Natural Computing **1:** 3–52, 2002.

Evolution strategies

A comprehensive introduction

HANS-GEORG BEYER and HANS-PAUL SCHWEFEL
Department of Computer Science XI, University of Dortmund, Joseph-von-Fraunhoferstr. 20, D-44221 Dortmund, Germany (E-mail: beyer@Ls11.cs.uni-dortmund.de; hps@udo.edu)

Abstract. This article gives a comprehensive introduction into one of the main branches of evolutionary computation – the evolution strategies (ES) the history of which dates back to the 1960s in Germany. Starting from a survey of history the philosophical background is explained in order to make understandable why ES are realized in the way they are. Basic ES algorithms and design principles for variation and selection operators as well as theoretical issues are presented, and future branches of ES research are discussed.

Key words: computational intelligence, Darwinian evolution, design principles for genetic operators, evolutionary computation, evolution strategies, optimization

Abbreviations: BBH – building block hypothesis; CMA – covariance matrix adaptation; CSA – cumulative step-size adaptation; EA – evolutionary algorithm; EC – evolutionary computation; ES – evolution strategy; EP – evolutionary programming; GA – genetic algorithm; GR – genetic repair; TSP – traveling salesman problem; TUB – Technical University of Berlin

ACM Computing Classification (1998): G.1.6, F1.2, G.3–4, I.2.8, J.3

1. Introduction

Biological inspiration in computer science, especially in algorithms engineering, can be traced back to the early pioneers in the field. The list of illustrious names includes John von Neumann, who had more in mind than single-processor number crunchers. Far beyond the technological possibilities of their era those pioneers already dreamt of intelligent and even self-reproducing machines.

Three paradigms aside of the strong artificial intelligence realm have survived the last three to four decades and recently found a common umbrella under domain names like computational intelligence (Michalewicz et al., 1994; Fogel et al., 1998), soft computing, or natural computation. Two of them mimic individual brain procedures by means of either using artificial neural networks or reasoning with fuzzy logic. The third paradigm, evolutionary computation (EC), aims at benefiting from collective phenomena in

adaptive populations of problem solvers underlying birth and death, variation and selection, in an iterative, respectively generational, loop.

Again, three of the nearly contemporaneous sources of the evolutionary algorithms (EA) have been kept alive over three decades and experienced an amazing increase of interest during the last fifteen years. Two of them are lying in the United States of America, the source of evolutionary programming (EP) in San Diego (Fogel, 1962; Fogel et al., 1966), the source of genetic algorithms (GA) in Ann Arbor (Holland, 1962; Holland, 1975). Evolution strategies (ES), the third main variant of EA, were founded by students at the Technical University of Berlin (TUB) (Rechenberg, 1965; Rechenberg, 1971; Schwefel, 1965; Schwefel, 1975).

This paper gives an overview over the ES, its birth and history, its basic ideas and philosophy, and its state-of-the-art. It is organized as follows. Section 2 is intended to provide deeper insight into the history of ES research building the basis and philosophy of contemporary ES. The basic ideas and principles as well as the ingredients for designing ES algorithms, such as mutation, recombination, and selection operators are introduced and explained in section 3. As we will see, performance aspects are closely related to the adaptive behavior of the operators mentioned. Section 4 is devoted to the different approaches to adaptation. Theoretical aspects of ES research such as the evolutionary dynamics, convergence, local and global performance issues in real-valued and discrete search spaces, and time complexity are the subject of section 5. The paper closes with an outlook providing some ideas on the future evolution of ES research.

2. A short history of evolution strategy research

In the following we concentrate on ES and refer the historically interested reader to a less concise report about the history of all EA in the Handbook of Evolutionary Computation (De Jong et al., 1997). An even more shimmering picture of the "fossil record" has been gathered in a voluptuous volume with annotated reprints dating back into the late fifties of the last century (Fogel, 1998).

2.1 *Experimental evolutionary optimization*

In the beginning, evolution strategies were not devised to compute minima or maxima of real-valued static functions with fixed numbers of variables and without noise during their evaluation. Rather, they came to the fore as a set of rules for the automatic design and analysis of consecutive experiments with stepwise variable adjustments driving a suitably flexible object/system

into its optimal state in spite of environmental noise, e.g., a slender 3D body in a wind tunnel flow into a shape with minimal drag per volume. The experimentum crucis was performed in June 1964 with a 2D joint plate in turbulent air flow and demonstrated that a simple randomized heuristic outperformed the univariate as well as a discrete gradient-oriented strategy, which were adopted from numerical mathematics. Under noisy and obviously multimodal conditions, the new strategy showed up to be effective and sufficiently efficient to be used for a couple of other experimental optimization tasks, e.g., the design of a 3D convergent-divergent hot water flashing nozzle (Schwefel, 1968; Klockgether and Schwefel, 1970). In order to be able to vary the length of the nozzle and the position of its throat, gene duplication and gene deletion was mimicked to evolve even the number of variables, i.e., the nozzle diameters at fixed distances. The perhaps optimal, at least unexpectedly good and so far best-known shape of the nozzle was counterintuitively strange, and it took a while, until the one-component two-phase supersonic flow phenomena far from thermodynamic equilibrium, involved in achieving such good result, were understood.

Being enthusiasts of the then "modern" disciplines cybernetics and bionics, the students called their simple set of rules for designing and evaluating the consecutive experiments "cybernetic solution path" and "evolution strategy." Actually, there were no more than two rules:

1. Change all variables at a time, mostly slightly and at random.
2. If the new set of variables does not diminish the goodness of the device, keep it, otherwise return to the old status.

Rule one seemed to resemble mutations in nature, and rule two literally modeled the "survival of the fittest", thus corresponding to Spencer's explanation of Darwin's natural selection principle. Since there was just one old (parent) and just one new (offspring) object (individual) per iteration (generation) and since the selection took place among those two, this simplest form of an ES has later been called two-membered or $(1 + 1)$-ES. Its opponents often mistook (malevolently?) the term mutation as a synonym for pure random trials, thus denying any kind of learning or adaptation in the course of stepwise improvements. Even today, chance is mostly taken as a synonym for accidents to be avoided than for opportunities, unpredicted/unexpected innovations, or lucky discoveries.

2.2 *From discrete to continuous decision variables and first theoretical results*

A diploma thesis (Schwefel, 1965) investigated the $(1 + 1)$-ES with binomially distributed mutations on a two dimensional parabolic ridge.[1] The study found out that the process could get stuck at certain positions, because

nearest-neighbor positions were all worse, whereas improvements needed larger mutations that were improbable if not impossible under the mutation distribution used. This led to the idea of using continuous variables and Gaussian distributions for the changes of the variables.

In 1971 the first dissertation (Rechenberg, 1971; Rechenberg, 1973) in the field of ES was completed. It presented approximate analyses of the $(1 + 1)$-ES with Gaussian mutations on two very different real-valued functions, the hypersphere and rectangular corridor models, showing that

- the convergence velocity, i.e., the expected distance traveled into the useful direction per iteration, is inversely proportional to the number of variables and proportional to an orographic measure of the isohypses' curvedness of the objective function;
- linear convergence order can be achieved if the mutation strength (or mean step-size or standard deviation of each component of the normally distributed mutation vector) is adjusted to the proper order of magnitude, permanently;
- the optimal mutation strength corresponds to a certain success probability that is independent of the dimension of the search space and in the range of one fifth for both model functions.

By means of simulating the first type of a multimembered ES, the $(\mu + 1)$- or steady-state ES, Rechenberg further demonstrated that

- crossover can speed up the evolution substantially if the speed is measured per generation, not per function evaluation;
- the population may learn itself to adjust the mutation strength underlying an embedded mutation process without external control.

There are μ parents at a time in the $(\mu + 1)$-ES. Two of them are chosen at random and recombined to give life to an offspring, which also underlies mutation. The selection resembles "extinction of the worst," may it be the offspring or one of the parents, thus keeping constant the population size.

2.3 *Numerical evolutionary optimization superimposed onto simulation models*

By the time the (main frame) computers entered the campuses, it was tempting to adapt the newborn strategy to problems of searching for optimal parameters within simulation models, e.g. models simulating stress and deviation of civil engineering structures under load (Hartmann, 1974). Such tasks were the domain of numerical optimization procedures that had seen a boom phase in the 1960s, mainly under the topic of operations research, scarcely in the engineering sciences. Obviously, it was necessary to compare the efficiency and robustness of the evolution strategies with that of its competitors, especially those, which do not need derivatives explicitly. The results of that

study were presented in another dissertation at the TUB in 1974 (Schwefel, 1975; Schwefel, 1977).

Like with all classical methods, the performance of the evolution strategies largely depends on the adjustment of the internal parameters, prominently the mutation strength(s). The observation that the above mentioned first multimembered ES has an intrinsic tendency toward reducing the mutation strength, whether this is appropriate or not, led Schwefel to introduce two further versions of multimembered ES, i.e.,

- the $(\mu + \lambda)$-ES, in which not only one offspring is created at a time or in a generation, but $\lambda \geq 1$ descendants, and, to keep the population size constant, the λ worst out of all $\mu + \lambda$ individuals are discarded;
- the (μ, λ)-ES, in which the selection takes place among the λ offspring only, whereas their parents are "forgotten" no matter how good or bad their fitness was compared to that of the new generation. Obviously, this strategy relies on a birth surplus, i.e., on $\lambda > \mu$ in a strict Darwinian sense of natural selection.

Whereas the first applications of ES in the field of experimental optimization were greeted as innovative despite of lacking proofs of convergence toward an absolute (global) optimum, people were much more sceptical with respect to numerical evolutionary optimization and its possible benefits. Besides of other reasons, the 1960s had seen substantial progress in classical numerical optimization including theory as well as practical recipes for deterministic heuristics. No need for other methods was seen, especially not for a randomized biologically inspired one operating with more than one search point at a time.

