, it is important to note that the maximum spread[2] of the population does not increase due to the reproduction operator, because no new solution is created by the reproduction operator.

Since reproduction tends to reduce variation (in general), it is important that the variation operators possess certain properties in order to adequately counteract this trend. Providing general rules depict how to quantitatively balance the effects of these two "antagonistic" operators seems impossible. However, it is possible to formulate some minimal (necessary) requirements and/or "desirable properties" that the variation operators and the EA as a whole, respectively, should fulfill in order to operate satisfactorily in a given problem domain. In the following, we present such postulates concerning the behavior of the variation operators and the EA as a whole.

A. Postulates on population mean

The purpose of a variation operator is to use the parent population formed after the reproduction operation to create an offspring population. Most EA variation operators do not use any fitness information, instead partial information between parents is shared and rearranged, respectively. For example, it was argued in [29] that the purpose of the reproduction operator is to *exploit* the search space by emphasizing good solutions and the purpose of a variation operator is to *explore* the search space.[3] If both operators use fitness information, there may be too much emphasis on the exploitation aspect and resulting GAs may prematurely converge to a suboptimal solution. Put another way, reproduction (or selection) operators use the information from the fitness space, whereas the variation operator explores

[2]The maximum distance between any two solutions.

[3]Note, up to now there are no generally accepted definitions and general consensus as to the notions of "exploration" and "exploitation." For a discussion of these notions in ES/EP, see [30].

the search region indicated by the parent population. Since an ideal variation operator may only process search space information of the selected population, the variation operator should not introduce any bias on that population level. Therefore, we postulate the following:

Postulate 1: Under a variation operator, the expected population mean should remain unchanged.

Remark 1: The postulate also makes sense intuitively. Since the variation operator, in general, does not use fitness information, there is no reason for it to shift the parent population mean in any direction in the search space. This is precisely the task of the reproduction operator in leading the population in an appropriate direction in the search space. The best the variation operator can do is to keep the population mean invariant while increasing or reducing the variance of the resulting population. In the following subsection, we describe postulates for increasing or decreasing population variance under a variation operator.

Remark 2: Standard mutation operators in ES/EP do not use fitness information. Instead they use a zero-mean probability distribution to create a mutated offspring. Thus, the expected population mean is not changed under a standard mutation operator. As we will see below, this is fulfilled also for the standard recombination operators in ESs and for the real-coded GA variation operators.

Remark 3: This also holds for the standard binary GA recombination operators one-point, n-point, and uniform crossover in the genotype search space $\mathbb{B}^\ell$ ($\mathbb{B} = \{0, 1\}$, ℓ – string length) because these operators only rearrange the alleles. Interestingly, it also holds for the resulting phenotype population when using standard binary to real $\mathbb{B}^\ell \mapsto \mathbb{R}^N$ encoding. (For a proof, see Appendix I.) Unlike crossover, mutation in binary GAs *does not* fulfill Postulate 1. This might serve as an argument as to why mutation in standard binary GAs with binary to real or integer encoding must be treated as a background operator applied with a small probability only and has led researchers to rely on mutation operators with a zero-mean probability distribution, such as the non-uniform mutation operator [31] or a creep mutation operator [8].

B. *Postulates on population variance*

It was argued earlier that depending on the fitness landscape and the position of the parental population, the population variance may decrease or increase after the reproduction operation. Thus, to avoid any premature convergence or stagnation, the population resulting after variation must adjust its variance accordingly, so as to keep the overall population variance at a reasonable value from generation to generation. Particularly, if the reproduction operator reduces the population variance, the variation operator should increase the variance of the population. This also becomes clear when considering flat and linear fitness landscapes discussed below.

In the following, we consider SA strategies for a few fitness landscapes. Some of them, especially the most simple ones – the flat and linear fitness landscapes – may be regarded as *local approximations* of the real fitness function to be optimized. Therefore, the proposed SA strategies describe local EA behavior with the general objective of increasing the local performance (fitness improvement per generation).

B.1 Flat fitness landscapes

Postulate 2: The expected population variance $\sigma^2[y]$ should exponentially increase[4] with generation number g. (For the definition of $\sigma^2[y]$, see Section III-B, Eq. (21).)

Remark 4: The intention here is to leave the region of constancy as fast as possible, simply because there is no selective preference for those parts of the search space occupied by the population currently. Under such conditions, the best an EA can do is to walk randomly through the search space trying larger and larger steps. This can increase the probability of finding a region of improved fitness. We would like to mention that this postulate is different from that in Ostermeier [32, p. 18], which suggests a zero drift in a similar situation.

B.2 Linear fitness landscapes (hyperplane)

Postulate 3: The expected population variance $\sigma^2[y]$ should increase exponentially with g.

Remark 5: Given a parental state in the search space, linear fitness landscapes divide the search space (with respect to the fitness) into two subspaces separated by a hyperplane. Considering isotropic mutations,[5] on average every other mutation (starting from a parental point) yields an improvement independent of the mutation strength used (i.e., the success probability is equal to 1/2). However, the larger the mutation strength, the better the expected improvement (fitness gain). Therefore, the mutation strength and the population variance, respectively, should increase over the generations. The same argument can be made for a recombination operator.

Remark 6: Instead of the above postulate on population variance, one can postulate that the distance traveled along the direction of the optimum by the parents should increase exponentially in order to run through the linear region exponentially fast.

[4] Exponential increase may be regarded as a strong form of SA. As a weak variant one might postulate expected monotonic increase.

[5] An isotropic mutation $\mathbf{z} \in \mathbb{R}^N$ is a random perturbation the density of which depends only on the length $||\mathbf{z}||$ but not on the direction of $\mathbf{z}$ in the search space $\mathbb{R}^N$. That is, the probability density function is of the special type $p(||\mathbf{z}||)$.

B.3 Unimodal fitness landscapes

If the parent population brackets the optimum, a controlled reduction in population variance by the variation operator is desired. Whereas, if the parent population does not bracket the optimum, the situation is similar to that in the linear fitness landscape and the variation operator should increase the population variance in order to reach the optimum quickly. For arbitrary fitness functions in unrestricted search spaces, the latter is more common because the location of the optimum is not usually known. Therefore, it seems better to use a variation operator that increases the population variance. The necessary variance reduction of the population during the end phase of optimization is accomplished by the selection operator.

B.4 Multimodal fitness landscapes

The increase or decrease of population variance should largely depend on the fitness landscape and the placement of the parent population in the search space. We argue that even in these cases, a (good) generic strategy would be to use a variation operator that increases the population variance from one generation to the next and then introduce adequate selection pressure by the reproduction operator to exploit the offspring population. We believe that this strategy has a better global perspective than using variation operators that would reduce the population variance.

B.5 Moving target problems (time-dependent optimum) and rapidly changing fitness landscapes

In both cases it may be of importance to *increase* the mutation strength/population variance after a period of shrinking. Therefore, the algorithms should have the ability to increase the variance. A simple way to ensure such a behavior is to implement a certain bias toward an increase of the population variance.

B.6 Rotational invariance

The above postulates should hold locally for any coordinate direction independently. The overall fitness function may not be flat with respect to all coordinate directions but may be selectively neutral with respect to some coordinate directions. Postulate 2 for the flat fitness landscape case should then hold for those coordinate directions

$$\forall\, \delta \in [-\delta_l, \delta_u]: \quad f(y_1, \ldots, y_k, \ldots, y_N) = f(y_1, \ldots, y_k + \delta, \ldots, y_N). \tag{1}$$

Similarly, Postulate 3 should then hold for those coordinate directions that are linearly separable (the linear fitness landscape case)

$$\forall\, \kappa \in [-\kappa_l, \kappa_u]: \quad f(y_1, \ldots, \kappa y_k, \ldots, y_N) = f(y_1, \ldots, y_k, \ldots, y_N) + \kappa f_k. \tag{2}$$

Remark 7: Empirical studies on quadratic ridge functions [22] have shown that Postulate 2 with (1) and also Postulate 3 with (2) are fulfilled in real-coded GAs with e.g. SBX recombination operator (for its definition, see below) and SA-ES versions using non-isotropic mutation distributions. That is, these EA versions exhibit an exponential increase of the population variance in the y_k direction. However, SA-ES with isotropic mutations exhibit only a linear increase fulfilling the "weak" versions of these postulates (see footnote 4).

A much harder requirement on the SA abilities arises when (1) and (2) are generalized for arbitrary search space directions $\mathbf{v}$ ($\|\mathbf{v}\| = 1$). The Postulate 2 for the flat fitness landscape should hold locally (with respect to a given parental state $\mathbf{y}$) for each direction $\mathbf{v}$ obeying

$$\forall\, \delta \in [-\delta_l, \delta_u]: \quad f(\mathbf{y} + \delta\mathbf{v}) = f(\mathbf{y}) \qquad \Rightarrow \qquad \mathbf{v}^{\mathrm{T}}\nabla f = 0. \tag{3}$$

Similarly, Postulate 3 should hold for

$$\forall\, \kappa \in [-\kappa_l, \kappa_u]: \quad f(\mathbf{y} + \kappa\mathbf{v}) = f(\mathbf{y}) + \kappa f_{\mathbf{v}} = f(\mathbf{y}) + \kappa\mathbf{v}^{\mathrm{T}}\nabla f. \tag{4}$$

Note that Eq. (3) can be regarded as a special case of (4).

Remark 8: Empirical studies on rotated quadratic ridge functions performed so far [22] indicate that the standard SA techniques are not able to exponentially increase the variance in an arbitrary $\mathbf{v}$ direction. Instead, one observes a linear increase. It remains as an open question whether there exists a variation operator relying on the parental population of size $\mu = \mathcal{O}(N)$ (or smaller) which may exhibit exponential behavior on such functions.

C. Behavior of SA-EAs on standard test fitness landscapes

Before we begin the analysis of the real-parameter crossover operators, we mention a number of test fitness landscapes where self-adaptive EAs, especially ES/EP, have been applied.

C.1 Sphere models

The sphere models are defined as

$$f(\mathbf{y}) := g(R), \quad \text{with} \quad R := \|\mathbf{y} - \hat{\mathbf{y}}\|. \tag{5}$$

There $g(x)$, $x \in \mathbb{R}$, $x \geq 0$, is usually a monotonic function depending on the distance of $\mathbf{y}$ to the optimum point $\hat{\mathbf{y}}$ only. The objective is to minimize $f(\mathbf{y})$.

A SA-EA should yield linear convergence order on sphere models, that is, one expects linearly falling curves of the logarithm of the residual distance of the population mean to the optimum (in the search space) with the number of generations, or mathematically

$$R(g) = R(0)\exp(-ag), \tag{6}$$

where a is the rate of decrease of $\log R$. This is what has been observed in self-adaptive ESs (e.g., see [14], [17]), and it seems reasonable to demand a similar behavior for other SA-EAs in unbounded search spaces. The efficiency of a SA-EA depends on how large the exponent a is. On the sphere fitness functions there exist theoretical estimates of the parameter a for different self-adaptive ESs and EP [15].

C.2 Ridge functions

The ridge models [33], [34], [35] are defined as

$$f_R(\mathbf{y}) := \mathbf{v}^{\mathrm{T}}\mathbf{y} - d\left[\sqrt{[(\mathbf{v}^{\mathrm{T}}\mathbf{y})\mathbf{v} - \mathbf{y}]^2}\right]^{\alpha}, \quad d \geq 0 \tag{7}$$

with ridge direction $\mathbf{v}$, $\mathbf{v}^{\mathrm{T}}\mathbf{v} = 1$, also referred to as ridge axis, along which the population is expected to move. The exponent α determines the ridge degree. The special cases $\alpha = 2$ and $\alpha = 1$ are known as parabolic ridge and sharp ridge, respectively. The objective is to maximize $f_R(\mathbf{y})$.

A SA-EA is expected to yield a linear divergence order on ridge models, where the optimum lies at infinity along the ridge axis. It has been shown experimentally for the *parabolic* ridge [23] that if the ridge axis coincides with a coordinate axis, a linear order

$$r(g) = r(0)\exp(ag) \tag{8}$$

can be achieved, where $r(g)$ is a measure of distance traveled along the ridge axis $\mathbf{v}$ by the population from a fixed point on the ridge at generation g. Whereas, if the ridge axis $\mathbf{v}$ is any arbitrary non-coordinate direction, a constant order

$$r(g) = r(0) + ag, \tag{9}$$

has been observed in [23]. Here it becomes obvious that the performance behavior (9) violates our Postulate 3 for functions arbitrarily rotated according to Eq. (4). It is an open question whether there exist SA-EAs that use parental population information only, e.g., no further history, but exhibit an exponential behavior, as in Eq. (8). If one uses control strategies that rely on a history, see e.g., [21], and/or population of size $\mu = \mathrm{O}(N^2)$ it might be possible to attain the behavior (8).

Getting appropriate adaptive behavior on the *sharp* ridge, i.e., $\alpha = 1$, with d sufficiently large in (7), appears as a problem for standard SA-ESs, as first noticed by Herdy [33, p.213]. Again it is not clear whether there are SA-EAs, relying on the parental population only, which are able to achieve at least the behavior (9) for large d.[6]

C.3 Commonly used fitness landscapes

Besides these two extreme cases of convergence and divergence properties that a SA-EA should exhibit, they should also be able to adapt to a number of other landscapes [19], [23], [21]:

Elliptic landscapes: Unlike in the sphere model, the emphasis of each coordinate direction is unequal in the fitness functions. In these functions, a SA-EA is also expected to exhibit a linear order convergence.

Time-varying fitness landscapes: The fitness function changes when an EA population has converged sufficiently near the current optimum. This also tests a SA-EA's ability to find the optimum even when the initialization is performed away from the optimum.

Fitness landscapes with non-zero correlations among variables: This tests a SA-EA's ability to adapt to fitness landscapes which are not linearly separable functions of decision variables. Because of the non-zero correlations among the variables, a linear order convergence towards the optimum may be harder, but is desirable from a SA-EA. The *rotated* versions of the elliptic landscapes or any of the above functions belong to this class of functions.

Multimodal fitness landscapes: A SA-EA should be able to avoid local optima and converge to the global optimum. After an initial transient, a linear order convergence is expected with a SA-EA.

[6]Note, for sufficiently small d, i.e. $d \to 0$, the fitness landscape degenerates to a hyperplane. For that case, SA-EAs have no difficulty in achieving the dynamics of (9).

C.4 Other fitness landscapes and modes of theoretical analysis

The above mentioned fitness functions are landscapes where SA-EAs have been tested in the past. There may exist other landscapes that would test different adapting abilities of the SA-EA. We recognize the need for a more thorough study on test function development for SA-EA, both experimentally and theoretically, but we do not belabor it here. Instead, we analyze variation operators for a number of EAs and suggest when each will exhibit adequate self-adaptive properties with respect to our postulates.

A theoretical analysis can be performed only to some special aspects of SA. Due to the enormous technical difficulties arising at such analysis, there are in principle three possibilities to tackle that matter:
(a) building models of the EA considered, i.e., simplification of the real algorithm,
(b) using a one-to-one mapping of the EA to theory, but considering sufficiently simple fitness functions,
(c) using a mixture of both (a) and (b)
In this paper we prefer approach (b), i.e., we will work with the real EA, but consider only simple fitness functions. That is why we have presented postulates on the simple fitness landscapes.