Had the $(\mu + 1)$-ES already been laughed at because it makes use not only of the so far best individual to produce an offspring, but also of the second best, even the worst out of μ parents, one can imagine that the $(\mu + \lambda)$-ES was welcomed as a further step into a wrong direction, because it does not make immediate use of new information gathered by the next offspring. Instead, it delays the selection decision until all λ descendants are born. The step from the plus to the comma version finally seemed to climb to the top of mount nonsense in so far as now even better intermediate solutions can get lost and be replaced by worse ones.

It is true that it is easier to assure convergence to a $(\mu + \lambda)$-ES because its worst behavior is premature stagnation, e.g., when the mutation strength becomes too low before reaching the optimum, whereas a comma version can even diverge, especially when the mutation strength is too high. The motifs for introducing the seemingly ridiculous ES versions with μ parents and λ offspring per generation were twofold:

1. Self-adaptation to the optimal mutation strength at a success probability P_s should generally require a birth surplus or selection pressure λ/μ of at least $1/P_s$. Otherwise, smaller mean step-sizes with higher success probabilities are preferred. The comma ES with negative progress in case of too large step sizes should enhance the self-adaptation by preferring offspring which step back least in such situation. Within large populations with diverse individuals, both with respect to the object variables as well as the strategy variable(s), i.e., the mean step-size(s) of the mutations, the difference between a plus and a comma version should fade away.
2. On parallel computers with λ processors the set of offspring could easily be evaluated at the same time, thus speeding up the evolution substantially – per generation.[2] However, nearly no expert in the 1970s believed in commercial parallel computing before the end of the millennium.

Next to the self-adaptation of one common standard deviation for all changes of N object variables, the multimembered ES offers a chance to let N variances vary individually. This should lead to an opportunity of self-scaling if it were important to largely differentiate the changes of the different object variables. One may achieve such a feature in several ways. Schwefel (1975) mentioned the possibility of introducing two separate factors for mutating the standard deviations by means of multiplying with one log-normally distributed factor common for all mean step-sizes and N additional log-normally distributed factors acting on individual mean step-sizes in different ways. His numerical experiments, however, used a special form of intermediary recombination to let the N standard deviations adapt individually. If the selection pressure is too high and the population size too small, there is a tendency to search in subspaces of the N-dimensional search space. Otherwise, the collective online learning of the internal or strategy parameters works quite well (Schwefel, 1987). The two-factors approach is dangerous just as the comma version and the mutative self-adaptation as such (Kursawe, 1999).

The utmost flexibility of the Gaussian normal distribution is reached when there are not only N individual positive variances but also up to $N(N-1)/2$ covariances leading to rotating hyperellipsoids as surfaces of equal mutation probability densities. Such proposal was made and investigated already in 1975, but not published before 1980/81 (e.g. Schwefel, 1981). Generating correlated mutations with proper characteristics can be achieved by means of an N-fold rotation procedure (Rudolph, 1992). The lack of ample number crunching power in the 1970s, the lack of thorough theoretical analyses, and – still today – the tendency to work with very small populations has led to neglect the full flexibility of the normal distribution.

3. The basic ES-Algorithm

This section presents the ingredients of state-of-the-art ES capable of *self-adaptation* (concerning the meaning of self-adaptation, see section 4.2). In section 3.1 the standard $(\mu/\rho \overset{+}{,} \lambda)$-ES notation will be introduced and the general ES algorithm will be outlined. Section 3.2 is devoted to the selection operator, section 3.3 deals with the mutation operator, and section 3.4 describes the recombination operators.

3.1 *The $(\mu/\rho \overset{+}{,} \lambda)$-ES Notation and Algorithm*

The usual goal of an evolution strategy is to optimize[3] (some) given objective or quality function(s) F with respect to a set of decision variables or control parameters $\mathbf{y} := (y_1, y_2, \ldots)$ – in ES context – often referred to as *object parameters*

$$F(\mathbf{y}) \to \text{opt.}, \qquad \mathbf{y} \in \mathcal{Y}. \tag{1}$$

In principle, $\mathcal{Y}$ can be any set of data structures of finite but not necessarily fixed length. Examples for $\mathcal{Y}$ are the real-valued N-dimensional search space $\mathbb{R}^N$, the integer search space $\mathbb{Z}^N$, the binary search space $\mathbb{B}^N$, the space of permutations $\mathbb{P}_N$ as well as mixtures of different spaces and subspaces (due to constraints).

Evolution strategies operate on populations $\mathfrak{P}$ of individuals $\mathfrak{a}$. An individual $\mathfrak{a}_k$ with index k comprises not only the specific object parameter set (or vector) $\mathbf{y}_k$ and its objective function value $F_k := F(\mathbf{y}_k)$, sometimes referred to as *fitness*, but usually also a set of *endogenous* (i.e. evolvable) *strategy parameters* $\mathbf{s}_k$

$$\mathfrak{a}_k := (\mathbf{y}_k, \mathbf{s}_k, F(\mathbf{y}_k)). \tag{2}$$

The endogenous strategy parameters are a peculiarity of ES: they are used to control certain statistical properties of the genetic operators, especially those of the mutation operator (see section 3.3). Endogenous strategy parameters can evolve during the evolution process and are needed in self-adaptive ES (see section 4.2).

Within one ES generation step, λ offspring individuals $\tilde{\mathfrak{a}}_l$ (note, the tilde is used to mark complete offspring) are generated from the set of μ parent individuals $\mathfrak{a}_m$. That is, the size λ of the offspring population $\mathfrak{P}_\text{o}$ is usually not equal to the size μ of the parent population $\mathfrak{P}_\text{p}$. The strategy-specific parameters μ and λ as well as ρ (the mixing number, see below) are called "*exogenous* strategy parameters" which are kept constant during the evolution run.

Procedure $(\mu/\rho \overset{+}{,} \lambda)$-ES;	line
Begin	1
$g := 0;$	2
$\texttt{initialize}\left(\mathfrak{P}_{\mathrm{p}}^{(0)} := \left\{\left(\mathbf{y}_m^{(0)}, \mathbf{s}_m^{(0)}, F(\mathbf{y}_m^{(0)})\right),\ m = 1, \ldots, \mu\right\}\right);$	3
Repeat	4
For $l := 1$ **To** λ **Do Begin**	5
$\mathfrak{E}_l := \texttt{marriage}\left(\mathfrak{P}_{\mathrm{p}}^{(g)}, \rho\right);$	6
$\mathbf{s}_l := \texttt{s\_recombination}(\mathfrak{E}_l);$	7
$\mathbf{y}_l := \texttt{y\_recombination}(\mathfrak{E}_l);$	8
$\tilde{\mathbf{s}}_l := \texttt{s\_mutation}(\mathbf{s}_l);$	9
$\tilde{\mathbf{y}}_l := \texttt{y\_mutation}\left(\mathbf{y}_l, \tilde{\mathbf{s}}_l\right);$	10
$\tilde{F}_l := F(\tilde{\mathbf{y}}_l)$	11
End;	12
$\mathfrak{P}_{\mathrm{o}}^{(g)} := \left\{\left(\tilde{\mathbf{y}}_l, \tilde{\mathbf{s}}_l, \tilde{F}_l\right),\ l = 1, \ldots, \lambda\right\};$	13
Case selection_type **Of**	14
(μ, λ) : $\mathfrak{P}_{\mathrm{p}}^{(g+1)} := \texttt{selection}\left(\mathfrak{P}_{\mathrm{o}}^{(g)}, \mu\right);$	15
$(\mu + \lambda)$: $\mathfrak{P}_{\mathrm{p}}^{(g+1)} := \texttt{selection}\left(\mathfrak{P}_{\mathrm{o}}^{(g)}, \mathfrak{P}_{\mathrm{p}}^{(g)}, \mu\right)$	16
End;	17
$g := g + 1;$	18
Until termination_condition	19
End	20

Figure 1. Pseudo-code of the $(\mu/\rho \overset{+}{,} \lambda)$-ES

The way the offspring population is generated is expressed by the $(\mu/\rho \overset{+}{,} \lambda)$ notation to be read as "*mu slash rho plus or comma lambda*." The ρ refers to the number of parents involved in the procreation of *one* offspring (mixing number). For $\rho = 1$ (cloning), we have the special ES cases *without* recombination, usually denoted by (μ, λ) and $(\mu+\lambda)$, respectively.[4] All other cases ($\rho > 1$) are strategies with recombination. The "$\overset{+}{,}$" refers to the kind of selection used, i.e., "+"- and ","-selection, respectively (see section 3.2, below).

Figure 1 shows the pseudo-code of the $(\mu/\rho \overset{+}{,} \lambda)$-ES. In what follows we give a brief description of the algorithm postponing most of the details to the next sections.

At generation $g = 0$, the parental population $\mathfrak{P}_{\mathrm{p}}^{(0)} := (\mathfrak{a}_1, \ldots, \mathfrak{a}_\mu)$ is initialized in line #3. After initialization the repeat-until-loop is entered (lines

#4–19). From the parental population $\mathfrak{P}_{\mathrm{p}}^{(g)}$ at generation g a new offspring population $\mathfrak{P}_{\mathrm{o}}^{(g)}$ is produced by running λ times through the lines #6–11. Each cycle generates one offspring: First, in the marriage step, a parent family $\mathfrak{E}$ of size ρ is randomly chosen from the parent pool of size μ. This marriage selection process is random-based, i.e., it is independent of the parental objective values F. (In the special case of $\rho = \mu$, all parents become members of the parent family $\mathfrak{E}$.) This is in contrast to standard selection techniques in genetic algorithms (Goldberg, 1989). Recombination of the endogenous strategy parameters takes place in line #7 and for the object parameters in line #8. Note, if $\rho = 1$, the recombinant is simply a copy of the parent. The mutation of the strategy parameters is done in line #9 and those of the object parameters in line #10. While the order of recombination (lines #7 and 8) can be exchanged, the application order of the mutation operators must not be changed in order to ensure a correctly working self-adaptation (see section 4.2).

After having a complete offspring population $\mathfrak{P}_{\mathrm{o}}^{(g)}$, selection is performed with the result of a new parent population $\mathfrak{P}_{\mathrm{p}}^{(g+1)}$.