We first analyze the operators for flat fitness landscapes. This is done for the GA recombination operators in the next section (Section III). A similar analysis for the ES recombination operators is performed in Section V. The results of Section III are used in Section IV to perform a "fair" comparison of the performance of the GA recombination operators on the sphere model. Choosing the parameters of these operators in such a way that they perform similarly in the flat fitness space leads to linear convergence order for all operators at the sphere model. Section VI compares the results from III and V for the evolution of population variance in flat fitness landscapes. Section VII reviews the current state of analyzing the linear fitness landscapes.

III. Analysis of Crossover Operators for Real-coded GAs in flat fitness landscapes

In this section, we analyze a number of crossover operators used in real-coded GAs. Specifically, the mean and variance of the offspring population are derived from the known distribution parameters of the parent population. These calculations then establish the basis for the determination of the dynamic behavior of the respective GAs. The theoretical predictions are compared with simulations.

A. Commonly-used recombination operators

Recombination in GAs is inspired mainly by the crossing over observed in nature. The standard procedure of recombination takes two parents and generates *two* offspring.[7] Since offspring are produced in pairs, we have $\mu/2$ pairs (that is, an even μ is assumed) those individuals will be numbered by 1 and 2 and by an index counter k running from 1 to $\mu/2$.

The recombination operator proposed by Deb and Agrawal [27], called simulated binary crossover (SBX), is defined as

$$\left\{ \begin{array}{l} y_{1,k} := \frac{1}{2}\left[(1-\beta_k)x_{1,k} + (1+\beta_k)x_{2,k}\right] \\ y_{2,k} := \frac{1}{2}\left[(1+\beta_k)x_{1,k} + (1-\beta_k)x_{2,k}\right]. \end{array} \right. \tag{10}$$

Here, $x_{1,k}$ and $x_{2,k}$ are independent samples from the parental population and β_k (≥ 0) is a sample from a random number generator having the density

$$p(\beta) := \left\{ \begin{array}{ll} \frac{1}{2}(\eta+1)\beta^{\eta}, & \text{if} \quad 0 \leq \beta \leq 1 \\ \frac{1}{2}(\eta+1)\frac{1}{\beta^{\eta+2}}, & \text{if} \quad \beta > 1. \end{array} \right. \tag{11}$$

This distribution can easily be obtained from a uniform $\mathrm{u}(0,1)$ random number source by the transformation

$$\text{SBX}: \quad \beta(\mathrm{u}) := \left\{ \begin{array}{ll} (2\mathrm{u})^{\frac{1}{\eta+1}}, & \text{if} \quad \mathrm{u}(0,1) \leq \frac{1}{2} \\ \left[2(1-\mathrm{u})\right]^{-\frac{1}{\eta+1}}, & \text{if} \quad \mathrm{u}(0,1) > \frac{1}{2}. \end{array} \right. \tag{12}$$

Note, the analysis to be performed is *not* restricted to the distribution (11). In addition we will investigate the BLX operator of Eshelman and Schaffer [26], and the so-called "fuzzy recombination" by Voigt et al. [25].

First, BLX is considered. By using the transformation

$$\beta = 2\xi - 1 \tag{13}$$

applied to Eq. (10), one obtains

$$\left\{ \begin{array}{llll} y_{1,k} := & (1-\xi_k)\, x_{1,k} & + & \xi_k\, x_{2,k} \\ y_{2,k} := & \xi_k\, x_{1,k} & + & (1-\xi_k)\, x_{2,k} \end{array} \right. \tag{14}$$

[7] From the algorithmic point of view, one might also consider operators producing only one offspring, as in evolution strategies (see Section V).

Now, choosing ξ uniformly from $[-\alpha, 1+\alpha]$, we arrive at Eshelman's and Schaffer's [26] blend crossover

$$\text{BLX-}\alpha: \quad \xi := \mathrm{u}[-\alpha, 1+\alpha]. \tag{15}$$

The FR operator (fuzzy recombination) proposed by Voigt et al. [25] is similar to the SBX defined by (11) in the sense that it puts emphasize on the parental states: The probability density of the offspring y_i is maximal at the parental x_i. That is, the density of β in (10) has its maximum at $\beta = 1$. The only difference between SBX and FR is in the shape of $p(\beta)$. The shape of FR is triangular. Using random numbers ζ obeying the triangular distribution density $p_\triangle(\zeta)$

$$p_\triangle(\zeta) := \begin{cases} \zeta + 1, & \text{if} \quad -1 \le \zeta < 0 \\ 1 - \zeta, & \text{if} \quad 0 \le \zeta \le 1, \end{cases} \tag{16}$$

the β value to be used in (10) is obtained by the transformation

$$\text{“Fuzzy” recombination (FR)}: \quad \beta = 1 + 2\zeta d. \tag{17}$$

Here, d is a strategy parameter similar to α in (15). It determines how far offspring can be located from the parents. The random number ζ with triangle distribution (16) can be obtained by the standard method of inversion, or more easily, as the sum of two independent uniformly distributed random numbers $\mathrm{u}(0,1)$:

$$\zeta := \mathrm{u}_1 + \mathrm{u}_2 - 1. \tag{18}$$

The above crossover operators can be considered as a hybrid of a $(\mu/2, \lambda)$-ES type “intermediate” recombination operator followed by a perturbation operator – very similar to a mutation operator – whose variance depends on the difference of the parents in the search space. This can be seen if we rewrite the first equation of (10) as follows

$$y_{1,k} := (x_{1,k} + x_{2,k})/2 + \beta_k (x_{2,k} - x_{1,k})/2. \tag{19}$$

The distribution of the perturbation operator takes the shape of the distribution of β. For SBX and FR, this distribution has a nonzero mean and a mode (maximum) at $\beta = 1$, whereas for BLX, this distribution has a zero mean but is uniform within a certain interval. We argue that a variation operator that either uses directly the difference of the parents in the search space or uses an explicit evolution of a strategy parameter controlling the spread of offspring (as in self-adaptive ESs) is essential for the resulting EA to exhibit self-adaptive behavior on the population level (for a detailed discussion, see [22]). For optimum performance (i.e. making maximal local progress), it is then a matter of choosing the appropriate characteristic parameter of the underlying variation operator.

B. Definition of the variance measure

The crossover operator picks two individuals (considering only one component from the vectors) from the parental generation (g) and produces two offspring for generation ($g+1$). For notational simplicity, denote an individual component by $x := x^{(g)}$ and the offspring by $y := x^{(g+1)}$. The population variance of the offspring is defined as [36]

$$\mathrm{Var}\{y\} := \frac{1}{\mu} \sum_{m=1}^{\mu} (y_m - \langle y \rangle)^2. \tag{20}$$

Here the subscript m is the individual counter running from one to the population size μ and $\langle y \rangle$ is the average (center of mass) individual. It is the aim of this section to calculate the *expected* value of $\mathrm{Var}\{y\}$, that is,

$$\sigma^2[y] := \mathrm{E}[\mathrm{Var}\{y\}] = \overline{\mathrm{Var}\{y\}}, \qquad \text{NB:} \quad \mathrm{E}[z] \equiv \overline{z}, \tag{21}$$

depending on the *expected* parental population variance, denoted by $\sigma^2[x]$ which is by definition

$$\sigma^2[x] := \mathrm{E}[(x - \overline{x})^2] = \overline{x^2} - \overline{x}^2. \tag{22}$$

C. Calculation of the population average $\langle y \rangle$

The population average is given by

$$\langle y \rangle := \frac{1}{\mu} \sum_{m=1}^{\mu} y_m. \tag{23}$$

Since the crossover operator produces two offspring using (10), one can rearrange (23) to

$$\langle y \rangle := \frac{1}{\mu}\sum_{k=1}^{\mu/2}(y_{1,k} + y_{2,k}). \tag{24}$$

By inserting (10) into (24) we obtain

$$\langle y \rangle := \frac{1}{\mu}\sum_{k=1}^{\mu/2}(x_{1,k} + x_{2,k}) = \frac{1}{\mu}\sum_{m=1}^{\mu} x_m. \tag{25}$$

Here it was taken into account that the $x_{1,k}$, $x_{2,k}$ are independent samples from the parental population and the x_m is just a renumbering of the individuals. Thus, the crossover operator preserves the population mean of the parental population. As discussed earlier, this behavior is considered as a desired property of crossover operators formulated by Postulate 1. Note, according to (10), for crossover operators (that is, producing *two* individuals) this also holds for each pair of offspring produced independent of the distribution of the random variate β used. However, this does not necessarily hold for the expectation of a single offspring. In the latter case, it depends on the $p(\beta)$ density.

D. *Calculation of the population variance*

Starting from definition (20), one can write alternatively

$$\mathrm{Var}\{y\} := \frac{1}{\mu}\sum_{m=1}^{\mu}(y_m - \langle y \rangle)^2 = \frac{1}{\mu}\sum_{k=1}^{\mu/2}\left[(y_{1,k} - \langle y \rangle)^2 + (y_{2,k} - \langle y \rangle)^2\right]. \tag{26}$$

Taking (10) and (25) into account, the $\mathrm{Var}\{y\}$ consists of a sum over $x_{i,k}x_{j,l}$ products. Since the final goal is to calculate $\sigma^2[y]$, the expected value of (26) is to be determined. This requires the calculation of the expected values of $x_{i,k}x_{j,l}$. Since the sampling process picks individuals at random from the parental pool during the crossover operator, all $x_{i,k}$ are independent and identically distributed. Therefore one has

$$\overline{x_{i,k}x_{j,l}} := \begin{cases} \overline{x^2}, & \text{if} \quad i = j \ \wedge \ k = l \\ \overline{x}^2, & \text{otherwise} \end{cases} \tag{27}$$

where $\overline{x}$ and $\overline{x^2}$ are the first- and second-order moments, respectively, of the parental population distribution. In order to calculate the expectation of (26) the square brackets in (26) must be rearranged in such a way that products $x_{i,k}x_{j,l}$ for which $i = j \ \wedge \ k = l$ are separated from the rest. The straightforward but lengthy calculation can be found in the Appendix II. As a result, one obtains the *evolution equation of the expected population variance* as follows:

$$\sigma^{2\ (g+1)} = \frac{1}{2}\left(1 - \frac{2}{\mu} + \overline{\beta^2}\right)\sigma^{2\ (g)}. \tag{28}$$

Its solution by recurrence leads to the equation of expected variance dynamics

$$\sigma^2[y]^{(g)} = \sigma^2[y]^{(0)}\left(\frac{1}{2} - \frac{1}{\mu} + \frac{\overline{\beta^2}}{2}\right)^g. \tag{29}$$

One observes an exponential change of the population variance. Collapsing or exploding of the population variance is thus controlled by μ and $\overline{\beta^2}$. This is calculated for some typical crossover operators below.

E. *Typical Crossover Operators*

It is now the aim to investigate the evolution equation (29) for the crossover operators SBX, BLX, and FR. The evolution equation (29) contains the $\overline{\beta^2}$ term, which depends on the random number distribution used. By definition, one has

$$\overline{\beta^2} = \int \beta^2 p(\beta)\mathrm{d}\beta. \tag{30}$$

For SBX and BLX the standard way of generating β is via transformation from the uniform variate $u = \mathrm{u}(0,1)$; one can write alternatively

$$\overline{\beta^2} = \int_{u=0}^{u=1}(\beta(u))^2\mathrm{d}u. \tag{31}$$

As to FR, Eq. (16) has to be used in order to calculate $\overline{\beta^2}$

$$\overline{\beta^2} = \int_{\zeta=-1}^{\zeta=1} (\beta(\zeta))^2 p_{\triangle}(\zeta)\mathrm{d}\zeta. \tag{32}$$

E.1 SBX-η

The SBX uses a random number distribution according to (12). The calculation of the second moment $\overline{\beta^2}$ is straightforward. Using (31) one has with (12)

$$\overline{\beta^2} = \int_{u=0}^{u=1/2} (2u)^{\frac{2}{\eta+1}} \mathrm{d}u + \int_{u=1/2}^{u=1} [2(1-u)]^{-\frac{2}{\eta+1}} \mathrm{d}u. \tag{33}$$

After substitution $s := 2u$ and $t := 2(1-u)$ for the first and the second integral, respectively, one ends up with the following equation for $\eta \neq 1$:

$$\overline{\beta^2} = \frac{1}{2}\left[\frac{1}{1+\frac{2}{\eta+1}} s^{\left(1+\frac{2}{\eta+1}\right)}\right]_0^1 + \frac{1}{2}\left[\frac{1}{1-\frac{2}{\eta+1}} t^{\left(1-\frac{2}{\eta+1}\right)}\right]_0^1 \tag{34}$$

which gives for $\eta > 1$

$$\overline{\beta^2} = \frac{1}{2}\left[\frac{\eta+1}{\eta+1+2} + \frac{\eta+1}{\eta+1-2}\right], \tag{35}$$

and finally

$$\overline{\beta^2} = \frac{(\eta+1)^2}{(\eta+3)(\eta-1)}. \tag{36}$$

The case $\eta \leq 1$ is not covered by the above equation. From Eq. (34), one immediately obtains $\overline{\beta^2} \to \infty$, that is, the second moment does not exist. In practice, however, the evolution of variance is restricted by the finite number representation of the computer.

Equation (36) along with (29) leads to the variance dynamics of the SBX operator for $\eta > 1$

$$\sigma^2[y]^{(g)} = \sigma^2[y]^{(0)} \left(1 + \frac{2}{(\eta+3)(\eta-1)} - \frac{1}{\mu}\right)^g. \tag{37}$$

Depending on the choice of η, the expected population can expand or contract for a given population size μ. Since variance increase is the desired behavior (see Postulate 2 in Section II-B), the terms in the large parenthesis of (37) must be greater than one. Therefore, one obtains the following expression for the minimal population size:

$$\text{population size:} \quad \mu > \frac{1}{2}(\eta+3)(\eta-1). \tag{38}$$

E.2 BLX-α

Starting from Eq. (30) and using (15) which leads to $p(\xi) = 1/(1+2\alpha)$ and taking (13) into account, one obtains the integral

$$\begin{aligned}
\overline{\beta^2} &= \frac{1}{1+2\alpha} \int_{\xi=-\alpha}^{\xi=1+\alpha} (2\xi-1)^2 \mathrm{d}\xi \\
&= \frac{1}{1+2\alpha} \int_{\xi=-\alpha}^{\xi=1+\alpha} (4\xi^2 - 4\xi + 1)\mathrm{d}\xi \\
&= \frac{1}{1+2\alpha} \left[\frac{4}{3}\xi^3 - 2\xi^2 + \xi\right]_{-\alpha}^{1+\alpha}
\end{aligned} \tag{39}$$

and finally

$$\overline{\beta^2} = \frac{1}{3}(2\alpha+1)^2. \tag{40}$$

Thus we get with (28) the evolution equation

$$\sigma^2[y]^{(g)} = \sigma^2[y]^{(0)} \left(1 + \frac{1}{3}(2\alpha^2 + 2\alpha - 1) - \frac{1}{\mu}\right)^g . \tag{41}$$

Again, the contraction or expansion of the population variance depends on the choice of α. For example, $\alpha = 0$ (BLX-0.0) leads immediately to a contracting population, no matter how μ is chosen.