Finally the termination condition is checked. As termination conditions the standard stopping rules can be used

- resource criteria:
  - maximum number of generations,
  - maximum cpu-time
- convergence criteria:
  - in the space of fitness values,
  - in the space of object parameters,
  - in the space of strategy parameters.

3.2 *Selection*

Each evolutionary algorithm needs a goal oriented selection operator in order to guide the search into promising regions of the object parameter space. Selection is thus the antagonist to the variation operators (also referred to as genetic operators) mutation and recombination. It gives the evolution a direction. Selection in ES is just like animal or plant breeding: only those individuals with promising properties, e.g., high fitness values (objective function values), get a chance of reproduction. That is, a new parental population at $(g + 1)$ is obtained by a *deterministic* process guaranteeing that only the μ best individuals $\mathfrak{a}$ from the selection pool of generation (g) are transferred into $\mathfrak{P}_{\mathrm{p}}^{(g+1)}$ (this selection technique is also called "truncation" or "breeding" selection)

$$\mathfrak{P}_{\mathrm{p}}^{(g+1)} := \{\mathfrak{a}_{1;\gamma}, \ldots, \mathfrak{a}_{\mu;\gamma}\}. \tag{3}$$

Here we have used the "$m;\gamma$" notation reminiscent of the "$\mu{:}\gamma$" notation of *order statistics* (see e.g. Arnold et al., 1992). It means

$$\mathfrak{a}_{m;\gamma} := \text{"Take the } m\text{th best individual out of } \gamma \text{ individuals."} \tag{4}$$

There are two versions of this selection technique, depending on whether or not the parental population at (g) is included in this process, i.e., *plus* selection, denoted by $(\mu + \lambda)$, and *comma* selection, denoted by (μ, λ), respectively.

In the case of (μ, λ) selection, only the λ newly generated offspring individuals, i.e. the $\mathfrak{P}_{\mathrm{o}}^{(g)}$ population, define the selection pool. In other words, the parents from generation (g) are forgotten *per definitionem* even when they are better than all offspring. It is quite clear that a necessary (but not sufficient) condition of convergence toward an optimal solution must be $\mu < \lambda$. The limit case $\mu = \lambda$ cannot work because all offspring are selected as parents. That is, selection provides no search-relevant information and as a result the population performs a random walk in the search space.

In contrast to comma selection, the plus selection takes the old parents into account. The $(\mu+\lambda)$ notation indicates that *both* the parents *and* the offspring are copied into the selection pool which is therefore of size $\gamma = \mu+\lambda$. Unlike comma selection, there is no theoretical restriction on the offspring number λ. Cases with $\mu = \lambda$ or $\mu > \lambda$ are possible. The special case $(\mu + 1)$ is also referred to as *steady-state* ES. It is often used in asynchronous parallel implementations on multiprocessor systems (see, e.g., Kappler et al., 1996).

Plus selection guarantees the survival of the best individual found so far. Since it preserves the best individual such selection techniques are also called *elitist*. Elitism is a sufficient condition a selection operator should obey in order to prove the ES's global convergence property. Due to the elitism in plus strategies, parents can survive an infinitely long time-span.

Both selection variants have their specific application areas. While the (μ, λ) selection is recommended for unbounded search spaces $\mathcal{Y}$, especially $\mathcal{Y} = \mathbb{R}^N$ (Schwefel, 1987), the $(\mu + \lambda)$ selection should be used in discrete finite size search spaces, e.g., in combinatorial optimization problems (Herdy, 1990; Beyer, 1992).

3.3 *Mutation*

3.3.1 *General aspects*

The mutation operator is usually a basic variation operator in ES. That is, it is the primary source of genetic variation.[5] The design of mutation operators is problem-dependent. While there is not an established design methodology up to now, some rules have been proposed (Beyer, 2001b) by analyzing successful ES implementations and by theoretical considerations:

1. reachability,
2. unbiasedness,
3. scalability.

3.3.1.1 *Reachability.* Given a parental state $(\mathbf{y}_\mathrm{p}, \mathbf{s}_\mathrm{p})$, the first requirement ensures that any other (finite) state $(\tilde{\mathbf{y}}, \tilde{\mathbf{s}})$ can be reached within a finite number of mutation steps or generations. This is also a necessary condition for proving global convergence.

3.3.1.2 *Unbiasedness.* This requirement can be derived from our "philosophy of Darwinian evolution." Selection and variation serve different and somehow antagonistic purposes: Selection *exploits* the fitness information in order to guide the search into promising search space regions, whereas variation *explores* the search space, i.e., it should not use any fitness information but the search space information from the parental population. Therefore, there is no preference of any of the selected individuals (parents) in ES, and the variation operators should not introduce any bias. Being maximally unbiased, given the parental state(s), is therefore a design criterion for variation operators. Following this road we end up in a natural manner at the *maximum entropy principle* (for an introduction, see e.g. Jaynes, 1979). The application of this principle leads immediately to the normal distribution (Gaussian distribution) in the case of unconstrained real-valued search spaces $\mathbb{R}^N$. In unconstrained integer search spaces $\mathbb{Z}^N$ Rudolph (1994) has shown that this principle suggests the geometrical distribution. Other cases have not been investigated so far.

3.3.1.3 *Scalability.* The *scalability requirement* states that the mutation strength[6] or the average length of a mutation step should be tunable in order to adapt to the properties of the fitness landscape. The goal of adaptation is to ensure the *evolvability* of the "ES system," i.e., of the ES algorithm in conjunction with the objective function. Here, we use the term "evolvability" in an intuitive and informal manner expressing the idea that the variations should be generated in such a way that improvement steps are likely, thus building a "smooth" evolutionary random path through the fitness landscape toward the optimum solution (Altenberg, 1994). Since the properties of the fitness landscape are determined by the objective function *and* the variation operators, "smoothness" of the fitness landscape (but not necessarily of the objective function) may be regarded as a prerequisite of efficient evolutionary optimization. The smoothness assumption is sometimes expressed in terms of the (strong) *causality* concept (Rechenberg, 1994) stating that small changes on the genetic level should result on average in small changes in the fitness values (for a formalization of this concept see also Sendhoff et al., 1997).

Since there is no general rule for ensuring evolvability *independently* of the optimization problem at hand, we fall back to the minimal requirement of scalability which can be guaranteed easily in real-valued search spaces. However, even this requirement can be difficult to meet, e.g., for combinatorial search problems where there is always a "minimal move step" defining the limits of scalability (for example, in symmetric traveling salesman problems this is the so-called Lin-2-Opt step, see Lin and Kernighan, 1973 and below).

Even though the design principles given may serve as guiding lines, it should be mentioned that – depending on the optimization problem considered – their importance can differ and a mutation operator violating these requirements does *not* necessarily fail in a specific application.

3.3.2 *Examples*
In what follows we present standard mutation operators for *object* parameters (i.e., the operator in line # 10, Figure 1), the operators for mutating endogenous strategy parameters are introduced in section 4.2.

3.3.2.1 *Real-valued search spaces.* Considering the $\mathbb{R}^N$ search space and given the standard deviation σ (mutation strength) as the only endogenous strategy parameter $\mathbf{s}$, the maximum entropy principle yields

$$\tilde{\mathbf{y}} := \mathbf{y} + \mathbf{z}, \tag{5}$$

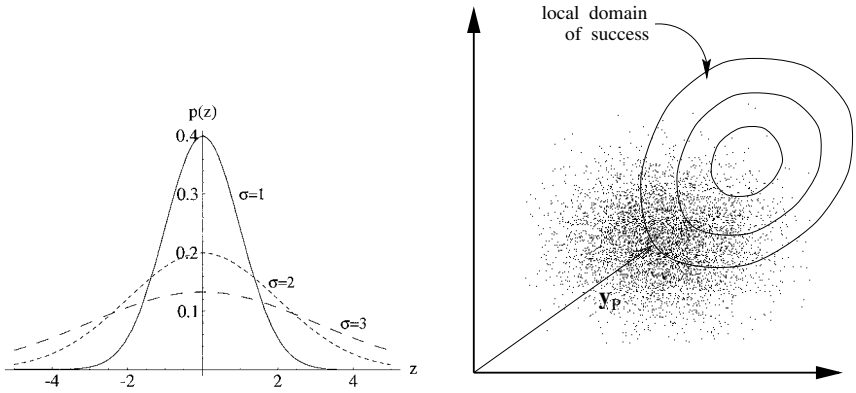
with

$$\mathbf{z} := \sigma(\mathcal{N}_1(0,1), \ldots, \mathcal{N}_N(0,1)), \tag{6}$$

where the $\mathcal{N}_i(0,1)$ are independent random samples from the standard normal distribution. That is, each component $\tilde{y}_i$ of the mutant $\tilde{\mathbf{y}}$ obeys the density function

$$p(\tilde{y}_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\frac{(\tilde{y}_i - y_i)^2}{\sigma^2}\right). \tag{7}$$

Figure 2a shows the effect of a single component mutation z. As one can see, the mutation operator prefers small changes depending on the mutation strength σ chosen. Due to its symmetry it introduces no bias: the expected value is zero and the mutants $\tilde{y}_i$ are distributed symmetrically around the parental state. This also transfers to the N-dimensional case. Furthermore, as one can infer from Figure 2b the samples are isotropically distributed around the parental state $\mathbf{y}_\mathrm{p}$. This can easily be shown by considering the joint density



a) 1-D $p(z)$ density curve b) 2-D mutation samples

Figure 2. (a) displays density functions of a single mutation component $z = \tilde{y}_i - y_i$ using different mutation strengths σ.
(b) depicts the shape of an isotropic mutation cloud of density $p(z_1)p(z_2)$ in a 2-dimensional search space. The three closed curves are lines of constant fitness, mutations located in the "local domain of success" represent fitness improvements.

distribution $p(\mathbf{z}) = \prod_{i=1}^{N} p_i(z_i)$: Since it is assumed that $p_i(z_i) = p(\tilde{y}_i - y_i)$, the surface of constant density builds concentric spheres around the parental state $\mathbf{y}$.