In [26], Eshelman and Schaffer suggested $\alpha = 1/2$ (BLX-0.5) because: "Only when $\alpha = 0.5$ does the probability that an offspring will lie outside its parents become equal to the probability that it will lie between its parents." Furthermore they inferred [26, p. 193]: "In other words, $\alpha = 0.5$ balances the convergent and divergent tendencies in the absence of selection pressure." While their former statement is true, the latter is true only for a particular population size μ. From Eq. (41) for $\alpha = 1/2$, the variance ratio becomes

$$\text{BLX-0.5:} \quad \frac{\sigma^{2\ (g+1)}}{\sigma^{2\ (g)}} = \frac{7}{6} - \frac{1}{\mu}. \tag{42}$$

That is, depending on μ the population converges or diverges. A simple calculation yields:

$$\text{BLX-0.5 divergence:} \quad \mu > 6. \tag{43}$$

Only for $\mu = 6$ convergent and divergent tendencies are balanced by BLX-0.5. This result is valid for GAs with random sampling, where the parents chosen for crossover are drawn at random from the parental pool of size μ. The CHC-GA of Eshelman [37] uses a more sophisticated selection technique in order to prevent "incest." Even in this case, there still remains a random sampling component producing similar results for the σ^2 dynamics. Only a study of the population variance evolution under CHC can reveal the true scenario.

Since a diverging population in flat fitness landscapes is desired (see Postulate 2 in Section II-B) in order to have the desired SA behavior, we may ask how to choose α given a fixed population size μ. Divergence is achieved when the large parenthesis in (41) is greater than 1. Resolving for α, one obtains

$$\alpha > \frac{\sqrt{3 + 6/\mu} - 1}{2}. \tag{44}$$

Similar to (38), one can calculate the minimal population size. By the criterion $\sigma^2[y]^{(g)} > \sigma^2[y]^{(0)}$ applied to (41) one finds

$$\text{population size:} \quad \mu > \frac{3}{2\alpha + 2\alpha^2 - 1}. \tag{45}$$

E.3 FR-d

In the case of the fuzzy recombination, the calculation of $\overline{\beta^2}$ starts from (32) and uses (16) and (17)

$$\overline{\beta^2} = \int_{\zeta=-1}^{\zeta=0} (1 + 2\zeta d)^2 (\zeta + 1) \mathrm{d}\zeta + \int_{\zeta=0}^{\zeta=1} (1 + 2\zeta d)^2 (1 - \zeta) \mathrm{d}\zeta. \tag{46}$$

The integration can be carried out to obtain

$$\overline{\beta^2} = 1 + \frac{2}{3} d^2. \tag{47}$$

This, with (29), yields the variance dynamics of the FR operator

$$\sigma^2[y]^{(g)} = \sigma^2[y]^{(0)} \left(1 + \frac{1}{3} d^2 - \frac{1}{\mu}\right)^g . \tag{48}$$

We can now determine the minimum population size by demanding that $\sigma^2[y]^{(g)} > \sigma^2[y]^{(0)}$ for flat fitness landscapes:

$$\text{population size:} \quad \mu > \frac{3}{d^2}. \tag{49}$$

In [25, p. 106], Voigt et al. suggested $d \geq 1/2$ and as a rule of thumb they used $d = 1/2$. In this case, our postulate demands $\mu > 12$. Interestingly, Voigt et al. realized that there is a critical population size (which they have denoted by N^* in [25]) below which the GA does not work appropriately [25, p. 107] and concluded that n^* depends on three other parameters except d. However, Eq. (49) shows that the minimum population size for a monotonic increase in population variance depends on the parameter d.

F. Some simulation examples

The predictive quality of the evolution equation (29) can be evaluated for the SBX, BLX, and FR crossover operators by making *real-parameter* GA runs on a flat fitness landscape (i.e. the fitness is constant). As a selection operator, the binary tournament or proportionate selection can be used. Figure 1 shows theoretical (37) and experimental population variance dynamics with simulations using the SBX operator for different values of η. Figure 2 shows GA runs using the BLX operator compared with the theoretical predictions of Eq. (41) and Figure 3 displays GA runs for the "fuzzy" recombination operator compared with Eq. (48). Each of these operators can exhibit exponentially increasing as well

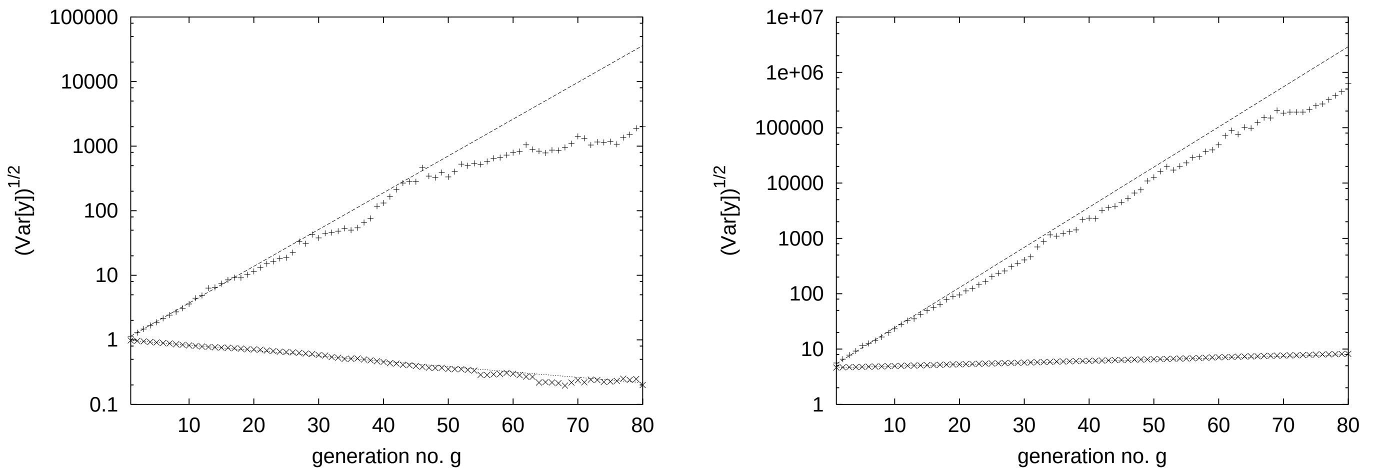

Fig. 1. Evolution of the square root of the expected population variance for the SBX operator in the flat fitness landscape. Left picture: population size 10 with $\eta = 2$ (upper, dashed line, simulations displayed by "+") and $\eta = 5$ (lower, dotted line, simulations displayed by "$\times$"). The population has been randomly initialized with variance 1. Right picture: population size 300 with $\eta = 2$ (upper, dashed line, simulations displayed by "+") and $\eta = 10$ (lower line, simulations displayed by "$\diamond$"). The population has been randomly initialized with variance 21. The simulation results have been obtained by averaging over 100 (pop-size = 300) and 3000 (pop-size = 10), respectively, independent runs.

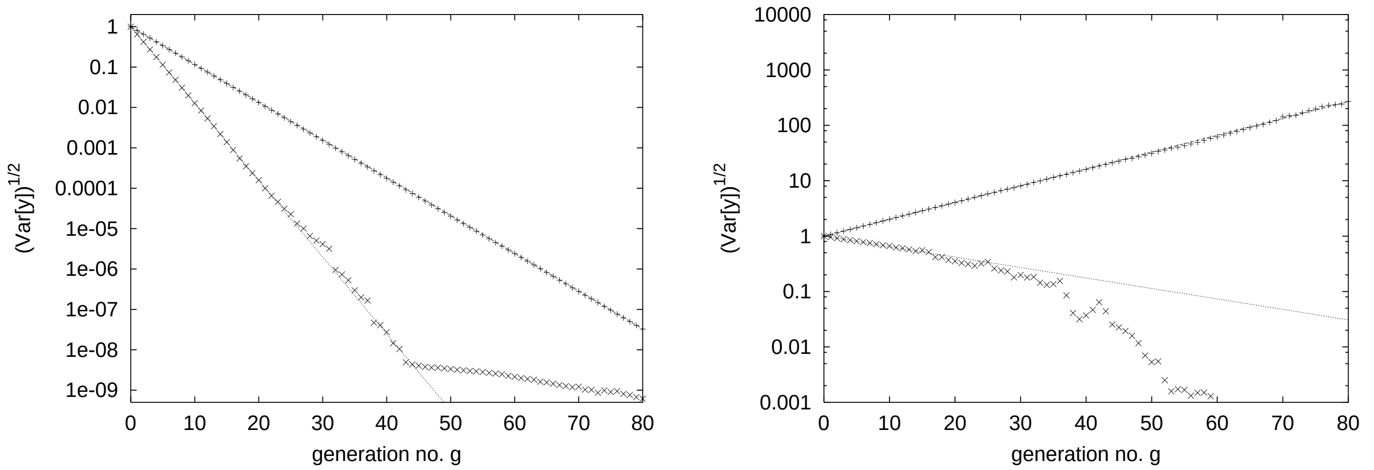

Fig. 2. Evolution of the square root of the expected population variance for the BLX operator in the flat fitness landscape. Left picture: BLX-0.0 for population size 60 (upper, dashed line, simulations displayed by "+") and 4 (lower, dotted line, simulations displayed by "$\times$"). Right picture: BLX-0.5 for population size 60 (upper, dashed line, simulations displayed by "+") and 4 (lower line, simulations displayed by "$\times$"). The simulations started with a random initialization with variance 1. The results have been obtained by averaging over 100 (pop-size = 60) and 2000 (pop-size = 4), respectively, independent runs.

as decreasing population variance, depending on the choice of corresponding characteristic parameter value (η, α, or d, respectively). The deviation of the simulation from the theoretical predictions are due to cumulation of statistical errors.

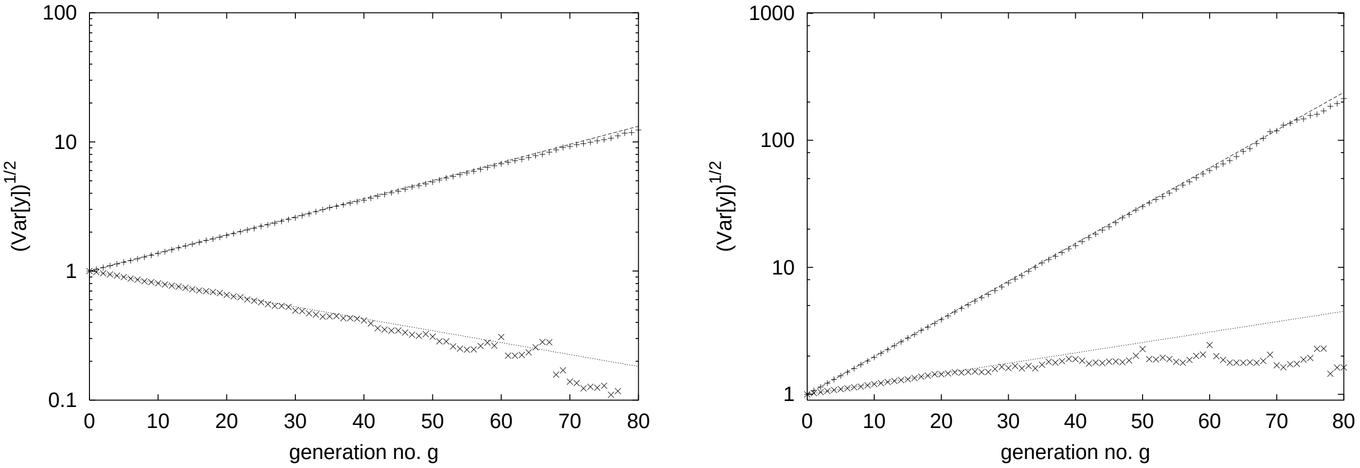

Fig. 3. Evolution of the square root of the expected population variance for the fuzzy recombination operator in the flat fitness landscape. Left picture: $d = 0.5$ for population size 60 (upper, dashed line, simulations displayed by "+") and 8 (lower, dotted line, simulations displayed by "×"). Right picture: $d = 0.7$ for population size 60 (upper, dashed line, simulations displayed by "+") and 8 (lower line, simulations displayed by "×"). The simulations started with a random initialization with variance 1. The results have been obtained by averaging over 100 (pop-size = 60) and 3000 (pop-size = 8), respectively, independent runs.

IV. Simulation Results on the Sphere Model

The above calculations on the population variance allow us to investigate whether or not GAs with different crossover operators, but the same variance properties in flat fitness landscapes, exhibit similar performance on the sphere model. Ideally we must analyze the population variance under a complete GA cycle (reproduction and variation) on the sphere model in order to compare the performances of different GAs. Such an analysis is certainly difficult, but must be pursued in the near future. From the analysis of the (μ, λ)-ES on the sphere model [38] we know, however, that the flat fitness case is included in the sphere case for the special case $\mu = \lambda$. That is, for weak selection pressure ($\mu \approx \lambda$), the flat fitness variance can serve as an upper bound on the sphere case. It is conjectured that similar properties should hold for the real-coded GAs. In any case, the above calculations provide some guidelines for GAs with different crossover implementations to be compared in any fitness landscape.

Equating the variance terms (Eq. (37), (41), and (48)) for the flat fitness case, we obtain the parameter values for the three crossover operators, given in Table I.

TABLE I

Parameter values for identical population variance growth in three crossover operators.

SBX	BLX	FR
η	α	d
2	0.662	1.095
3	0.500	0.707
5	0.419	0.433
10	0.381	0.226

In the experiments, we use a 20-dimensional sphere model. Along with the respective crossover operators, we use a binary tournament selection and no mutation. A population size of 100 is chosen. We have initialized the population at $x_i \in [-5, 5]$. For the three crossover operators, we choose the parameter setting, presented in the first row of Table I, obtained for the flat fitness landscapes. Figure 4 shows the distance of the best solution in the population from the optimum for all three crossover operators (averaged over 10 independent runs). It is clear from the plot that GAs with all three crossover operators exhibit similar performance (residual distance to optimum at generation g). Especially, they all show linear convergence order. This is also demonstrated in Figure 5, which plots the population standard deviation in the arbitrarily chosen variable x_{10}.

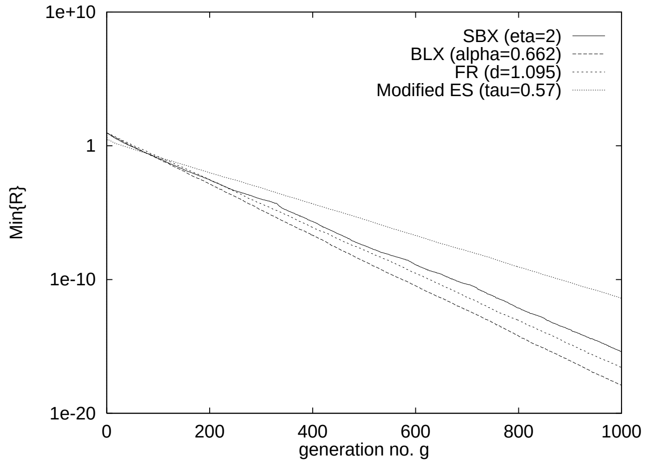

Fig. 4. Population best distance from the optimum. The curve labeled 'Modified ES' is described in Section VI.

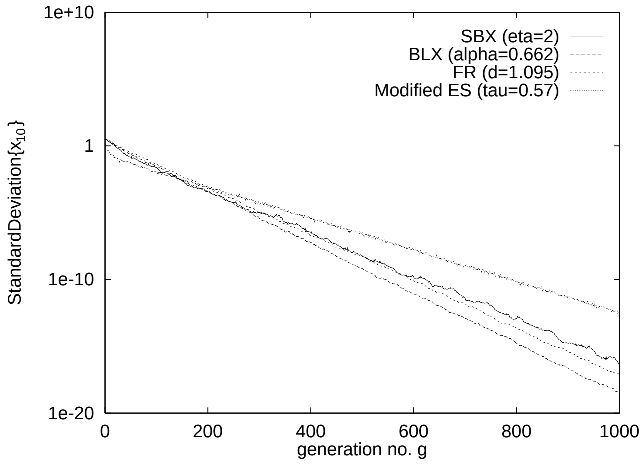

Fig. 5. Standard deviation in x_{10}. The curve labeled 'Modified ES' is described in Section VI.

V. Analysis of self-adaptive ES in flat fitness landscapes

In this section, we present a brief review of SA-ES, but remind the readers that good introductions to SA-ESs can be found in [19], [39], [40].

The basic idea of SA-ESs (and also of Meta-EPs [5], the versions of EP facilitating SA) consists in the evolution of a second set of parameters, the so-called endogenous strategy parameters, along with the decision variables (often referred to as object parameters in the ES literature). The endogenous strategy parameters control the statistical parameters by which the parental object parameters are mutated. By putting the object and strategy parameters into the individual's genome, it is expected that the survival of the fittest individuals (defined by the objective function controlled by the object parameters) also guarantees (statistically) the correct evolution of the strategy parameters. By "correct" it is meant that the mutation parameters are controlled in such a way that the ES algorithm exhibits near optimal performance (near maximal progress toward the optimum). Although the optimal update rule of endogenous strategy parameters is known for the $(1,\lambda)$-ES on the sphere model [15], it is not yet known for other ES (see [41]) or for other objective functions.

This section is organized as follows. First, a short introduction of the SA-ES is given. After that, the recombination and the mutation operators are defined. The variance calculations are performed in V-D and the comparison with real ES runs follows in Section V-E.

A. The standard SA-ES algorithm

The standard SA-ES algorithm generates λ offspring individuals from μ parent individuals, where the parents are obtained from the offspring of the preceding generation by so-called truncation selection. In this paper we will restrict our ES analysis to strategies of type $(\mu/\mu,\lambda)$. In these strategies, all μ parents are involved in the creation of a new offspring individual.

For simplicity, we consider only isotropic Gaussian mutations applied to the object parameters. That is, there is only one strategy parameter per individual, denoted by σ, determining the mutation strength by which the individual's object parameter vector $\mathbf{y}$ is mutated. The generalization to the case of N independent mutation strengths (axis parallel mutation ellipsoid) is trivial and therefore not considered here. However, the treatment of arbitrarily rotated Gaussian mutation ellipsoids (as used in Schwefel's ES version [3]) remains as a future task.

In order to generate a new offspring individual with index l for generation $g+1$, first the strategy parameter set is recombined and after that the result is mutated by multiplication with a random number Ξ_σ. The obtained new mutation strength $\sigma_l^{(g+1)}$ serves as the standard deviation for the mutation of offspring l and is incorporated in the offspring genome. The object parameter vector $\mathbf{y}_l$ is obtained by first recombining the μ parental object parameter vectors and then applying a Gaussian mutation with strength $\sigma_l^{(g+1)}$. Therefore, we have the following ES update rule expressing the ES algorithm in a condensed form

$$\forall\, l=1,\ldots,\lambda: \quad \begin{cases} \sigma_l^{(g+1)} := \mathrm{Reco}_\sigma(\sigma_{1;\lambda}^{(g)},\ldots,\sigma_{\mu;\lambda}^{(g)})\cdot\Xi_\sigma \\ \mathbf{y}_l^{(g+1)} := \mathrm{Reco}_{\mathbf{y}}(\mathbf{y}_{1;\lambda}^{(g)},\ldots,\mathbf{y}_{\mu;\lambda}^{(g)}) + \sigma_l^{(g+1)}\vec{\mathcal{N}}(0,1). \end{cases} \tag{50}$$

The parents are selected by truncation, that is, taking the μ best (as to their fitness) out of the λ offspring from the previous generation, denoted by $\sigma_{m;\lambda}^{(g)}$ and $\mathbf{y}_{m;\lambda}^{(g)}$, respectively. Here, we use the "$m;\lambda$" notation in order to refer to the

mth best individual in the offspring population of size λ.

Since we are interested in the flat fitness landscape behavior, the evolution of the strategy parameter $\sigma^{(g)}$ is decoupled from the evolution of the object parameters. Therefore, the population variance of a single object parameter entry $y_i := (\mathbf{y})_i$ after selection is given by (writing $y = y_i$)

$$\sigma^2[y]^{(g+1)} = \mathrm{E}\left[\mathrm{Var}\left\{\mathrm{Reco}_{\mathbf{y}}(\mathbf{y}^{(g)}_{1;\lambda}, \dots, \mathbf{y}^{(g)}_{\mu;\lambda})\right\}\right] + \mathrm{E}\left[\sigma^{2\,(g+1)}\right]. \tag{51}$$

This follows immediately from the second line in (50). The population variance is the sum of the (independent) population variances after recombination and mutation.

B. Recombination operators

The recombinational variance depends on the way recombination is performed. Usually, in ESs ρ parental individuals contribute to the creation of *one* new offspring individual (called recombinant, for an introduction, see e.g. [42, p. 83]). This is in contrast to standard crossover in GAs. We will restrict analysis to $\rho = \mu$ multirecombination often used in practice. We will consider intermediate and dominant recombination.

If there is (μ/μ_I)-intermediate recombination, the center of mass of all μ parents is calculated. This procedure does not produce any variance in the offspring population, since all recombinants are the same. Whereas, in dominant (μ/μ_D) recombination, a random sampling is performed for each component $(\mathbf{y})_i$ $(i = 1 \dots N)$ of the parental vectors $\mathbf{y}_{m;\lambda}$ $(m = 1 \dots \mu)$ in order to obtain the recombinant. That is, there is always one parental component (for each offspring component) that "dominates" the rest. This operator can be formalized by introducing an orthonormal basis system $\{\mathbf{e}_i\}$, $\mathbf{e}_i^{\mathrm{T}}\mathbf{e}_k = \delta_{ik}$, as

$$\text{dominant } (\mu/\mu_D)\text{-recombination:} \quad \mathrm{Reco}_{\mathbf{y}_D} := \sum_{m=1}^{N} \left(\mathbf{e}_i^{\mathrm{T}}\mathbf{y}^{(g)}_{m_i;\lambda}\right)\mathbf{e}_i =: \mathbf{y}^{(g+1)}_{r_D}, \quad m_i = \mathrm{Random}\{1, \dots, \mu\}, \tag{52}$$

where m_i is a uniform random sample from the integer set $\{1, \dots, \mu\}$ sampled anew for all $i = 1 \dots N$. Given the expected parental population variance $\sigma^2[y]^{(g)}$, the variance of the recombinants (after dominant recombination) becomes $(1 - 1/\mu)\sigma^2[y]^{(g)}$ [36]. Thus, one has

$$\mathrm{E}\left[\mathrm{Var}\left\{\mathrm{Reco}_{\mathbf{y}}(\mathbf{y}^{(g)}_{1;\lambda}, \dots, \mathbf{y}^{(g)}_{\mu;\lambda})\right\}\right] = \begin{cases} 0, & \text{if} \quad (\mu/\mu_I, \lambda)\text{-ES} \\ \left(1 - \frac{1}{\mu}\right)\sigma^2[y]^{(g)}, & \text{if} \quad (\mu/\mu_D, \lambda)\text{-ES.} \end{cases} \tag{53}$$

Other recombination variants might be considered. However, it is quite clear that for standard ES recombination techniques

$$0 \le \mathrm{E}\left[\mathrm{Var}\left\{\mathrm{Reco}_{\mathbf{y}}(\mathbf{y}^{(g)}_{1;\lambda}, \dots, \mathbf{y}^{(g)}_{\mu;\lambda})\right\}\right] \le \sigma^2[y]^{(g)} \tag{54}$$

holds. The variance can only reduce or remain unchanged by the recombination operator. The main contribution to the population variance, however, is given by the mutation operators, those standard deviation is computed in the first line of (50).

There are different ways to perform recombination on the strategy parameters. The usual way of recombination is the intermediate (μ/μ_I) one, resulting in an average recombinant individual. This is accomplished by calculating the arithmetic mean of the μ parental σ values

$$\text{intermediate } (\mu/\mu_I)\text{-recombination:} \quad \mathrm{Reco}_{\sigma_I} := \frac{1}{\mu}\sum_{m=1}^{\mu} \sigma^{(g)}_{m;\lambda} =: \sigma^{(g+1)}_{r_I}. \tag{55}$$

Alternatively one can also use the geometric mean

$$\text{geometric } (\mu/\mu_G)\text{-recombination:} \quad \mathrm{Reco}_{\sigma_G} := \sqrt[\mu]{\prod_{m=1}^{\mu} \sigma^{(g)}_{m;\lambda}} =: \sigma^{(g+1)}_{r_G}. \tag{56}$$

Usually the former (55) is the standard way of recombining the mutation strength. Interestingly, the dominant (also referred to as discrete) recombination (52) often used for object parameters has proved to be not suited for strategy parameters [39]. Therefore, it will be not considered here. Alternatively, the geometric version (56) has been suggested by Hansen [43, p. 29], which claims that geometric recombination is a "drift-free" operator in contrast to intermediate (arithmetic) (μ/μ_I)-recombination and therefore, geometric recombination should get the preference. This recommendation in [43] is based on the investigation of the expected $\ln(\sigma^{(g)})$ evolution. However, with respect to our postulates given above, the evolution of the expected values of $\sigma^{(g)}$ and $\sigma^{2(g)}$, respectively, are to be considered (cf. Postulate 2 and Eq. (51)). As shown below, our theoretical investigations cannot provide any support in favor of geometric recombination.

C. Mutation operators for the strategy parameters

While the mutation of the object parameters is done simply by adding normally distributed random numbers with zero mean and a standard deviation σ, the change of the mutation strength σ is done by multiplicative mutations. These mutations have been denoted by Ξ_σ in the first line of (50). The mutation operator can be realized in different ways, e.g.,

$$\Xi_\sigma := \exp(\zeta), \tag{57}$$

with a random variate ζ. The most prominent version uses $\mathcal{N}(0, \tau^2)$ normally distributed random numbers ζ [44]

$$\zeta_{LN} := \tau \mathcal{N}(0,1). \tag{58}$$

That is, Ξ is log-normally distributed and the mutation operator (57) is called the log-normal operator. Besides other mutation rules discussed in [15], the symmetric two-point rule is often used [2, p. 47]. The symmetric version reads

$$\zeta_{TP} := \begin{cases} +\varepsilon, & \text{if} \quad \mathrm{u}[0,1] \le \frac{1}{2} \\ -\varepsilon, & \text{if} \quad \mathrm{u}[0,1] > \frac{1}{2} \end{cases}, \quad \varepsilon > 0. \tag{59}$$

Therefore, ζ_{TP} yields e^{ε} and $\mathrm{e}^{-\varepsilon}$ with the same probability 1/2. Note, usually this rule is rewritten by the substitution $\alpha := \mathrm{e}^{\varepsilon}$, as follows:

$$\Xi_{TP} := \begin{cases} \alpha, & \text{if} \quad \mathrm{u}[0,1] \le \frac{1}{2} \\ 1/\alpha, & \text{if} \quad \mathrm{u}[0,1] > \frac{1}{2}. \end{cases} \tag{60}$$

D. Calculation of the population variance

The recursive equation of the population variance can be derived from (51). Since the first term in (51) is given by (53), the calculation of $\mathrm{E}[\sigma^{2\,(g+1)}]$ remains to be done. It is important to realize that due to the selection neutrality in flat fitness landscapes, the first line in (50) is decoupled from the second line. That is, the σ evolution is fully determined by the first line in (50). It suffices to calculate the expected variance produced by the recombination operator in the first line of (50). This will be done in the next two points; furthermore we will see that the evolution of the expected σ_r (r stands for the recombinant) and σ_r^2 is different, that is, $\overline{\sigma_r^{2\,(g)}} \neq \overline{\sigma_r^{(g)}}^2$. After that, the calculation of the expected population variance $\sigma^2[y]^{(g)}$ will be performed.

D.1 Determining $\overline{\sigma_r}$

In order to calculate the expected value of the strategy parameter σ after recombination in the first line of (50), denoted by $\overline{\sigma_r}$, we have to take (57) into account. The $\sigma_{m;\lambda}^{(g)}$ are generated according to the first line in (50), (55), and (56), respectively, that is, the σ value of the lth offspring is obtained as

$$\sigma_l^{(g)} := \sigma_r^{(g)} \mathrm{e}^{\zeta_l^{(g)}}. \tag{61}$$

Since selection does not prefer any instantiation of (61), the recombination result using μ individuals becomes

$$\text{intermediate } (\mu/\mu_I)\text{-recombination:} \quad \sigma_{r_I}^{(g+1)} := \frac{1}{\mu} \sum_{m=1}^{\mu} \sigma_{r_I}^{(g)} \mathrm{e}^{\zeta_m^{(g)}} \tag{62}$$

and

$$\text{geometric } (\mu/\mu_G)\text{-recombination:} \quad \sigma_{r_G}^{(g+1)} := \sqrt[\mu]{\prod_{m=1}^{\mu} \sigma_{r_G}^{(g)} \mathrm{e}^{\zeta_m^{(g)}}}. \tag{63}$$

Determining the expected value taking the statistical independence into account, one gets for the (μ/μ_I) version

$$\overline{\sigma_{r_I}^{(g+1)}} = \overline{\sigma_{r_I}^{(g)}} \frac{1}{\mu} \sum_{m=1}^{\mu} \overline{\mathrm{e}^{\zeta_m^{(g)}}} = \overline{\sigma_{r_I}^{(g)}}\ \overline{\mathrm{e}^{\zeta}} \tag{64}$$

and for the geometric recombination

$$\overline{\sigma_{r_G}^{(g+1)}} = \overline{\sigma_{r_G}^{(g)}}\ \mathrm{E}\left[\prod_{m=1}^{\mu} \exp\left(\frac{\zeta_m^{(g)}}{\mu}\right)\right] = \overline{\sigma_{r_G}^{(g)}} \left(\overline{\mathrm{e}^{\zeta/\mu}}\right)^{\mu}. \tag{65}$$

The expectations $\overline{\mathrm{e}^{\zeta}}$ and $\overline{\mathrm{e}^{\zeta/\mu}}$ can be calculated easily for the log-normal distribution and the two-point case (we shall tabulate these parameters later). Since the expectation of the log-normal variate $\overline{\exp(\gamma\mathcal{N}(0,1))} = \exp(\gamma^2/2)$, we have

$$\text{log-normal mutation:} \quad \overline{\mathrm{e}^{\zeta_{LN}}} = \mathrm{e}^{\tau^2/2} \quad \text{and} \quad \overline{\mathrm{e}^{\zeta_{LN}/\mu}} = \mathrm{e}^{\tau^2/2\mu^2}. \tag{66}$$

For the symmetric two-point rule (59) we find immediately

$$\text{two-point mutation:} \quad \overline{\mathrm{e}^{\zeta_{TP}}} = \frac{1}{2}(\mathrm{e}^{\varepsilon} + \mathrm{e}^{-\varepsilon}) = \cosh(\varepsilon) \quad \text{and} \quad \overline{\mathrm{e}^{\zeta_{TP}/\mu}} = \cosh(\varepsilon/\mu). \tag{67}$$

From Eq. (64) and (65) one can derive the evolution equations, i.e., the g-dynamics of $\overline{\sigma_r}$. By iteration, one obtains the results displayed in Table II. As long as $\tau > 0$ and $\varepsilon > 0$, respectively, the expected value of σ increases exponentially

TABLE II

Dynamics of the expected value of σ after recombination. $\overline{\sigma_r^{(0)}}$ is the initial value of mutation strength.