The isotropic mutation operator defined by equations (5) and (6) has the advantage that it needs only one endogenous strategy parameter for its control (as for the adaptation of this parameter, see section 4). However, there are situations where it can be beneficial to have mutation vectors those surfaces of constant density are ellipsoidal

$$\mathbf{z} := (\sigma_1 \mathcal{N}_1(0,1), \ldots, \sigma_N \mathcal{N}_N(0,1)). \tag{8}$$

Such a situation is depicted in Figure 3.

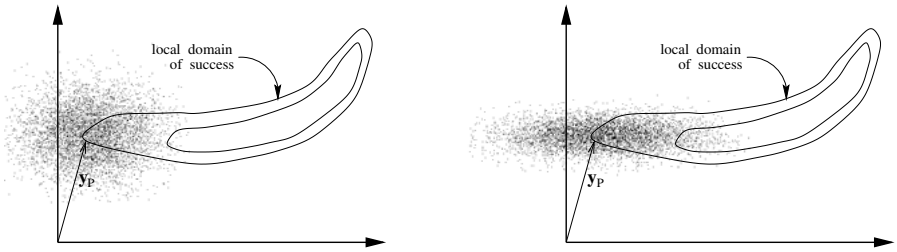


Figure 3. Simple situation of a fitness landscape where it is beneficial to use non-isotropic Gaussian mutations (right-hand side) instead of isotropic mutations (left-hand side).

It shows the simplest case of an axes-parallel mutation ellipsoid. The set of endogenous strategy parameters associated with comprises a vector of N

standard deviation parameters $\mathbf{s} := (\sigma_1, \ldots, \sigma_N)$ to be adapted appropriately. The most general situation is given when the ellipsoid is arbitrarily rotated in the search space (see Figure 4).

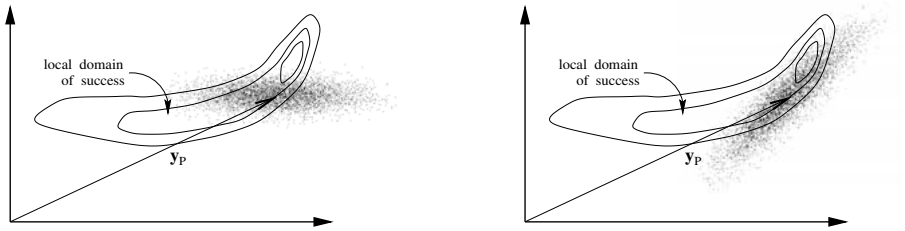


Figure 4. Situation of a fitness landscape where it is beneficial to have correlated Gaussian mutations (right-hand side) instead of non-correlated (left-hand side).

$$\mathbf{z} := \mathbf{M}\,(\sigma_1 \mathcal{N}_1(0,1), \ldots, \sigma_N \mathcal{N}_N(0,1))', \tag{9}$$

where $\mathbf{M}$ is a (orthgonal) rotation matrix. In this most general situation, the matrix $\mathbf{M}$ introduces *correlations* between the components of $\mathbf{z}$. That is, the correlation matrix $\mathbf{C} = \mathbf{M}'\mathbf{M}$ is nondiagonal and the density reads

$$p(\mathbf{z}) = \frac{1}{\left(\sqrt{2\pi}\right)^N} \frac{1}{\sqrt{\det[\mathbf{C}]}} \exp\left(-\frac{1}{2}\,\mathbf{z}'\mathbf{C}^{-1}\mathbf{z}\right). \tag{10}$$

Using $\mathbf{M}$ in (9) introduces $N(N-1)/2$ additional parameters. Thus, the whole mutation operator (9) comprises $N(N+1)/2$ strategy parameters to be controlled accordingly.

3.3.2.2 *Binary search spaces.* In binary search spaces $\mathcal{Y} \subseteq \mathbb{B}^N$, mutation is realized by random bit-flips of the binary parent vector $\mathbf{y}$. Unbiasedness is accomplished here by "microscopic reversibility", i.e., the probability p_m of flipping a 1-bit is equal to the opposite process. The *mutation rate* p_m can be considered as the pendant to the mutation strength.

In the simplest case, each bit position in $\mathbf{y}$ is mutated independently. In such situations there is empirical as well as some theoretical evidence for choosing $p_m = 1/N$ (Bäck, 1996; plus selection should be used). The mutation rate p_m can be controlled, however, its usefulness in practical applications has not been demonstrated satisfactorily up until now. Only recently Jansen and Wegener (2000) presented a proof that a periodically cycling $p_m = f(g)$ with $1/N \leq p_m \leq 1/2$ can improve the performance of a $(1+1)$-ES on a very synthetic (specially designed) pseudo-boolean fitness function.

The inclusion of bitwise and higher-order correlations would be the next logical step marked out by the experiences gathered from real-valued ES applications. Interestingly, only recently Mühlenbein and Mahnig (1999), who propose the so-called *factorized distribution algorithm*, as well as Pelikan et al. (1999), proposing the so-called *Bayesian optimization algorithm*, followed this road indicated long before by former ES results.

3.3.2.3 *Combinatorial search spaces.* The realm of combinatorial optimization tasks is extremely versatile. We will restrict our presentation to simple ordering problems where the fitness of an individual is determined by the specific order of its components. Different objective states are represented by the permutation order of these components. A typical example is the traveling salesman problem (TSP). Mutations are realized by permutation of the parental order. This can be done in different ways. Figure 5 presents a collection of *elementary move steps*. These elementary search step operators define certain *neighborhoods*, i.e., numbers of states which can be reached from the parental state within one step.

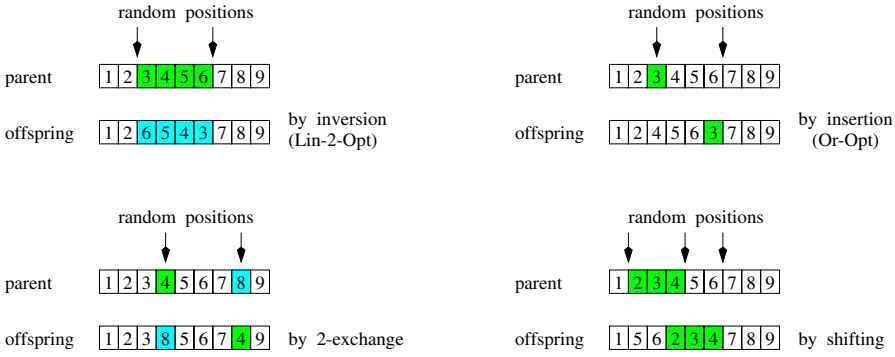


Figure 5. The effect of the four mostly used elementary (per-) mutation operators for ordering problems. Each numbered cell indicates an object component. The operators change this ordering by randomly cutting the connections between the cells and exchanging the cells in a specific way. Top left: "inversion" (also known as "Lin-2-Opt"), top right: "insertion" (also known as "Or-Opt"), bottom left: "2-exchange" (also known as "reciprocal exchange"), bottom right: "shifting" (also known as "displacement").

The neighborhood size can be regarded as the pendant to the mutation strength in real-valued search spaces. Since the neighborhood size is usually constant, the scalability requirement of the mutation steps cannot be guaranteed. There is a rudimentary way of realizing scalability by generating more complex mutations through the repeated application of, say s, elementary move steps. Thus, s becomes an endogenous strategy parameter which can be regarded as "mutation strength." However, the mutation strength depending

operator realized in this way has only restricted scaling behavior. Its mutation strength is bounded from above by $\mathcal{O}(N)$, i.e., on average it takes only $s < \text{const.} \cdot N$ elementary steps to jump through the whole search space. More critically, usually the mutation strength s cannot be smaller than one. Due to the fixed neighborhood size, there are no "fractional" move steps allowing for a successive and evolutionary smooth path to a global optimum state. An *ad hoc* approach for realizing "fractional" move steps may be the artificial restriction of the neighborhood size using optimization problem specific information taking strong causality properties into account (Herdy, 1990).

3.4 *Recombination*

3.4.1 *Standard ES recombination operators*

While mutation performs search steps based on the information of only one parent and – if applicable – on its endogenous strategy parameters, recombination shares the information from up to ρ parent individuals (Schwefel, 1975; Rechenberg, 1978; Rechenberg, 1994). If $\rho > 2$, we speak of multi-recombination. Unlike standard crossover in GA (see, e.g., Goldberg, 1989) where two parents produce two offspring, the application of the standard ES recombination operator to a parent family of size ρ produces only one offspring (see Figure 6).

There are two standard classes of recombination used in ES: "discrete recombination" sometimes referred to as "dominant recombination" and the "intermediate recombination."

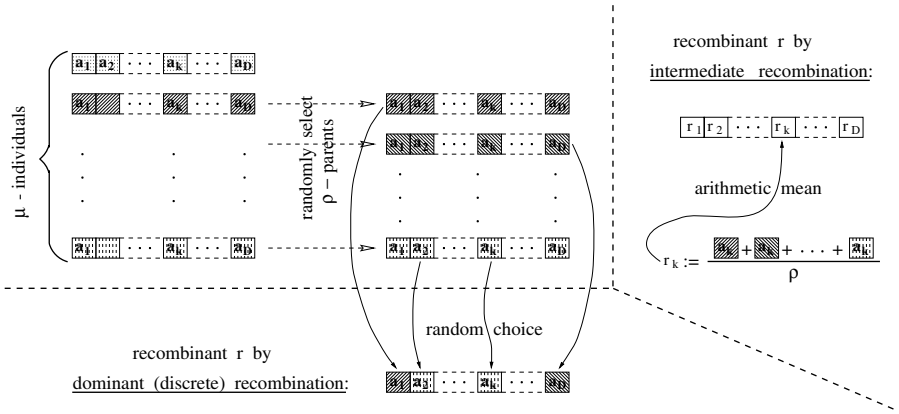


Figure 6. Standard μ/ρ (multi-) recombination operators in ES. The upper left part of the figure pictures the mating process as realized by the marriage operator in line # 6, Figure 1. It generates the parent family of size ρ. Intermediate ρ recombination is displayed to the right and the dominant ρ recombination is at the bottom.