	(μ/μ_I)-recombination	(μ/μ_G)-recombination
log-normal rule	$\overline{\sigma_r^{(g)}} = \overline{\sigma_r^{(0)}} \exp\left(\frac{\tau^2}{2}g\right)$	$\overline{\sigma_r^{(g)}} = \overline{\sigma_r^{(0)}} \exp\left(\frac{\tau^2}{2\mu}g\right)$
two-point rule	$\overline{\sigma_r^{(g)}} = \overline{\sigma_r^{(0)}}[\cosh(\varepsilon)]^g$	$\overline{\sigma_r^{(g)}} = \overline{\sigma_r^{(0)}}\left[\cosh\left(\frac{\varepsilon}{\mu}\right)\right]^{\mu g}$

with g. The results on those ES versions using geometric recombination are astonishing and seem to be in contradiction to the assertions in [43, p. 29] that geometric recombination yields a "drift-free" σ evolution process. The solution to this contradiction lies in the way how drift is measured quantitatively. In [43] the expected value evolution of a nonlinearly transformed σ process has been investigated, i.e., the evolution of $\ln(\sigma^{(g)})$. However, when considering the *real* expected σ evolution, we see that both intermediate and geometric recombination do not introduce any drift. The drift comes from the mutation operators. Both the log-normal and the symmetric two-point rule are biased toward an increase in σ, because both the expected values (66) and (67) are greater than one. Looking at (64), intermediate recombination *conserves* the expected effect of the mutation operator which is $\overline{\mathrm{e}^{\zeta}}$. In contrast to this conserving property of intermediate recombination, Equation (65) reveals that the application of the geometric recombination changes the expected value of the σ values produced by the mutation operator. Actually, it *reduces* the expected value, because $\overline{\mathrm{e}^{\zeta}} > \left(\overline{\mathrm{e}^{\zeta/\mu}}\right)^{\mu}$ holds for both the log-normal mutation rule and the two-point rule. This effect is also observed in the variance evolution $\sigma^2[y]^{(g)}$ of the parental object parameters (see Section V-D.3 below and Figure 6).

As we have already pointed out and formalized by Postulate 2, an increase of the expected σ is desirable for the functioning of the EA in flat fitness landscapes. However, one can also construct σ mutation operators with $\overline{\mathrm{e}^{\zeta}} = 1$. This would guarantee that $\overline{\sigma_r^{(g)}}$ remains constant. Interestingly, even for that case, the expected population variance $\sigma^2[y]^{(g)}$ can increase over the time. The reason for this astonishing behavior becomes clear if one takes into account that $\overline{\sigma_r^{2\,(g)}} = \overline{\sigma_r^{(g)}}^2 + \mathrm{D}^2[\sigma_r]^{(g)}$: As long as σ_r fluctuates with a non-zero variance D^2, we have $\overline{\sigma_r^{2\,(g)}} \neq \overline{\sigma_r^{(g)}}^2$. Actually, $\mathrm{D}^2[\sigma_r]^{(g)}$ has influence on the expected population variance $\sigma^2[y]^{(g)}$. That is why we have to consider $\overline{\sigma_r^{2\,(g)}}$ next.

D.2 Determining $\overline{\sigma_r^2}$

The calculation of $\overline{\sigma_r^2}$ starts from (62) and (63), respectively. Taking the square of (62), we obtain for the intermediate recombination

$$\begin{aligned} \sigma_{r_I}^{2\,(g+1)} &= \frac{1}{\mu^2}\sum_{k=1}^{\mu}\sum_{j=1}^{\mu}\sigma_{r_I}^{2\,(g)}\mathrm{e}^{\zeta_k^{(g)}}\mathrm{e}^{\zeta_j^{(g)}} \\ &= \sigma_{r_I}^{2\,(g)}\frac{1}{\mu^2}\sum_{m=1}^{\mu}\mathrm{e}^{2\zeta_m^{(g)}} + \sigma_{r_I}^{2\,(g)}\frac{1}{\mu^2}\sum_{k=1}^{\mu}\sum_{j\neq k}\mathrm{e}^{\zeta_k^{(g)}}\mathrm{e}^{\zeta_j^{(g)}}. \end{aligned} \tag{68}$$

Now the expected value can be calculated taking the statistical independence of the identically distributed ζ_k, ζ_j, and ζ_m into account

$$\begin{aligned} \overline{\sigma_{r_I}^{2\,(g+1)}} &= \overline{\sigma_{r_I}^{2\,(g)}}\left[\frac{1}{\mu^2}\sum_{m=1}^{\mu}\overline{\mathrm{e}^{2\zeta}} + \frac{1}{\mu^2}\sum_{k=1}^{\mu}\sum_{j\neq k}\overline{\mathrm{e}^{\zeta}}^2\right] \\ &= \overline{\sigma_{r_I}^{2\,(g)}}\left[\frac{1}{\mu}\overline{\mathrm{e}^{2\zeta}} + \frac{\mu^2-\mu}{\mu^2}\overline{\mathrm{e}^{\zeta}}^2\right]. \end{aligned} \tag{69}$$

Therefore, we get

$$\overline{\sigma_{r_I}^{2\,(g+1)}} = \overline{\sigma_{r_I}^{2\,(g)}} \left(\overline{\mathrm{e}^{\zeta}}^2 + \frac{1}{\mu} \left[\overline{\mathrm{e}^{2\zeta}} - \overline{\mathrm{e}^{\zeta}}^2 \right] \right) \tag{70}$$

or alternatively

$$\overline{\sigma_{r_I}^{2\,(g+1)}} = \overline{\sigma_{r_I}^{2\,(g)}} \left(\overline{\mathrm{e}^{\zeta}}^2 + \frac{1}{\mu} \mathrm{D}^2[\mathrm{e}^{\zeta}] \right) \quad \text{with} \quad \mathrm{D}^2[\mathrm{e}^{\zeta}] = \overline{\mathrm{e}^{2\zeta}} - \overline{\mathrm{e}^{\zeta}}^2, \tag{71}$$

where $\mathrm{D}^2[\mathrm{e}^{\zeta}]$ is the variance of the mutation operator. Finally, we obtain the equation of the σ_r^2 dynamics

$$\overline{\sigma_{r_I}^{2\,(g)}} = \overline{\sigma_{r_I}^{2\,(0)}} \left(\overline{\mathrm{e}^{\zeta}}^2 + \frac{1}{\mu} \mathrm{D}^2[\mathrm{e}^{\zeta}] \right)^g . \tag{72}$$

One can clearly see $\overline{\sigma_{r_I}^{2\,(g)}} = \overline{\sigma_{r_I}^{(g)}}^2$ holds only for $\mu \to \infty$ because $\mathrm{D}^2[\mathrm{e}^{\zeta}] > 0$. As pointed out in the previous subsection, even when $\overline{\mathrm{e}^{\zeta}} = 1$ is guaranteed (leading to $\overline{\sigma_{r_I}^{(g)}} = \overline{\sigma_{r_I}^{(0)}}$, see Table II), $\overline{\sigma_{r_I}^{2\,(g)}}$ will be an monotonically increasing function.

As to the calculation of $\overline{\sigma_{r_G}^{2\,(g+1)}}$ (geometric recombination) we start from (63)

$$\sigma_{r_G}^{2\,(g+1)} = \left(\prod_{m=1}^{\mu} \sigma_{r_G}^{(g)} \mathrm{e}^{\zeta_m^{(g)}} \right)^{\frac{2}{\mu}} = \sigma_{r_G}^{2\,(g)} \prod_{m=1}^{\mu} \exp\left(\frac{2\zeta_m^{(g)}}{\mu} \right). \tag{73}$$

The expected value reads

$$\overline{\sigma_{r_G}^{2\,(g+1)}} = \overline{\sigma_{r_G}^{2\,(g)}} \prod_{m=1}^{\mu} \overline{\mathrm{e}^{2\zeta/\mu}} = \overline{\sigma_{r_G}^{2\,(g)}} \left(\overline{\mathrm{e}^{2\zeta/\mu}} \right)^{\mu} . \tag{74}$$

We have taken the statistical independence of the identically distributed ζ_m into account. One can rewrite (74) by introducing the variance of $\mathrm{e}^{\zeta/\mu}$

$$\mathrm{D}^2[\mathrm{e}^{\zeta/\mu}] = \overline{\mathrm{e}^{2\zeta/\mu}} - \overline{\mathrm{e}^{\zeta/\mu}}^2 \tag{75}$$

leading to

$$\overline{\sigma_{r_G}^{2\,(g+1)}} = \overline{\sigma_{r_G}^{2\,(g)}} \left(\overline{\mathrm{e}^{\zeta/\mu}}^2 + \mathrm{D}^2[\mathrm{e}^{\zeta/\mu}] \right)^{\mu} . \tag{76}$$

The evolution equation of $\overline{\sigma_{r_G}^{2\,(g)}}$ then becomes

$$\overline{\sigma_{r_G}^{2\,(g)}} = \overline{\sigma_{r_G}^{2\,(0)}} \left(\overline{\mathrm{e}^{\zeta/\mu}}^2 + \mathrm{D}^2[\mathrm{e}^{\zeta/\mu}] \right)^{\mu g} . \tag{77}$$

One can see that – similar to the intermediate recombination case – a vanishing drift, that is, $\overline{\mathrm{e}^{\zeta/\mu}} = 1$ (leading to $\overline{\sigma_{r_G}^{(g)}} = \overline{\sigma_{r_G}^{(0)}}$, see Table II), does not imply $\overline{\sigma_{r_G}^{2\,(g)}} = \overline{\sigma_{r_G}^{2\,(0)}}$. The reason lies in the stochasticity of the σ mutation operator.

D.3 Determination of the expected population variance of the object parameters

The calculation of $\sigma^2[y]^{(g+1)}$ starts from Eq. (51). The first term of (51) is already known from Eq. (53). Therefore, it remains to calculate $\mathrm{E}[\sigma^{2\,(g+1)}]$. Since $\sigma^{(g+1)}$ is generated in the first line of Eq. (50) and Ξ_σ is independent of the σ recombination, we find with (57) and (55) or (56)

$$\mathrm{E}[\sigma^{2\,(g+1)}] = \overline{\mathrm{e}^{2\zeta}}\; \overline{\sigma_r^{2\,(g)}}. \tag{78}$$

Consider the intermediate object parameter recombination case first. With (51), (53), and (78) we have (writing g instead of $g+1$)

$$\sigma^2[y]^{(g)} = 0 + \overline{\mathrm{e}^{2\zeta}}\; \overline{\sigma_r^{2\,(g-1)}} \tag{79}$$

and therefore the evolution dynamics becomes with (72)

$$\underline{(\mu/\mu_I)\text{-recombination:}} \qquad \sigma^2[y]^{(g)} = \overline{\sigma_r^{2\,(0)}}\; \overline{\mathrm{e}^{2\zeta}} \left(\overline{\mathrm{e}^{\zeta}}^2 + \frac{1}{\mu} \mathrm{D}^2[\mathrm{e}^{\zeta}] \right)^{(g-1)} . \tag{80}$$

for the "II" recombination case.[8] For the "IG" recombination version we obtain with (77)

$$\underline{(\mu/\mu_{IG})\text{-recombination:}} \qquad \sigma^2[y]^{(g)} = \overline{\sigma_r^{2\,(0)}}\ \overline{e^{2\zeta}} \left(\overline{e^{\zeta/\mu}}^2 + \mathrm{D}^2[e^{\zeta/\mu}]\right)^{\mu(g-1)}. \tag{81}$$

Both the (μ/μ_{II}) and the (μ/μ_{IG}) versions can exhibit exponential variance increase as suggested by Postulate 2 (see Section II-B). As a necessary and sufficient condition, the value of the expression in the large parenthesis of (80) and (81), respectively, must be greater than 1. This condition does even hold when the σ mutation operator is "drift free", that is, $\overline{e^{\zeta}} = 1$ and $\overline{e^{\zeta/\mu}} = 1$, respectively.

The cases with dominant object parameter recombination need further considerations. Instead of (79) we now get from (53), (51), and (78)

$$\sigma^2[y]^{(g)} = \frac{\mu-1}{\mu}\sigma^2[y]^{(g-1)} + \overline{e^{2\zeta}}\ \overline{\sigma_r^{2\,(g-1)}}. \tag{82}$$

Considering (72) and (77), respectively, (82) is of the form

$$a^{(g)} = ba^{(g-1)} + \left(\frac{c}{d}\right) d^g, \tag{83}$$

with

$$a := \sigma^2[y], \qquad b := \frac{\mu-1}{\mu}, \qquad c := \overline{\sigma_r^{2\,(0)}}\ \overline{e^{2\zeta}} \tag{84}$$

and

$$d_I := \overline{e^{\zeta}}^2 + \frac{1}{\mu}\mathrm{D}^2[e^{\zeta}] \tag{85}$$

for the intermediate σ recombination and

$$d_G := \left(\overline{e^{\zeta/\mu}}^2 + \mathrm{D}^2[e^{\zeta/\mu}]\right)^{\mu} \tag{86}$$

for the geometric σ recombination.

By recursion, Eq. (83) can be expressed as a sum of a geometric progression. It reads

$$a^{(g)} = a^{(0)}b^g + \frac{c}{d}\sum_{k=0}^{g-1} b^k d^{g-k} = a^{(0)}b^g + \frac{cd^g}{d}\sum_{k=0}^{g-1}(b/d)^k. \tag{87}$$

Thus, the closed-form solution is

$$a^{(g)} = a^{(0)}b^g + \frac{c}{d}d^g\frac{1-(b/d)^g}{1-(b/d)} = a^{(0)}b^g + c\frac{d^g - b^g}{d-b}. \tag{88}$$

Finally, we get for (88) with (84) and (85) the population variance dynamics of the "DI" case

$$\underline{(\mu/\mu_{DI})\text{-recombination:}}$$

$$\sigma^2[y]^{(g)} = \sigma^2[y]^{(0)}\left(1-\frac{1}{\mu}\right)^g + \overline{\sigma_r^{2\,(0)}}\ \overline{e^{2\zeta}}\ \frac{\left(\overline{e^{\zeta}}^2 + \frac{1}{\mu}\mathrm{D}^2[e^{\zeta}]\right)^g - \left(1-\frac{1}{\mu}\right)^g}{\left(\overline{e^{\zeta}}^2 + \frac{1}{\mu}\mathrm{D}^2[e^{\zeta}]\right) - \left(1-\frac{1}{\mu}\right)}. \tag{89}$$

For the "DG" recombination version, one respectively obtains with (86)

$$\underline{(\mu/\mu_{DG})\text{-recombination:}}$$

$$\sigma^2[y]^{(g)} = \sigma^2[y]^{(0)}\left(1-\frac{1}{\mu}\right)^g + \overline{\sigma_r^{2\,(0)}}\ \overline{e^{2\zeta}}\ \frac{\left(\overline{e^{\zeta/\mu}}^2 + \mathrm{D}^2[e^{\zeta/\mu}]\right)^{\mu g} - \left(1-\frac{1}{\mu}\right)^g}{\left(\overline{e^{\zeta/\mu}}^2 + \mathrm{D}^2[e^{\zeta/\mu}]\right)^{\mu} - \left(1-\frac{1}{\mu}\right)}. \tag{90}$$

In order to use these formulae, the statistical parameters of the mutation operators are needed. For convenience these expressions are collected in Table III.

[8] In order to simplify references to different variants of recombination, we will use the notation "OS" where "O" stands for the object parameter recombination types $I|D$ and "S" stands for the strategy parameter recombination types $I|D|G$, that is, "II" means intermediate object parameter recombination and intermediate strategy parameter recombination.

TABLE III

STATISTICAL PARAMETERS OF THE STANDARD σ MUTATION OPERATORS.

	log-normal rule	symmetric two-point rule
$\overline{\mathrm{e}^{\zeta}}$	$\mathrm{e}^{\tau^2/2}$	$\cosh(\varepsilon)$
$\overline{\mathrm{e}^{2\zeta}}$	$\mathrm{e}^{2\tau^2}$	$\cosh(2\varepsilon)$
$\overline{\mathrm{e}^{\zeta/\mu}}$	$\mathrm{e}^{\tau^2/2\mu^2}$	$\cosh(\varepsilon/\mu)$
$\overline{\mathrm{e}^{2\zeta/\mu}}$	$\mathrm{e}^{2\tau^2/\mu^2}$	$\cosh(2\varepsilon/\mu)$
$\mathrm{D}^2[\mathrm{e}^{\zeta}]$	$\mathrm{e}^{\tau^2}(\mathrm{e}^{\tau^2}-1)$	$\cosh(2\varepsilon)-(\cosh(\varepsilon))^2$
$\mathrm{D}^2[\mathrm{e}^{\zeta/\mu}]$	$\mathrm{e}^{\tau^2/\mu^2}(\mathrm{e}^{\tau^2/\mu^2}-1)$	$\cosh(2\varepsilon/\mu)-(\cosh(\varepsilon/\mu))^2$

When comparing the variance evolution of the "II" and "IG" type (Eq. (80), (81)), with that of "DI" and "DG" (Eq. (89), (90)), one sees that the dominant object parameter recombination adds decay terms to the variance evolution. That is, the initial parental population variance fades away exponentially. However, unlike the intermediate (μ/μ) object parameter recombination, there is still a certain "memory" of the older parental states. It is an open question whether this property has some performance benefit in nonlinear fitness landscapes.

E. Simulation examples and discussion

Figure 6 compares the variance evolution formulae (80), (81), (89), and (90) with *real* ES runs on the flat fitness landscape. A $(6/6, 60)$-ES has been tested using the four standard combinations of the recombination operators (52), (55, 56) and the log-normal mutation operator defined by (57, 58). One observes a good fit between the predictions of the theory and the experiments.

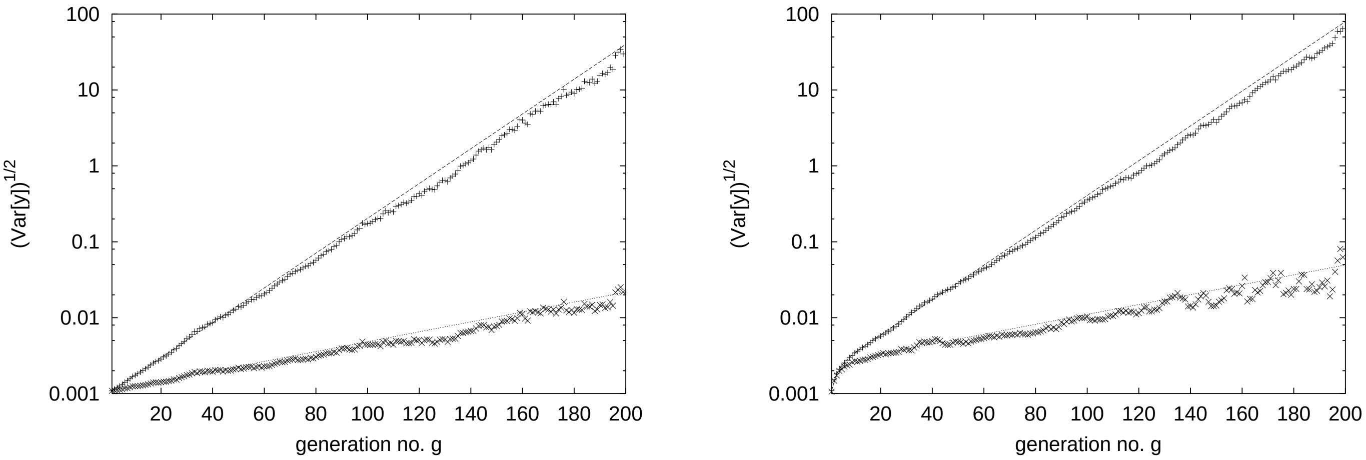

Fig. 6. Evolution of the square root of the expected population variance in a $(6/6, 60)$-ES with log-normally mutated strategy parameters applied to a flat fitness landscape. Left picture: intermediate object parameter recombination. The upper curve (dashed line) displays the dynamics, Eq. (37), for the intermediate recombination on the strategy parameters, the "+" represents the corresponding simulation results. The lower curve (dotted line), Eq. (81), displays the geometric recombination, the simulations are depicted by "×". Right picture: dominant object parameter recombination. The upper curve (dashed line) displays the dynamics, Eq. (89), for the intermediate recombination on the strategy parameters, the "+" represents the corresponding simulation results. The lower curve (dotted line), Eq. (90), displays the geometric recombination, the simulations are depicted by "×". The simulation results have been obtained by averaging over 500 independent runs. In all experiments $\sigma^2[y]^{(0)} = 0$ and $\sigma_r^{(0)} = 0.001$ has been chosen. A learning parameter $\tau = 0.3$ has been used.

Even the geometric recombination variants exhibit an exponential increase in population variance. However, the rate of the variance increase with geometric recombination is slower than that with intermediate recombination in strategy parameters. For example, considering (89) and (90) one can estimate that the geometric recombination has at most μ times slower rate than intermediate recombination. Choosing μ sufficiently large, one could infer from simulation experiments that the ES versions with geometric recombination are indeed "drift free," as has been postulated in [43, p. 29] by considering the expected value evolution of $\ln(\sigma^{(g)})$. However, by the analysis presented here, it becomes clear that the variance-increasing behavior (population variance as well as the "drift" of the expected value of $\sigma^{(g)}$) comes from the σ-mutation operators, those standard versions (57, 58) and (60) are already biased toward an σ^2 increase. From this point of view, geometric recombination is more "biased" than intermediate recombination because of its strong μ-dependent damping behavior.

VI. Real-coded GAs versus Self-adaptive ESs

With the above calculations of the expected population variance for real-parameter variation operators and SA-ES operators on flat fitness landscapes, it becomes easier to explain why similar performances with them have been observed earlier [22]. Although the calculations will be different for other fitness landscapes, we conjecture that as long as the expected population variances of a SA-ES and a real-coded GA are similar, their performance will be similar. For flat fitness landscapes, this allows us to equate the growth of expected population variances of two operators and to find corresponding characteristic parameter values.

As an example, consider the SA-ES with log-normal σ-mutations and (μ/μ_{II})-recombination and a real-coded GAs with fuzzy recombination (FR) defined by (10, 17, 16). By equating their growth ratio of the expected population variances (Eq. (80 and (48)), one finds a relationship between the characteristic parameter values τ and d

$$\frac{\sigma^2[y]^{(g)}}{\sigma^2[y]^{(g-1)}} = \mathrm{e}^{\tau^2} + \frac{1}{\mu_{ES}}\mathrm{e}^{\tau^2}\left(\mathrm{e}^{\tau^2} - 1\right) = 1 + \frac{d^2}{3} - \frac{1}{\mu_{GA}}. \tag{91}$$

For large population sizes, the following relationship is obtained

$$\tau \approx \sqrt{\ln\left(1 + \frac{d^2}{3}\right)}. \tag{92}$$

Using the above relationship between the SA-ES and the real-coded GA with the FR operator, we would expect to have similar dynamics of the expected population variance on a flat fitness landscape. However, when applied to other fitness landscapes, the expected population variance may not be the same. One reason for the expected differences is due to the different reproduction/selection operators traditionally used in real-coded GAs and SA-ESs. This makes it difficult to compare them in landscapes other than flat fitness landscapes. As mentioned earlier, the reproduction operator has an effect of reducing the population variance in general. Thus, if reproduction operators used in SA-EAs do not have similar properties in reducing the population variance, the comparison of variation operators with identical variance growth properties would be meaningless.

In order to support this explanation, the behavior on the sphere model is reconsidered. We run a SA-ES with $\tau = 0.57$ (calculated using Eq. (91) with $d = 1.095$ and $\mu_{GA} = 100$). In order to have a similar selection pressure as that of real-coded GAs, we use binary tournament selection, instead of the truncation selection usually applied in ESs. Thus, we have $\mu_{ES} = \rho = 50$, but use an offspring population size $\lambda = 100$. Simulation results with (μ/μ_{II})-recombination operator are shown in Figures 4 and 5 (average of 10 runs). The performance of SA-ES is not very different from that of the real-coded GAs. The difference in the performance arises due to the use of the estimate (92) for flat fitness landscapes in the sphere model simulations: When choosing $\tau \approx 0.4$, obtained by experiments, the SA-ES shows very similar performance compared with the real-coded GAs.

It is interesting to note that the expected population variance in flat fitness landscapes *never* reduces with SA-ESs. This is due to the way self-adaptive mutation operators are constructed. On the other hand, the expected population variance in real-coded GAs can increase or decrease depending on the population size. If the population size is small, a decrease in expected population variance is likely. As to the variance evolution in flat fitness landscapes, GAs seem to be more sensible than ESs. Therefore, there is a greater need for choosing an appropriate population size in GAs than in SA-ESs.

The calculations of expected population variances and their relevance to the performance of the algorithms demonstrated here mark a first step towards finding deeper connections among different self-adaptive evolutionary algorithms.

VII. On the current state of the analysis of EAs in real-valued linear fitness landscapes

Postulate 3 in Section II-B concerns the behavior of SA-EAs in linear fitness landscapes. Up until now, the analysis of self-adaptive properties in EAs on linear fitness landscapes has not been done. As an exception, the analysis of the $(1, \lambda)$-σSA-ES on the sphere model was presented in [15]. Although not explicitly considered there, the linear fitness case is implicitly included in that analysis and will be elaborated here. This can be accomplished easily because of some nice scaling properties inherent in the sphere model theory:

Provided that the mutation strength σ is fixed, the normalized mutation strength σ^*, defined by

$$\sigma^* := \sigma\,\frac{N}{R}, \tag{93}$$

with R as the actual parental distance to the optimum and N the object parameter space dimension, goes to zero for $R \to \infty$. This means that with increasing R the spherical landscape appears for the mutations with strength σ more and more plane. In the limit one gets the behavior of a hyperplane (linear fitness landscape), provided that $\sigma^* \to 0$. Therefore, by considering the limit $\sigma^* \to 0$ for $0 < \sigma < \infty$, the result of the sphere model theory can be used to describe the behavior of the ES in linear fitness landscapes.

A. Determining the expected σ evolution

The central quantity of the σSA theory is the so-called *self-adaptation response function* $\psi(\sigma^{*(g)})$. It is defined for the $(1,\lambda)$-σSA-ES as

$$\psi(\sigma^{*(g)}) := \mathrm{E}\left[\frac{\sigma^{(g+1)} - \sigma^{(g)}}{\sigma^{(g)}}\right]. \tag{94}$$

It measures the relative change of the strategy parameter σ given a parental state σ and σ^*, respectively. Figure 7 shows typical plots of $\psi(\sigma^{*(g)})$ functions (reproduced from [15, p.326]). From this curves only the values at $\sigma^* = 0$ are

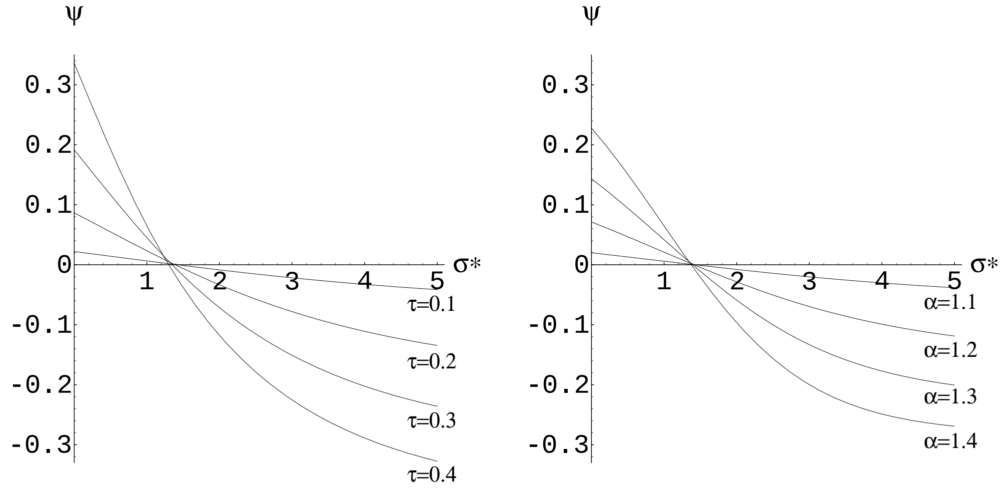

Fig. 7. The self-adaptation response $\psi(\sigma^*)$ for the $(1,10)$-σSA-ES with $N = 30$. Figure left: log-normal rule (57/58) used. Figure right: symmetric two-point rule (60) used.

of interest for linear fitness landscapes. Unfortunately, in almost all cases there is no analytical expression for $\psi(\sigma^*)$. The only exception is for the $(1,2)$-σSA-ES with two-point mutation operator (60). One finds in this special case [45, p. 27], [16, Eq. (7.89)]

$$\text{symmetrical two-point rule:} \qquad \psi_{1,2}(0) = \frac{1}{2}\frac{(\alpha-1)^2}{\alpha} > 0. \tag{95}$$

For small learning parameters τ and $\alpha - 1$, respectively, it is possible to derive approximate $\psi(\sigma^*)$ expressions [15, p. 328]

$$\text{log-normal rule:} \qquad \psi_{1,\lambda}(0) = \tau^2\left(d^{(2)}_{1,\lambda} - \frac{1}{2}\right) + \ldots > 0, \tag{96}$$

$$\text{symmetrical two-point rule:} \qquad \psi_{1,\lambda}(0) = (2-\alpha)(\alpha-1)^2\left(d^{(2)}_{1,\lambda} - \frac{1}{2}\right) + \ldots > 0, \tag{97}$$

with $d^{(2)}_{1,\lambda}$ the second order progress coefficient (see e.g. [15, p. 325]).