In Figure 6 both versions are depicted. Given a parental vector $\mathbf{a} = (a_1, \ldots, a_D)$ (object or strategy parameter vector), the dominant ρ recombination produces a recombinant $\mathbf{r} = (r_1, \ldots, r_D)$ by coordinate-wise random selection from the ρ corresponding coordinate values of the parent family

$$(\mathbf{r})_k := (\mathbf{a}_{m_k})_k, \quad \text{with} \quad m_k := \text{Random}\{1, \ldots, \rho\}. \tag{11}$$

In other words, the kth component of the recombinant is determined exclusively by the kth component of the randomly (uniformly) chosen parent individual m_k that can also be interpreted as "dominance."

In contrast to discrete (dominant) recombination the intermediate recombination takes all ρ parents equally into account. It simply calculates the center of mass (centroid) of the ρ parent vectors $\mathbf{a}_m$.

$$(\mathbf{r})_k := \frac{1}{\rho} \sum_{m=1}^{\rho} (\mathbf{a}_m)_k. \tag{12}$$

While this procedure is well defined for real-valued state spaces, the application in discrete spaces needs an additional procedure such as rounding or probabilistic rounding in order to map back onto the discrete domain.

3.4.2 *On the benefits of recombination and design principles*

3.4.2.1 *The building block hypothesis.* There is still an ongoing debate as to the usefulness and functioning principles of recombination. In GA research the *building block hypothesis* (BBH) (Goldberg, 1989; Holland, 1995) has been suggested in order to explain the working mechanism of crossover: different good building blocks from different parents are mixed together, thus combining the good properties of the parents in the offspring. While this explanation is intuitively appealing, finding test functions to support this hypothesis appeared surprisingly difficult (Mitchell et al., 1994). Only recently Jansen and Wegener (2001) were able to construct an artificial test function where one-point crossover essentially improves the time complexity, thus providing a clue toward the usefulness of the BBH in that special situation.

3.4.2.2 *Genetic repair and similarity extraction.* During the 1990s, ES theory research has revealed another interesting functioning mechanism of recombination – the *genetic repair* (GR) effect (Beyer, 1995), giving rise to the GR hypothesis (Beyer, 1997). This hypothesis is in counterposition to the BBH: Not the different (desired) features of the different parents flow through the application of the recombination operator into the offspring, but their *common* features. That is, recombination extracts the *similarities* from

the parents. This becomes immediately evident when considering two-parent recombination ($\rho = 2$) in $\mathbb{B}^N$. Bit positions common to both parents are conserved by the standard recombination operators, whereas the other positions are randomly chosen from the parents. That is, considering dominant $\rho = 2$ recombination as well as *uniform crossover* (as used in GA, see Syswerda, 1989), these non-common positions are randomly filled up. Such a mechanism is somehow reminiscent of *self-adaptation* (see also section 4.2): It is reasonable to assume that corresponding components in the parental genomes which are similar to each other carry a higher probability of being beneficial with respect to the individual's fitness (because they are in the fittest, i.e. selected, individuals). The best a variation operator can do (from the viewpoint of maximum entropy) is to conserve these components. The other components are more or less irrelevant or they even may be harmful. Therefore, they can be mutated or randomly rearranged.

The performance gain by similarity extraction and the GR effect become especially evident when considering (μ/μ)-multirecombination of intermediate type in $\mathbb{R}^N$ search spaces using isotropic mutations. The object parameter update rule of a $(\mu/\mu, \lambda)$-ES with this kind of recombination can be expressed very concisely since all mutations $\mathbf{z}$ are applied to the parental centroid $\langle \mathbf{y} \rangle$

$$\langle \mathbf{y} \rangle := \frac{1}{\mu} \sum_{m=1}^{\mu} \tilde{\mathbf{y}}_{m;\lambda}. \tag{13}$$

Using the notations from equations (4), (5), and (12), we have

$$\langle \mathbf{y} \rangle^{(g+1)} := \langle \mathbf{y} \rangle + \frac{1}{\mu} \sum_{m=1}^{\mu} \mathbf{z}_{m;\lambda}. \tag{14}$$

That is, the new parental centroid is obtained from the old one by adding the vector average $\langle \mathbf{z} \rangle$ of the μ best mutations.

In order to visually explain the GR mechanism, we decompose each $\mathbf{z}$ mutation into a component toward the desired improvement direction (usually the direction to the optimum) $x\mathbf{e}_{\text{opt}}$ and a perpendicular part $\mathbf{h}$ representing the harmful components driving the offspring away from the improvement direction.

$$\mathbf{z} = x\mathbf{e}_{\text{opt}} + \mathbf{h}, \quad \text{with} \quad \mathbf{h}'\mathbf{e}_{\text{opt}} = 0. \tag{15}$$

The insertion of (15) into (14) yields

$$\begin{aligned}\langle \mathbf{y} \rangle^{(g+1)} &= \langle \mathbf{y} \rangle + \langle \mathbf{z} \rangle = \langle \mathbf{y} \rangle + \mathbf{e}_{\text{opt}} \frac{1}{\mu} \sum_{m=1}^{\mu} x_{m;\lambda} + \frac{1}{\mu} \sum_{m=1}^{\mu} \mathbf{h}_{m;\lambda} \\ &= \langle \mathbf{y} \rangle + \langle x \rangle \mathbf{e}_{\text{opt}} + \langle \mathbf{h} \rangle.\end{aligned}$$

This decomposition is depicted in Figure 7.

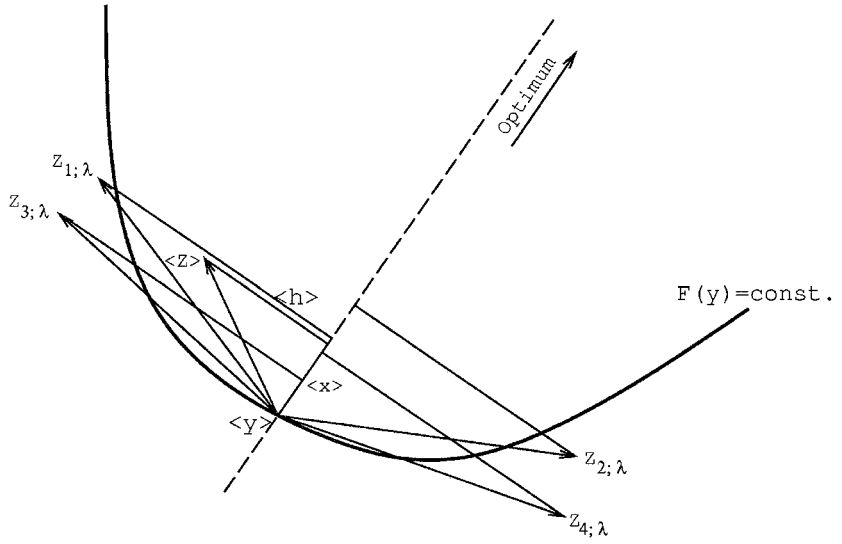


Figure 7. On the visualization of the genetic repair effect in a real-valued search space. The bold curve represents the line of constant fitness $F(\mathbf{y}) = F(\langle \mathbf{y} \rangle)$. Better fitness values are located above this curve. Although even the four best mutants are worse than the parental centroid $\langle \mathbf{y} \rangle$, after recombination the new centroid $\langle \mathbf{y} \rangle + \langle \mathbf{z} \rangle$ yields better fitness.

Due to selection, the x components of the best mutations have a certain (positive) tendency toward the improvement direction $\mathbf{e}_{\text{opt}}$.[7] However, the $\mathbf{h}$ components are almost isotropically distributed in the subspace orthogonal to $\mathbf{e}_{\text{opt}}$. This has been expressed in Figure 7 by drawing the best four $\mathbf{z}$ vectors almost symmetrically. Now taking the average in order to determine $\langle \mathbf{z} \rangle$, one can see that the improvement component is more or less conserved. Its value $\langle x \rangle$ is within the bounds $\min\{x_{1;\lambda}, \ldots, x_{\mu;\lambda}\} \leq \langle x \rangle \leq \max\{x_{1;\lambda}, \ldots, x_{\mu;\lambda}\}$. As to the $\mathbf{h}$ components, averaging has a beneficial effect: it reduces the length, it "genetically repairs" the harmful parts. This is so because of the isotropy and therefore statistical independence of the $\mathbf{h}$ vectors. Put it another way, the $\mathbf{h}$ vectors are almost uncorrelated. As has been shown (Beyer, 1995), assuming statistical independence of the $\mathbf{h}$ components, the expected length of $\langle \mathbf{h} \rangle$ can be up to a factor of $\sqrt{\mu}$ smaller than the expected length of a single

h vector. This is the performance increasing effect of genetic repair, it reduces the fitness decreasing effect of the harmful parts of the mutants, thus, also allowing for a larger mutation strength and therefore a larger improvement step. To summarize, by conserving the beneficial parts of the mutations and dampening their harmful parts through genetic repair, intermediate recombination provides an efficient mechanism for similarity extraction in real-valued search spaces.

While the effect of similarity extraction in real-valued as well as binary search spaces is quite obvious, the derivation of design rules for recombination operators for combinatorial optimization problems is still a challenge. Similarity must be defined in a problem specific manner *and* the outcome of recombination should be a visible individual (otherwise a special repair procedure must be applied that might violate the similarity condition). Concerning symmetric TSPs, an example which adheres to the design principles just discussed can be found in Herdy (1990). Similar design criteria for the breeding of programs are presented in Droste and Wiesmann (2000).

4. Adaptation of endogenous strategy parameters

This section is devoted to the adaptation of strategy parameters controlling the statistical properties of the variation operators, especially of the mutation operators. In section 4.1 we will present the motivation and the famous *1/5th*-rule for controlling the mutation strength in $(1 + 1)$-ES. The second section will introduce into the ideas of self-adaptation – the standard evolutionary adaptation technique based on (temporo-) local population information. Section 4.3 gives a short introduction into advanced adaptation techniques using nonlocal search space information.

4.1 *Motivation and the* 1/5th-*rule*

4.1.1 *The discovery of the "evolution window"*

The necessity of controlling endogenous strategy parameters becomes evident when running a simple $(1 + 1)$-ES with isotropic Gaussian mutations (6) and *constant* mutation strength σ on a simple objective function $F(\mathbf{y})$. Such a minimization run, using a sphere model $F_{\mathrm{sp}}(\mathbf{y})$

$$F(\mathbf{y}) = F_{\mathrm{sp}}(\mathbf{y}) = c\|\mathbf{y}\|^{\alpha}, \quad \mathbf{y} \in \mathbb{R}^N \tag{16}$$

as objective function with $c = 1$, $\alpha = 2$, $N = 10$, is depicted in Figure 8. A constant mutation strength $\sigma = 1$ has been chosen and the parent at generation $g = 0$ was initialized at $y_k^{(0)} = 10$.

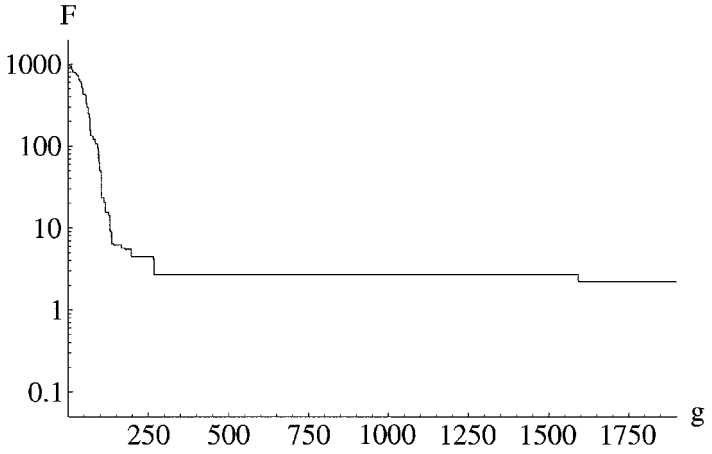
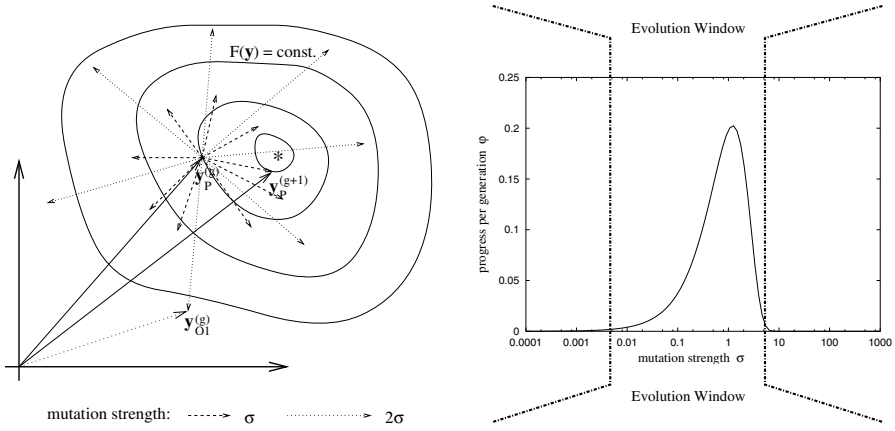


Figure 8. Evolutionary dynamics of a $(1 + 1)$-ES minimizing a sphere model. The mutation operator uses isotropic Gaussian mutations with *constant* mutation strength $\sigma = 1$ throughout the whole ES run. After a period of good performance the ES has lost its evolvability. Therefore, the mutation strength must be adapted appropriately.

Even though one can show that this strategy is globally convergent (Rechenberg, 1973), one sees that after a period of improvements the strategy becomes extremely slow – the system has lost its evolvability. This is due to the fixed mutation strength σ.

In order to get an intuitive feeling how σ influences the local performance of the ES in real-valued search spaces, Figure 9a presents a view in an $\mathbb{R}^2$ search space with a fitness landscape depicted by lines of constant $F(\mathbf{y})$ values. The optimum is located at the "$*$"-position. Since we consider a $(1 + 1)$-ES, there is only one parental state $\mathbf{y}_\mathrm{p}$ at generation (g) from which offspring are generated by isotropic mutations. Because of equation 6, the expected length of the mutations is proportional to σ. If σ is chosen very small, on average every second mutation will hit the local success domain, which is defined by the interior of the curve $F(\mathbf{y}) = F(\mathbf{y}_\mathrm{p}) = \text{const.}$, and will be accepted as the parent of generation $(g + 1)$. This is so because of the smoothness of the fitness landscape and the symmetry of the mutations used. In other words, using a very small mutation strength ensures a high degree of evolvability. The *success probability* P_s by which an offspring replaces a parent becomes $P_\mathrm{s} \to 1/2$. However, since the improvement steps scale with σ, progress toward the optimum point scales with σ, too. The measure describing this expected local approach to the optimum is called the *progress rate* φ (see Figure 12, section 5.2.2, below). Summarizing, we have for smooth fitness landscapes

$$\sigma \to 0: \qquad P_\mathrm{s} \to 1/2, \quad \text{but} \quad \varphi \to 0. \tag{17}$$



a) Mutations in an $\mathbb{R}^2$ search space. b) Progress rate and "evolution window."

Figure 9. On the influence of the mutation strength σ on the evolvability and performance of the ES system. As one can infer from a), using (very) small mutations ensures a high degree of evolvability. However, b) reveals an "evolution window," i.e., performance φ of the ES depends sensitively on the mutation strength chosen.

That is, even though we have a high degree of evolvability for sufficiently small mutation strengths, the performance of the ES will be rather poor.

Now, let us consider the opposite case choosing σ very large. The mutations produced from the parental state $\mathbf{y}_\mathrm{p}$ are too large such that hitting the local success domain becomes a rare event. Therefore $P_\mathrm{s} \to 0$, the evolvability changes to the worse, and also the expected progress φ will be small due to the smallness of P_s

$$\sigma \to \infty : \qquad P_\mathrm{s} \to 0 \quad \text{and} \quad \varphi \to 0. \tag{18}$$

Between the two extremes (17) and (18) there is a bandwidth of mutation strengths σ guaranteeing nearly optimal φ performance – the "evolution window," a term which has been coined by Rechenberg (1973). As can be seen in Figure 9b, choosing the right mutation strength σ allows for maximizing the performance.

4.1.2 *The* 1/5th-*rule*

Based on the fact that both the performance φ and the success probability P_s depend on σ, a σ control rule can be envisaged. Figure 10 displays P_s and the normalized progress rate φ^* calculated for the sphere model (16) in the asymptotic limit case $N \to \infty$ (Rechenberg, 1973; Beyer, 2001b). As we can see, for the model considered, the progress rate is a function of the success probability $\varphi^* = f(P_\mathrm{s})$. There is an optimal success probability $\hat{P}_\mathrm{s} \approx 0.27$

maximizing the progress rate. An investigation of the totally different model function "corridor" (a linear objective function with $2(N-1)$ inequality constraints) provided a surprisingly similar result $\hat{P}_s \approx 0.184$ (Rechenberg 1973, p. 122). Since both models have been regarded as somehow "typical" models of real objective functions, a compromise success probability of 0.2 was proposed by Rechenberg – leading to the *1/5th*-rule:

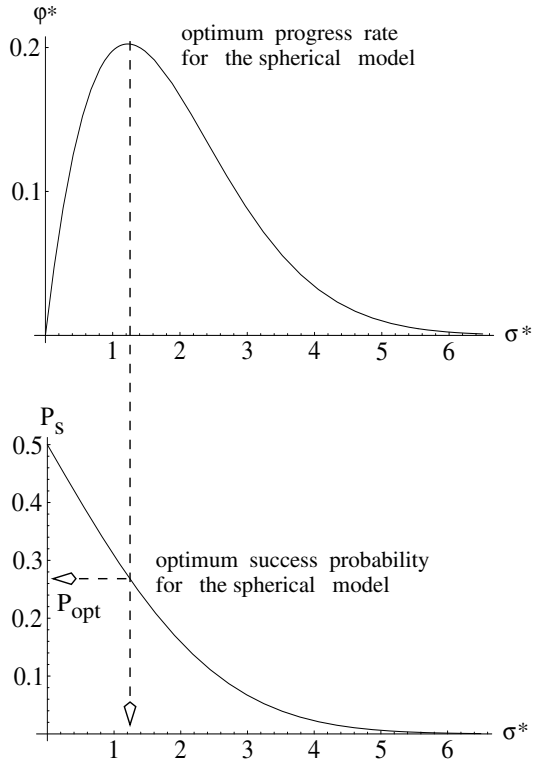


Figure 10. On the dependency of the normalized progress rate φ^* on the success probability P_s for the sphere model (16).

> *In order to obtain nearly optimal (local) performance of the* $(1+1)$*-ES in real-valued search spaces, tune the mutation strength in such a way that the (measured) success rate is about 1/5.*

Due to the monotonicity of P_s with respect to the normalized σ (denoted as $\sigma^* := \sigma N/R$, cf. Figure 10), the tuning of σ can be accomplished easily: If $P_s < 1/5$ the mutation strength must be reduced, whereas in the opposite case $P_s > 1/5$, σ must be increased.

The implementation of the *1/5th*-rule does not fully fit into the algorithmic skeleton of Figure 1, therefore we present a short description in Figure 11.