In order to calculate the σ dynamics on flat fitness landscapes, Eq. (94) is used to express the expected σ at generation $g+1$ by that of g. One immediately finds with $\sigma^* = 0$

$$\overline{\sigma^{(g+1)}} = (1 + \psi(0))\,\overline{\sigma^{(g)}}. \tag{98}$$

Therefore, the dynamics of the expected σ evolution are

$$\overline{\sigma^{(g)}} = \sigma^{(0)}(1+\psi(0))^g. \tag{99}$$

Since $\psi(0) > 0$ for the σ mutation rules considered here, one observes an exponentially increase of the expected mutation strength. As a consequence, the population variance of the object parameters must increase as well. According to Postulate 3, this is regarded as a desired behavior. In [32, p. 11] it was claimed that a $(1,2)$-σSA-ES does not exhibit such a behavior. Since [32] uses the symmetric two-point rule, the exact result (95) asserts that the $(1,2)$-σSA-ES does necessarily increase the expected σ according to (98) as long as $\alpha > 1$. The error in [32] comes again from consideration of improvement probabilities. It could be avoided by looking at the expected σ changes, as has been done here.

B. Determining the expected distance traveled

Unlike the cases considered in the previous sections, the calculation of the parental variance does not make sense for the case of $(1, \lambda)$ strategies considered here (note, there is only one parent). One could consider the offspring variance instead. Alternatively one can investigate the expected distance traveled towards the optimum.

Assuming small learning parameters, the expected distance $\mathrm{E}[\Delta x]$ traveled from generation g to $g+1$ is roughly

$$\mathrm{E}[\Delta x]^{(g)} = \overline{x^{(g)}} - \overline{x^{(g-1)}} = c_{1,\lambda}\overline{\sigma^{(g)}} + \ldots, \tag{100}$$

with the progress coefficient[9] $c_{1,\lambda}$ (for a table of values, see e.g. [15, p. 325]). Formula (100) is exact for small learning parameters (that is, $\tau \to 0$, $\alpha \to 1$). However, it may be used as a first approximation here. From (100) one obtains with (99)

$$\mathrm{E}[\Delta x]^{(g)} = c_{1,\lambda}\sigma^{(0)}(1+\psi(0))^g + \ldots \tag{101}$$

and therefore, by assuming an initial $x^{(0)} = 0$,

$$\overline{x^{(g)}} = \overline{x^{(g-1)}} + c_{1,\lambda}\sigma^{(0)}(1+\psi(0))^g + \ldots. \tag{102}$$

By recurrence one obtains a sum of geometric progression

$$\begin{aligned}
\overline{x^{(g)}} &= c_{1,\lambda}\sigma^{(0)} \sum_{k=1}^{g} (1+\psi(0))^k + \ldots \\
&= c_{1,\lambda}\sigma^{(0)} \left(-1 + (1+\psi(0))^g + \sum_{k=0}^{g-1} (1+\psi(0))^k \right) + \ldots \\
&= c_{1,\lambda}\sigma^{(0)} \left(-1 + (1+\psi(0))^g + \frac{1-(1+\psi(0))^g}{1-(1+\psi(0))} \right) + \ldots.
\end{aligned} \tag{103}$$

After rewriting (103) we end up with

$$\overline{x^{(g)}} = c_{1,\lambda}\sigma^{(0)} \left(1 + \frac{1}{\psi(0)} \right) \left[(1+\psi(0))^g - 1 \right] + \ldots. \tag{104}$$

For sufficiently large g the self-adaptation algorithm exhibits an exponential increase of the distance traveled in optimum direction. As an example, the $(1, 10)$-σSA-ES is simulated and the graph of (99) and (104) are plotted in Figure 8 (the progress coefficient $c_{1,\lambda}$ is $c_{1,10} \approx 1.5388$). One observes a good fit over many orders of magnitude.

VIII. Conclusions

In this paper, we have postulated some properties for population mean and variance for self-adaptive evolutionary algorithms (SA-EAs) in real-valued (unrestricted) search spaces. With the task of exploration and exploitation in mind, we argued that the population mean should not be changed by the variance operator. Although the manner in which the population variance should be changed by a variance operator must depend on the fitness landscape and the associated reproduction operator, we argued that it may be better strategy, in general, to have a tendency toward increasing the population variance by a variation operator (that is, by recombination and mutation).

We have analyzed the population mean and variance of three crossover operators – blend crossover (BLX), simulated binary crossover (SBX), and fuzzy recombination (FR) – commonly-used in real-coded GAs. Theoretical predictions of the increasing and decreasing nature of population variance are validated with experimental results on a flat fitness landscape. By choosing an appropriate characteristic parameter for each crossover operator, the growth of the population variance can be matched. We have shown that GAs with the above three crossover operators exhibit similar self-adaptive features on a sphere model.

For the first time, the population variance has been calculated for self-adaptive evolution strategies in flat fitness landscapes. The theoretical results have also been supported by performing real ES runs. An estimate of the expected distance traveled on a linear fitness function has also been calculated.

Using the estimates for the expected population growth on a flat fitness landscape, the performance of self-adaptive ESs and real-coded GAs have been compared using a sphere model. Although the results have shown similar behavior, a closer similarity could be achieved if the expected population variance estimates could be obtained for both reproduction and variation operators. From the experiences concerning the analysis of (μ, λ)-ES using the sphere model [38], however, one can infer that such an analysis might appear as a difficult task.

[9]Eq. (100) holds exactly for isotropic normally distributed mutations (that is, for $\tau \to 0$ or $\alpha \to 1$). In this case, the problem reduces to a one-dimensional: Starting from a point $x^{(g-1)}$, λ offspring are generated by $\sigma\mathcal{N}(0, 1)$ mutations z_l. The largest mutation, denoted by $z_{\lambda:\lambda}$ produces the best offspring and therefore the parent $x^{(g)}$ for the next generation. The variate $z_{\lambda:\lambda}$ is known as the λth order statistics (see e.g. [46]) of the $\mathcal{N}(0, \sigma^2)$ variate, and its expectation is $c_{1,\lambda}\sigma$, with the progress coefficient $c_{1,\lambda}$.

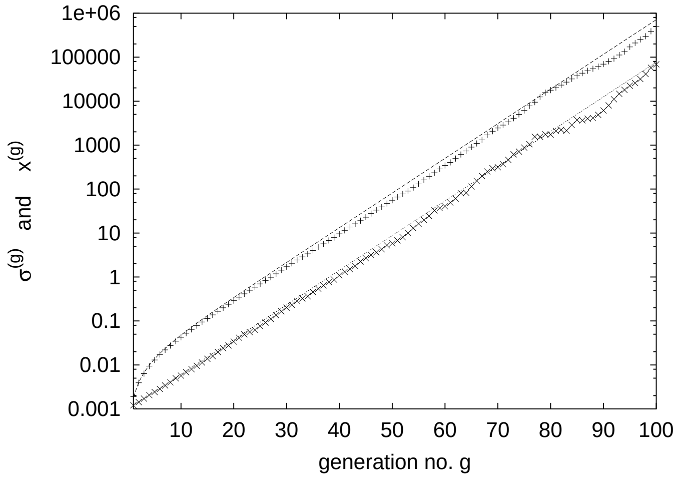

Fig. 8. The σ evolution and the distance x traveled in optimum direction produced by a standard $(1, 10)$-σSA-ES using log-normal mutations on a linear fitness landscape. As learning parameter $\tau = 0.3$ has been chosen, the parameter space dimension (search space) was $N = 30$. The x evolution is displayed by the upper (dashed) curve, the corresponding simulations are depicted by "+" symbols. The σ evolution is displayed by the lower (dotted) curve, the corresponding simulations are depicted by "×" symbols. As initial values $x^{(0)} = 0$ and $\sigma^{(0)} = 0.001$ have been chosen. The results are depicted starting from generation $g = 1$. The simulation results have been obtained by averaging over 500 independent runs.

Acknowledgments

The authors thank the anonymous reviewers for their helpful and supporting comments. The first author acknowledges support as Heisenberg Fellow of the Deutsche Forschungsgemeinschaft under grant Be1578/4-1. The second author acknowledges the support from Alexander von Humboldt Foundation, Germany and Department of Science and Technology, Government of India, during the course of this study.

Appendix

I. On the preservation of the phenotypic population mean under crossover in GAs with binary to integer or real mapping

Standard crossover operators used in binary-coded GAs preserve the phenotypic population mean. It will be proven here for the one-point crossover operator on a pair of parent strings in $\mathbb{B}^\ell = \{0, 1\}^\ell$. For the sake of simplicity we consider the mapping $\mathbb{B}^\ell \mapsto \{0, \mathbb{N}\}$ performed by standard binary coding

$$y = \sum_{i=0}^{\ell-1} 2^i x_i, \quad x_i \in \mathbb{B} \tag{105}$$

Let a and b are two parent Boolean vectors in $\mathbb{B}^\ell$. Let us also assume that the crossing over takes place at the k-th site. The mean of the parents is given by

$$\langle x \rangle = \sum_{i=0}^{\ell-1} 2^i \left(a_i + b_i\right)/2. \tag{106}$$

After crossover, the decoded value of the offspring y_1 and y_2 are as follows

$$y_1 = \sum_{i=0}^{k} 2^i a_i + \sum_{i=k+1}^{\ell-1} 2^i b_i, \tag{107}$$

$$y_2 = \sum_{i=0}^{k} 2^i b_i + \sum_{i=k+1}^{\ell-1} 2^i a_i. \tag{108}$$

The mean of the offspring is $\langle y \rangle = (y_1 + y_2)/2$. Substituting y_1 and y_2 terms from the above two equations, one observes that $\langle y \rangle = \langle x \rangle$. Thus, the mean of a pair of parent solutions is the same as the mean of resulting pair of the offspring. A little thought will reveal that this property is also true for multi-point crossover and uniform crossover operators: one simply has to decompose the sum of (105) in (107) and (108) according to the crossover locations. Avoiding the sampling issue of a finite population, it can also be argued that the expected population mean of a parent population is the same as the expected population mean of the next generation.

II. Details on the calculation of the expected population variance

In order to calculate (13) with (12), the expression in the square brackets of (26) must be rearranged (26) in such a way that products $x_{i,k}x_{j,l}$ for $i = j \wedge k = l$ are separated from the rest. To this end we consider the kth sum terms in (26). Using Eq. (10) and (25) one gets

$$\begin{aligned} y_{1,k} - \langle y \rangle &= \left(\frac{1}{2} - \frac{\beta_k}{2}\right) x_{1,k} + \left(\frac{1}{2} + \frac{\beta_k}{2}\right) x_{2,k} - \frac{1}{\mu}\sum_{l=1}^{\mu/2}(x_{1,l} + x_{2,l}) \\ &= \left(\frac{1}{2} - \frac{1}{\mu} - \frac{\beta_k}{2}\right) x_{1,k} + \left(\frac{1}{2} - \frac{1}{\mu} + \frac{\beta_k}{2}\right) x_{2,k} - \frac{1}{\mu}\sum_{l\neq k}^{\mu/2}(x_{1,l} + x_{2,l}) \end{aligned} \tag{109}$$

and

$$y_{2,k} - \langle y \rangle = \left(\frac{1}{2} - \frac{1}{\mu} + \frac{\beta_k}{2}\right) x_{1,k} + \left(\frac{1}{2} - \frac{1}{\mu} - \frac{\beta_k}{2}\right) x_{2,k} - \frac{1}{\mu}\sum_{l\neq k}^{\mu/2}(x_{1,l} + x_{2,l}). \tag{110}$$

Taking the square yields

$$\begin{aligned} (y_{1,k} - \langle y \rangle)^2 &= \left(\frac{1}{2} - \frac{1}{\mu} - \frac{\beta_k}{2}\right)^2 x_{1,k}^2 + \left(\frac{1}{2} - \frac{1}{\mu} + \frac{\beta_k}{2}\right)^2 x_{2,k}^2 + \frac{1}{\mu^2}\left(\sum_{l\neq k}^{\mu/2}(x_{1,l} + x_{2,l})\right)^2 \\ &\quad +2\left[\left(\frac{1}{2} - \frac{1}{\mu}\right)^2 - \frac{\beta_k^2}{4}\right] x_{1,k}x_{2,k} - 2\left(\frac{1}{2} - \frac{1}{\mu} - \frac{\beta_k}{2}\right) x_{1,k}\frac{1}{\mu}\sum_{l\neq k}^{\mu/2}(x_{1,l} + x_{2,l}) \\ &\quad -2\left(\frac{1}{2} - \frac{1}{\mu} + \frac{\beta_k}{2}\right) x_{2,k}\frac{1}{\mu}\sum_{l\neq k}^{\mu/2}(x_{1,l} + x_{2,l}) \end{aligned} \tag{111}$$

and

$$\begin{aligned} (y_{2,k} - \langle y \rangle)^2 &= \left(\frac{1}{2} - \frac{1}{\mu} + \frac{\beta_k}{2}\right)^2 x_{1,k}^2 + \left(\frac{1}{2} - \frac{1}{\mu} - \frac{\beta_k}{2}\right)^2 x_{2,k}^2 + \frac{1}{\mu^2}\left(\sum_{l\neq k}^{\mu/2}(x_{1,l} + x_{2,l})\right)^2 \\ &\quad +2\left[\left(\frac{1}{2} - \frac{1}{\mu}\right)^2 - \frac{\beta_k^2}{4}\right] x_{1,k}x_{2,k} - 2\left(\frac{1}{2} - \frac{1}{\mu} + \frac{\beta_k}{2}\right) x_{1,k}\frac{1}{\mu}\sum_{l\neq k}^{\mu/2}(x_{1,l} + x_{2,l}) \\ &\quad -2\left(\frac{1}{2} - \frac{1}{\mu} - \frac{\beta_k}{2}\right) x_{2,k}\frac{1}{\mu}\sum_{l\neq k}^{\mu/2}(x_{1,l} + x_{2,l}). \end{aligned} \tag{112}$$

Putting (111) and (112) together, one obtains

$$\begin{aligned} &(y_{1,k} - \langle y \rangle)^2 + (y_{2,k} - \langle y \rangle)^2 \\ &= 2\left[\left(\frac{1}{2} - \frac{1}{\mu}\right)^2 + \frac{\beta_k^2}{4}\right](x_{1,k}^2 + x_{2,k}^2) + \frac{2}{\mu^2}\left(\sum_{l\neq k}^{\mu/2}(x_{1,l} + x_{2,l})\right)^2 \\ &\quad +4\left[\left(\frac{1}{2} - \frac{1}{\mu}\right)^2 - \frac{\beta_k^2}{4}\right] x_{1,k}x_{2,k} - 4\left(\frac{1}{2} - \frac{1}{\mu}\right) x_{1,k}\frac{1}{\mu}\sum_{l\neq k}^{\mu/2}(x_{1,l} + x_{2,l}) \\ &\quad -4\left(\frac{1}{2} - \frac{1}{\mu}\right) x_{2,k}\frac{1}{\mu}\sum_{l\neq k}^{\mu/2}(x_{1,l} + x_{2,l}) \end{aligned} \tag{113}$$