1. perform the $(1 + 1)$-ES for a number G of generations:
   - keep σ constant during this period,
   - count the number G_s of successful mutations during this period
2. determine an estimate of P_s by

$$P_s := G_s/G \tag{19}$$

3. change σ according to

$$\sigma := \begin{cases} \sigma/a, & \text{if} \quad P_s > 1/5 \\ \sigma \cdot a, & \text{if} \quad P_s < 1/5 \\ \sigma, & \text{if} \quad P_s = 1/5 \end{cases} \tag{20}$$

4. goto 1.

Figure 11. On the implementation of the *1/5th*-rule.

The changing of σ is done multiplicatively. The optimal value of the factor a depends on the objective function to be optimized, the dimensionality N of the search space, and on the number G. If N is sufficiently large $N \geq 30$, $G = N$ is a reasonable choice. Under this condition Schwefel (1975) recommended using $0.85 \leq a < 1$.

4.2 *Self-adaptation*

In section 4.1 we presented the motivation for tuning the endogenous strategy parameter σ in order to get nearly (local) optimal performance of the $(1 + 1)$-ES in real-valued search spaces. While the arguments for adapting σ are of general interest, the adaptation rule presented there, the *1/5th*-rule, was a very special one: the control heuristic represents the simplest version of deterministic nonlocal adaptation. It uses the *global* information of success probability, to be obtained by collecting statistical data over a number G of generations, in order to deterministically change the mutation strength. It is quite clear that the *1/5th*-rule has its specific application domain:

- it is restricted to the application of one strategy parameter,
- the fitness landscape must obey certain properties in order to ensure that for sufficiently small mutation strengths $P_s > 1/5$ (otherwise one may observe premature stagnation, see Schwefel, 1995, pp. 116f),
- it is usually used in $(1 + 1)$-ES only (other strategies have different optimal P_s).

In this section we present a more flexible and *evolutionary*, i.e. temporo-local, control method which can be regarded as the standard approach in ES – the self-adaptation. The basic idea of self-adaptation will be explained in the next section, implementation details are given in section 4.2.2

4.2.1 *The basic idea*
The principal idea of self-adaptation in ES rests on the *individual* coupling of endogenous (evolvable) strategy parameters with the object parameters. That is, as already introduced in equation (2), each individual $\mathfrak{a}$ has its own set of strategy parameters. These endogenous strategy parameters are subject to variation. This has been taken into account in the pseudo-code of the $(\mu/\rho \overset{+}{,} \lambda)$-ES, Figure 1. The strategy parameters may undergo recombination and are (always) subject to mutation. The mutated strategy parameters are used then to control the mutation operator applied to the individual's object parameters. Due to the inclusion of the strategy parameters in the individual's genome, they are selected and inherited *together* with the individual's object parameters. Thus, they have a higher probability of survival when they "encode" object parameter variations that produce (on average) fitter object parameters. Following these plausibility arguments, one can hope that the individuals will learn the (nearly) optimal strategy parameters during the evolution process.

4.2.2 *Implementation details*
The concrete realization of self-adaptive ES depends on the kind of strategy parameters to be adapted. In what follows we consider the adaptation of a single mutation strength parameter σ in detail. The generalization to more than one mutation parameter will be discussed briefly. Concerning other strategy parameter types as well as self-adaptive behavior in real-coded GA the reader is referred to Beyer and Deb (2001) and the references listed there.

4.2.2.1 *How to mutate the mutation strength σ.* The design principles of section 3.3 should also hold for the strategy parameter operators. However, practical experience shows that some of the principles can be violated to a certain extent without severe consequences. Probably the most critical requirement concerns the scalability in $\mathbb{R}^N$ search spaces.

In order to ensure scalability in $\mathbb{R}^N$ the mutation of the mutation strength must be performed *multiplicatively*. There are at least two reasons. First, σ represents the standard deviation of the mutation distribution, this is per definition positive. An additive mutation operator cannot guarantee positiveness, there is always a certain probability for mutations leading to negative σ. Secondly, by plausibility arguments considering the sphere model (for a more detailed discussion see Beyer, 2001b, p. 260). It is quite clear that there

exists always an optimal mutation strength $\hat{\sigma}$ given a parental state $\mathbf{y}_\mathrm{p}$. Due to the symmetry of the sphere model, $\hat{\sigma}$ depends only on the parental distance r_p to the optimum. If the state space and the mutations are scaled by a constant factor, say κ, both $\hat{\sigma}$ and r_p are scaled in the same way. Thus, we have $\hat{\sigma} \propto r$, i.e., $\hat{\sigma}/r = \text{const}$. Now, consider two consecutive generations, it follows

$$\hat{\sigma}^{(g)}/r^{(g)} = \hat{\sigma}^{(g+1)}/r^{(g+1)} \quad \Rightarrow \quad \hat{\sigma}^{(g+1)} = \hat{\sigma}^{(g)}\frac{r^{(g+1)}}{r^{(g)}}. \tag{21}$$

Under stationary conditions, i.e., correctly working self-adaptation, the expected relative r change should be constant (scaling behavior!). Therefore, in expectation σ should change by a constant factor.

Using the arguments from above, the **s** mutation operator in Figure 1, line # 9, can be expressed as

$$\tilde{\sigma}_l := \sigma_l \, \exp(\zeta_l), \tag{22}$$

where ζ_l is a sample from a random number source to be specified below. The use of the exponential function in (22) is, however, mainly for cosmetic reasons: taking the natural logarithm on both sides of (22), one gets $\ln(\tilde{\sigma}_l) := \ln(\sigma_l) + \zeta_l$. On a logarithmic scale the strategy mutations are performed in the same manner as the mutations of the object parameters. It is therefore reasonable to sample ζ from a normal distribution (Schwefel, 1975). The mutation operator realized in this way is also referred to as "log-normal operator" because of the logarithmic normal $\tilde{\sigma}$ distribution generated

$$\tilde{\sigma}_l := \sigma_l \, \exp(\tau \mathcal{N}_l(0, 1)). \tag{23}$$

The τ in (23) is the so-called learning parameter. This *exogenous* strategy parameter determines the rate and precision of self-adaptation. Theoretical as well as empirical investigations suggest that τ should be chosen according to (Schwefel, 1975; Beyer, 1996)

$$\tau \propto \frac{1}{\sqrt{N}} \tag{24}$$

(as a first guess, $\tau = 1/\sqrt{N}$ may be chosen; in highly multimodal fitness landscapes smaller learning rates, such as $\tau = 1/\sqrt{2N}$, should be tried).

It might appear as a little surprise, but $\zeta \sim \tau \mathcal{N}(0, 1)$ is not the only choice possible. One may even violate the reachability requirement of section 3.3.1 and can still obtain a satisfactorily working self-adaptive ES. Rechenberg's

(1994) "two-point rule" is of such a kind. It simply flips a coin in order to choose $\zeta = \pm\epsilon$. Writing $\epsilon = \ln(\alpha)$, the two-point rule reads

$$\tilde{\sigma}_l := \begin{cases} \sigma_l\, \alpha, & \text{if } \mathrm{u}(0,1] \leq \frac{1}{2} \\ \sigma_l/\alpha, & \text{if } \mathrm{u}(0,1] > \frac{1}{2} \end{cases} \qquad \alpha > 1, \tag{25}$$

where $\mathrm{u}(0, 1]$ is a uniform random source and α a learning parameter. As has been shown (Beyer, 1996), choosing $\alpha = 1 + \tau$ provides nearly optimal performance on the sphere model.

4.2.2.2 *Mutation rules for a set of strategy parameters.* The multiplicative σ mutation technique can be extended to the case where we have a vector of strategy parameters $\mathbf{s} = \sigma = (\sigma_1, \ldots, \sigma_N)$ as needed in equation (8). Schwefel (1977) suggests using an extended log-normal rule which reads (suppressing the offspring index)

$$\begin{aligned} \tilde{\sigma} \;:=\; & \exp(\tau_0 \mathcal{N}_0(0,1)) \cdot \Big(\sigma_1 \exp(\tau\, \mathcal{N}_1(0,1)),\ \ldots, \\ & \sigma_N \exp(\tau\, \mathcal{N}_N(0,1))\Big). \end{aligned} \tag{26}$$

The idea here is to have a general mutative multiplier with learning parameter τ_0 and coordinate-wise mutations with learning parameter τ. Each component of σ is mutated independently and finally the whole vector is mutatively scaled by the random factor $\exp(\tau_0 \mathcal{N}_0(0, 1))$. This technique allows for learning axes-parallel mutation ellipsoids. As learning parameters

$$\tau_0 = \frac{c}{\sqrt{2N}} \quad \text{and} \quad \tau = \frac{c}{\sqrt{2\sqrt{N}}} \tag{27}$$

are recommended ($c = 1$ is a reasonable choice for a $(10, 100)$-ES, see Schwefel, 1995, p. 388).

Going a step further, correlated mutations (9) could be considered next. However, this is beyond the scope of this introductory paper (see, e.g., Schwefel, 1995, Schwefel and Rudolph, 1995, Hansen and Ostermeier, 2001).

4.2.2.3 *How to recombine the endogenous strategy parameters.* As argued in section 3.4 one main effect of recombination is the extraction of similarities. With respect to strategy parameter adaptation, this is a desired effect. As observed in experiments, the evolutionary dynamics of the strategy parameters appears very often as a noisy process with large fluctuations. These fluctuations usually degrade the performance of the ES. Therefore, techniques

capable of reducing the fluctuations are needed. It is this the reason why *intermediate* (μ/μ) recombination is highly recommended.

There are also other techniques for reducing strategy parameter fluctuations, especially the method of *rescaled* mutations should be mentioned here (Rechenberg, 1994; Beyer, 2000).

4.3 *Nonlocal adaptation approaches*

4.3.1 *Motivation*

Self-adaptation on the level of the individuals as introduced in section 4.2 can fail. The reason for such possible failures is often a behavior that might be called "opportunism:" Evolution rewards short term success, i.e., if local progress and global progress are not positively correlated, evolution might choose those offspring states that finally end up in a local optimum or even may exhibit so-called premature convergence.