In order to proceed, we calculate the $(\cdots)^2$ term in the second line of (113)

$$
\begin{aligned}
&\left(\sum_{l\neq k}^{\mu/2}(x_{1,l}+x_{2,l})\right)^2 \\
&= \sum_{l\neq k}^{\mu/2}\sum_{m\neq k}^{\mu/2}(x_{1,l}+x_{2,l})(x_{1,m}+x_{2,m}) \\
&= \sum_{l\neq k}^{\mu/2}\sum_{m\neq k}^{\mu/2} x_{1,l}x_{1,m}+x_{1,l}x_{2,m}+x_{2,l}x_{1,m}+x_{2,l}x_{2,m} \\
&= \sum_{l\neq k}^{\mu/2}\sum_{m\neq k}^{\mu/2}(x_{1,l}x_{1,m}+x_{2,l}x_{2,m})+\sum_{l\neq k}^{\mu/2}\sum_{m\neq k}^{\mu/2}x_{1,l}x_{2,m}+\sum_{l\neq k}^{\mu/2}\sum_{m\neq k}^{\mu/2}x_{1,m}x_{2,l} \\
&= \sum_{l\neq k}^{\mu/2}\sum_{m\neq k}^{\mu/2}(x_{1,l}x_{1,m}+x_{2,l}x_{2,m})+2\sum_{l\neq k}^{\mu/2}\sum_{m\neq k}^{\mu/2}x_{1,l}x_{2,m} \\
&= \sum_{l\neq k}^{\mu/2}(x_{1,l}^2+x_{2,l}^2)+\sum_{l\neq k}^{\mu/2}\sum_{m\neq l}^{\mu/2}(x_{1,l}x_{1,m}+x_{2,l}x_{2,m})+2\sum_{l\neq k}^{\mu/2}\sum_{m\neq k}^{\mu/2}x_{1,l}x_{2,m}.
\end{aligned}
\tag{114}
$$

Inserting this into (113) gives

$$
\begin{aligned}
&(y_{1,k}-\langle y\rangle)^2+(y_{2,k}-\langle y\rangle)^2 \\
&= 2\left[\left(\frac{1}{2}-\frac{1}{\mu}\right)^2+\frac{\beta_k^2}{4}\right](x_{1,k}^2+x_{2,k}^2)+\frac{2}{\mu^2}\sum_{l\neq k}^{\mu/2}(x_{1,l}^2+x_{2,l}^2) \\
&\quad +4\left[\left(\frac{1}{2}-\frac{1}{\mu}\right)^2-\frac{\beta_k^2}{4}\right]x_{1,k}x_{2,k}-4\left(\frac{1}{2}-\frac{1}{\mu}\right)\frac{1}{\mu}\sum_{l\neq k}^{\mu/2}(x_{1,k}x_{1,l}+x_{1,k}x_{2,l}+x_{2,k}x_{1,l}+x_{2,k}x_{2,l}) \\
&\quad +\frac{2}{\mu^2}\sum_{l\neq k}^{\mu/2}\sum_{m\neq l}^{\mu/2}(x_{1,l}x_{1,m}+x_{2,l}x_{2,m})+\frac{4}{\mu^2}\sum_{l\neq k}^{\mu/2}\sum_{m\neq k}^{\mu/2}x_{1,l}x_{2,m}.
\end{aligned}
\tag{115}
$$

In order to calculate (21), the expectation of the bracket in (26) must be determined. Using (115) and (27) one obtains

$$
\begin{aligned}
&\mathrm{E}\left[(y_{1,k}-\langle y\rangle)^2+(y_{2,k}-\langle y\rangle)^2\right] \\
&= \left[4\left(\frac{1}{2}-\frac{1}{\mu}\right)^2+\overline{\beta_k^2}\right]\overline{x^2}+\frac{4}{\mu^2}\overline{x^2}\sum_{l\neq k}^{\mu/2}1 \\
&\quad +\left[4\left(\frac{1}{2}-\frac{1}{\mu}\right)^2-\overline{\beta_k^2}\right]\overline{x}^2-4\left(\frac{1}{2}-\frac{1}{\mu}\right)\frac{1}{\mu}\overline{x}^2\sum_{l\neq k}^{\mu/2}(1+1+1+1) \\
&\quad +\frac{2}{\mu^2}\overline{x}^2\sum_{l\neq k}^{\mu/2}\sum_{m\neq l}^{\mu/2}(1+1)+\frac{4}{\mu^2}\overline{x}^2\sum_{l\neq k}^{\mu/2}\sum_{m\neq k}^{\mu/2}1.
\end{aligned}
\tag{116}
$$

Collecting all $\overline{x^2}$ and $\overline{x}^2$ terms yields a remarkably simple result

$$
\begin{aligned}
&\mathrm{E}\left[(y_{1,k}-\langle y\rangle)^2+(y_{2,k}-\langle y\rangle)^2\right] \\
&= \left[\overline{\beta_k^2}+4\left(\frac{1}{2}-\frac{1}{\mu}\right)^2+\left(\frac{\mu}{2}-1\right)\frac{4}{\mu^2}\right]\overline{x^2} \\
&\quad -\left[\overline{\beta_k^2}-4\left(\frac{1}{2}-\frac{1}{\mu}\right)^2+16\left(\frac{1}{2}-\frac{1}{\mu}\right)\frac{1}{\mu}\left(\frac{\mu}{2}-1\right)-\frac{4}{\mu^2}\left(\frac{\mu}{2}-1\right)\left(\frac{\mu}{2}-2\right)-\frac{4}{\mu^2}\left(\frac{\mu}{2}-1\right)^2\right]\overline{x}^2 \\
&= \left[1-\frac{2}{\mu}+\overline{\beta_k^2}\right]\left(\overline{x^2}-\overline{x}^2\right).
\end{aligned}
\tag{117}
$$

After back-substitution in (21) taking (26) into account one obtains

$$\begin{aligned}\sigma^2[y] &= \frac{1}{\mu}\sum_{k=1}^{\mu/2} \mathrm{E}\left[(y_{1,k}-\langle y\rangle)^2+(y_{2,k}-\langle y\rangle)^2\right] \\ &= \frac{1}{\mu}\sum_{k=1}^{\mu/2}\left[1-\frac{2}{\mu}+\overline{\beta_k^2}\right]\left(\overline{x^2}-\overline{x}^2\right) \\ &= \frac{1}{\mu}\frac{\mu}{2}\left[1-\frac{2}{\mu}+\overline{\beta_k^2}\right]\left(\overline{x^2}-\overline{x}^2\right). \end{aligned} \tag{118}$$

Since the β_k are samples from the same distribution, we write $\overline{\beta_k^2} = \overline{\beta^2}$. Taking (22) is into account, we finally arrive at the evolution equation of the expected population variance

$$\sigma^{2\,(g+1)} = \frac{1}{2}\left(1-\frac{2}{\mu}+\overline{\beta^2}\right)\sigma^{2\,(g)}. \tag{119}$$

References

[1] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Frommann–Holzboog Verlag, Stuttgart, 1973.

[2] I. Rechenberg. *Evolutionsstrategie '94.* Frommann–Holzboog Verlag, Stuttgart, 1994.

[3] H.-P. Schwefel. *Evolution and Optimum Seeking.* Wiley, New York, NY, 1995.

[4] D. B. Fogel. *Evolving Artificial Intelligence.* PhD thesis, University of California, San Diego, 1992.

[5] D. B. Fogel. *Evolutionary Computation.* IEEE Press, New York, 1995.

[6] T. Bäck. The Interaction of Mutation Rate, Selection, and Self-Adaptation Within a Genetic Algorithm. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature, 2*, pages 85–94. North Holland, Amsterdam, 1992.

[7] J. Smith and T. C. Fogarty. Self-Adaptation of Mutation Rates in a Steady State Genetic Algorithm. In *Proceedings of 1996 IEEE Int'l Conf. on Evolutionary Computation (ICEC '96)*, pages 318–323. IEEE Press, NY, 1996.

[8] L. Davis. Adapting Operator Probabilities in Genetic Algorithms. In J. D. Schaffer, editor, *Proc. 3rd Int'l Conf. on Genetic Algorithms*, pages 61–69, San Mateo, CA, 1989. Morgan Kaufmann.

[9] J.D. Schaffer and A. Morishima. An Adaptive Crossover Distribution Mechanism for Genetic Algorithms. In J.J. Grefenstette, editor, *Genetic Algorithms and their Applications: Proc. of the Second Int'l Conference on Genetic Algorithms*, pages 36–40, 1987.

[10] H.-G. Beyer. Some Aspects of the 'Evolution Strategy' for Solving TSP-like Optimization Problems. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature, 2*, pages 361–370, Amsterdam, 1992. Elsevier.

[11] K. Chellapilla and D.B. Fogel. Exploring Self-Adaptive Methods to Improve the Efficiency of Generating Approximate Solutions to Traveling Salesman Problems Using Evolutionary Programming. In P.J. Angeline, R.G. Reynolds, J.R. McDonnell, and R. Eberhart, editors, *Evolutionary Programming VI: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, pages 361–371. Springer, Berlin, 1997.

[12] T. Bäck and M. Schütz. Evolution Strategies for Mixed-Integer Optimization of Optical Multilayer Systems. In J. R. McDonnell, R. G. Reynolds, and D. B. Fogel, editors, *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, pages 33–51. MIT Press, Cambridge, MA, 1995.

[13] L. J. Fogel, P. J. Angeline, and D. B. Fogel. An evolutionary programming approach to self-adaptation on finite state machines. In J. R. McDonnell, R. G. Reynolds, and D. B. Fogel, editors, *Proceedings of the Fourth International Conference on Evolutionary Programming*, pages 355–365, , 1995.

[14] H.-P. Schwefel. Collective phenomena in evolutionary systems. In P. Checkland and I. Kiss, editors, *Problems of Constancy and Change — the Complementarity of Systems Approaches to Complexity, Papers presented at the 31st Annual Meeting of the Int'l Soc. for General System Research*, volume 2, pages 1025–1033, Budapest, 1.–5. Juni 1987. Int'l Soc. for General System Research.

[15] H.-G. Beyer. Toward a Theory of Evolution Strategies: Self-Adaptation. *Evolutionary Computation*, 3(3):311–347, 1996.

[16] H.-G. Beyer. *The Theory of Evolution Strategies.* Natural Computing Series. Springer, Heidelberg, 2001. ISBN 3-540-67297-4.

[17] X. Yao and Y. Liu. Fast Evolutionary Programming. In L. J. Fogel, P. J. Angeline, and T. Bäck, editors, *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 451–460. The MIT Press, Cambridge, MA, 1996.

[18] P. J. Angeline. The Effects of Noise on Self-Adaptive Evolutionary Optimization. In L. J. Fogel, P. J. Angeline, and T. Bäck, editors, *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 432–439. The MIT Press, Cambridge, MA, 1996.

[19] T. Bäck. Self-adaptation. In T. Bäck, D. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, pages C7.1:1–C7.1:15. Oxford University Press, New York, 1997.

[20] N. Hansen, A. Ostermeier, and A. Gawelczyk. On the Adaptation of Arbitrary Normal Mutation Distributions in Evolution Strategies: The Generating Set Adaptation. In L. J. Eshelman, editor, *Proc. 6th Int'l Conf. on Genetic Algorithms*, pages 57–64, San Francisco, CA, 1995. Morgan Kaufmann Publishers, Inc.

[21] A. Ostermeier and N. Hansen. An Evolution Strategy with Coordinate System Invariant Adaptation of Arbitrary Mutation Distributions Within the Concept of Mutative Strategy Parameter Control. In W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, and R.E. Smith, editors, *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 902–909, San Francisco, CA, 1999. Morgan Kaufmann.

[22] K. Deb and H.-G. Beyer. Self-Adaptive Genetic Algorithms with Simulated Binary Crossover. *Evolutionary Computation*, 9(2):197–221, 2001.

[23] K. Deb and H.-G. Beyer. Self-Adaptation in Real-Parameter Genetic Algorithms with Simulated Binary Crossover. In W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, and R.E. Smith, editors, *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 172–179, San Francisco, CA, 1999. Morgan Kaufmann.

[24] H. Kita. A comparison study on self-adaptation in evolution strategies and real-coded genetic algorithms. Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, Japan, 1998.

[25] H.-M. Voigt, H. Mühlenbein, and D. Cvetković. Fuzzy recombination for the Breeder Genetic Algorithm. In L. J. Eshelman, editor, *Proc. 6th Int'l Conf. on Genetic Algorithms*, pages 104–111, San Francisco, CA, 1995. Morgan Kaufmann Publishers, Inc.

[26] L. J. Eshelman and J. D. Schaffer. Real-coded genetic algorithms and interval schemata. In L. D. Whitley, editor, *Foundations of Genetic Algorithms, 2*, pages 187–202. Morgan Kaufmann, San Mateo, CA, 1993.

[27] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148, 1995.

[28] I. Ono and S. Kobayashi. A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover. In T. Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 246–253, Morgan Kauffman, San Mateo, CA, 1997.

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

[30] H.-G. Beyer. On the "Explorative Power" of ES/EP-like Algorithms. In V.W. Porto, N. Saravanan, D. Waagen, and A.E. Eiben, editors, *Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming*, pages 323–334, Heidelberg, 1998. Springer-Verlag.

[31] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer, Berlin, 2nd edition, 1994.

[32] A. Ostermeier. *Schrittweitenadaptation in der Evolutionsstrategie mit einem entstochastisierten Ansatz.* Doctoral thesis, Technical University of Berlin, Berlin, 1997.

[33] M. Herdy. Reproductive Isolation as Strategy Parameter in Hierarchically Organized Evolution Strategies. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature, 2*, pages 207–217. Elsevier, Amsterdam, 1992.

[34] A. I. Oyman, H.-G. Beyer, and H.-P. Schwefel. Where Elitists Start Limping: Evolution Strategies at Ridge Functions. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature, 5*, pages 34–43, Heidelberg, 1998. Springer.

[35] A. I. Oyman. *Convergence Behavior of Evolution Strategies on Ridge Functions.* Ph.D. Thesis, University of Dortmund, Department of Computer Science, 1999.

[36] H.-G. Beyer. On the Dynamics of EAs without Selection. In W. Banzhaf and C. Reeves, editors, *Foundations of Genetic Algorithms, 5*, pages 5–26, San Mateo, CA, 1999. Morgan Kaufmann.

[37] L. J. Eshelman. The CHC Adaptive Search Algorithm: How to Have Save Search When Engaging in Nontraditional Genetic Recombination. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms, 1*, pages 265–283. Morgan Kaufmann, San Mateo, CA, 1991.

[38] H.-G. Beyer. Toward a Theory of Evolution Strategies: The (μ, λ)-Theory. *Evolutionary Computation*, 2(4):381–407, 1995.

[39] T. Bäck and H.-P. Schwefel. An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation*, 1(1):1–23, 1993.

[40] T. Bäck, U. Hammel, and H.-P Schwefel. Evolutionary computation: comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1(1):3–17, 1997.

[41] L. Grünz and H.-G. Beyer. Some Observations on the Interaction of Recombination and Self-Adaptation in Evolution Strategies. In P.J. Angeline, editor, *Proceedings of the CEC'99 Conference*, pages 639–645, IEEE, Piscataway, NJ, 1999.

[42] H.-G. Beyer. Toward a Theory of Evolution Strategies: On the Benefit of Sex – the $(\mu/\mu, \lambda)$-Theory. *Evolutionary Computation*, 3(1):81–111, 1995.

[43] N. Hansen. *Verallgemeinerte individuelle Schrittweitenregelung in der Evolutionsstrategie.* Doctoral thesis, Technical University of Berlin, Berlin, 1998.

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

[45] H.-G. Beyer. Towards a Theory of 'Evolution Strategies': The $(1, \lambda)$-Self-Adaptation. Technical Report SYS-1/95, Department of Computer Science, University of Dortmund, 1995.

[46] B. C. Arnold, N. Balakrishnan, and H. N. Nagaraja. *A First Course in Order Statistics.* Wiley, New York, 1992.