It should be clear that there is no general solution to these convergence problems, however, countermeasures may be taken in order to reduce the risk of such events in concrete situations. As to the adaptation of strategy parameters, e.g., "optimality" of a parameter chosen may be better assessed by comparing partially isolated evolution processes. This immediately leads to the idea of *nested* ES, i.e., competing populations by meta-ES with isolation time parameter γ (Herdy, 1992; Rechenberg, 1978; Rechenberg, 1994). This approach is clearly based upon *nonlocal* information (nonlocality with respect to time).

In this section we will focus on advanced techniques for strategy parameter adaptations which are based upon nonlocal information. By this we mean the collection of search space information gathered from more than one population either by considering a history or aggregation of (specially defined) state variables or by explicit use of "parallel" evolving (sub-) populations. The simplest of such techniques has already been discussed in detail – the *1/5th*-rule. Here we will shortly review the application of meta-ES. In a second section we will present the idea of *cumulative path-length control*.

4.3.2 *Meta-ES*

Meta-ES or synonymously "nested ES" or "hierarchically organized ES" are strategies to be described in terms of the

$$\left[\mu'/\rho' \stackrel{+}{,} \lambda' \; (\mu/\rho \stackrel{+}{,} \lambda)^{\gamma}\right]\text{-ES} \tag{28}$$

notation. There are μ' parental populations of $(\mu/\rho \stackrel{+}{,} \lambda)$ strategies producing λ' offspring $(\mu/\rho \stackrel{+}{,} \lambda)$ strategies which run without any communication a time period of γ generations. After γ generations, selection in the outer

brackets takes place in order to determine the best μ' of the $(\mu/\rho \overset{+}{,} \lambda)$ strategies building the basis for the next population of λ' offspring strategies. There are at least three distinctive application domains of such nested strategies:

1. As meta-heuristic search strategies for global optimization in – principally – simple search spaces. The inner strategies perform "local search" and the outer strategy performs search on the set of the local optima approximated by the inner strategies (Lohmann, 1992; Rechenberg, 1994).
2. Search spaces composed from different kinds of simple search spaces, e.g., mixed-integer search spaces and combinatorial optimization with continuous weights (neural network design). In such applications, the evolution of the structure is usually performed by the outer ES, while the weights are evolved by the inner ES (see, e.g., Yao, 1999).
3. Optimizing the performance of the inner ES through "breeding" by the outer ES.

The third application is of interest here. The outer ES is used to optimize the endogenous strategy parameters of the inner ES. In the simplest case, this can be done implicitly by a $[1, 2\ \ (\mu/\rho, \lambda)^1]$-ES without any isolation. The duty of the outer loop is simply to select the population with the better objective value of the new parental centroid (i.e., after selecting the offspring in the inner ES). This simple meta-ES is already able to adapt *optimal* mutation strengths in the case of (μ/μ) recombination, a problem where it is known that the performance of the ES depends sensitively on the learning parameter τ when using standard self-adaptation techniques (Grünz and Beyer, 1999).

In a more general case, adapting a larger set of endogenous strategy parameters will usually necessitate the use of isolation $\gamma > 1$. If the strategy parameter set is of order N, the isolation time should be chosen accordingly, $\gamma \propto N$. In this situation it can also be reasonable to keep the endogenous strategy parameters in the inner ES constant and perform their mutations in the outer ES.

4.3.3 *Cumulative path-length control – CSA-ES*

There is a class of *deterministic* nonlocal techniques for adapting the mutation operators in $\mathbb{R}^N$ the idea of which can be attributed to Ostermeier et al. (1994): the *cumulative path-length control*. In order to get an intuitive feeling of what is meant by this wording, let us consider a so-called evolution path $\mathbf{v}$, that is, – in the simplest case – the vector sum of the actually realized evolution steps over a number of G generations.

Provided that there were no selective pressure and $\sigma = \text{const.}$, we would have random selection resulting in

$$\mathbf{u} = \sum_{g=1}^{G} \mathbf{z}^{(g)} = \sigma \sum_{g=1}^{G} \mathcal{N}^{(g)}(\mathbf{0}, \mathbf{1}) \tag{29}$$

(writing $\mathbf{u}$ instead of $\mathbf{v}$) and the length u becomes

$$u = \sigma \sqrt{\sum_{k=1}^{N} \sum_{g=1}^{G} (\mathcal{N}_k^{(g)}(0,1))^2}. \tag{30}$$

There is a sum of $G \cdot N$ independent squared standard normal variates in (30), i.e., this sum is χ^2 distributed with degree GN and the expected length $\overline{u}$ is simply proportional to the expectation of χ

$$\overline{u} = \sigma \overline{\chi}_{NG} \simeq \sigma \sqrt{NG}. \tag{31}$$

Now, consider the influence of selection. If the mutation steps are too small, then on average selection will prefer larger steps, the expected value of the actual length v of the evolution path $\mathbf{v}$ will be larger than $\overline{u}$ and σ should be increased. If, however, the mutation steps are too large, the smaller steps will be selectively preferred. Therefore, the expected v will be smaller than $\overline{u}$ and σ should be decreased (see also Hansen and Ostermeier, 1996).

The basic idea presented is somehow reminiscent of the 1/5th-rule (20). The main difference is in the way how the nonlocal population information is used. While the 1/5th-rule discards any search space information, but simply counts the number of individuals hitting the local success domain, the evolution path related techniques elaborate search space information in a more advanced manner. This becomes evident when considering implementations of the basic idea such as the *cumulative step-size adaptation* (CSA) and the *covariance matrix adaptation* CMA (Hansen and Ostermeier, 2001). In equation (32) the standard implementation of the CSA-ES is presented

$$\left.\begin{aligned}
&\forall l = 1, \ldots, \lambda: \quad \mathbf{w}_l^{(g)} := \sigma^{(g)} \mathcal{N}_l(\mathbf{0}, \mathbf{1}), && \text{(a)}\\
&\mathbf{z}^{(g)} := \frac{1}{\mu} \sum_{m=1}^{\mu} \mathbf{w}_{m;\lambda}^{(g)}, && \text{(b)}\\
&\mathbf{y}^{(g+1)} := \mathbf{y}^{(g)} + \mathbf{z}^{(g)}, && \text{(c)}\\
&\mathbf{v}^{(g+1)} := (1-c)\mathbf{v}^{(g)} + \sqrt{c(2-c)}\frac{\sqrt{\mu}}{\sigma^{(g)}}\mathbf{z}^{(g)}, && \text{(d)}\\
&\sigma^{(g+1)} := \sigma^{(g)} \exp\left[\frac{\|\mathbf{v}^{(g+1)}\| - \overline{\chi}_N}{D\overline{\chi}_N}\right]. && \text{(e)}
\end{aligned}\right\} \tag{32}$$

The first three lines in (32) represent the standard $(\mu/\mu, \lambda)$-ES with intermediate recombination in object parameter space: produce λ normally distributed mutations $\mathbf{w}_l$ (a), recombine the μ fittest mutations by a centroid operation (b), and update the new parental state (c). The CSA is realized by cumulating the actual evolution path $\mathbf{v}$ (d) in a weighted manner. The mutation strength update (e) is accomplished by comparing the length of $\mathbf{v}$ with the expected length $\overline{\chi}_N$ of the N-dimensional $\mathcal{N}(0, 1)$ random vector.

As for all adaptation techniques, there are some exogenous strategy parameters to be fixed appropriately (Hansen and Ostermeier, 1997). The cumulation time parameter c, $0 \leq c \leq 1$, is usually chosen to be $c \propto 1/\sqrt{N}$, the damping parameter $D \propto \sqrt{N}$, and for $\overline{\chi}_N$ the approximation $\overline{\chi}_N \approx \sqrt{N}(1 - 1/4N + 1/21N^2)$ may be used.

The idea of cumulation can be extended to allow for the adaptation of covariance matrices $\mathbf{C}$ needed to produce correlated mutations (10). The description of the respective CMA-ES is beyond the scope of this paper, the reader is referred to Hansen and Ostermeier (2001).

5. Theoretical aspects

5.1 *Motivation*

It is often claimed in the EC community that the usefulness of theory in this field is rather questionable (see, e.g., Eiben et al., 1999, p. 126). However, stepping back and having a much wider perspective, it should be clear that each EA practitioner uses "some kind" of/or "his" theory. Theory in this sense should be understood as a collection of experiences or knowledge useful for making predictions on the behavior of the system considered. This includes, of course, mathematical theorems and their proofs, but it also comprises "softer knowledge" the correctness of which has not been proven or those validity might prove itself restricted to special cases with the progression of rigorous mathematical theory in future. From this perspective, most of the material presented in this paper so far belongs to the latter category. This does not necessarily exclude that there are certain clues from current (rigorous mathematical) theory. Some of the design principles presented in sections 3 and 4 can be regarded as *extrapolations* from a (mathematical) theory which considers the evolutionary algorithm *and* the fitness function as an *evolutionary system* obeying a specific *evolutionary dynamics*. From this point of view the goal of a mathematical theory of evolutionary algorithms (EA) is to *predict* the *EA-dynamics*.

This section is mainly devoted to quantitative aspects of the ES-dynamics, i.e., the mathematical description and investigation of the ES in the time

domain. Evolutionary algorithms are Markovian processes and as such they are governed by so-called Chapman-Kolmogorov equations, the generalization of Markov chains used in discrete search spaces (see, e.g., Fisz, 1971). These equations map the joint state density $p(\cdot)$ (or the probability vector in the discrete case) of the (parental) population $\mathfrak{P}_\mathrm{p}$ from one generation to the next. Using the notations from section 3 this reads

$$p^{(g)}(\mathfrak{P}_\mathrm{p}^{(g)}) \quad \mapsto \quad p^{(g+1)}(\mathfrak{P}_\mathrm{p}^{(g+1)}). \tag{33}$$

The general treatment of the stochastic mapping (33), i.e., the determination of the "full dynamics" is almost always excluded, except for the simplest cases. On the other side, the information provided by $p(\mathfrak{P}_\mathrm{p})$ is difficult to interpret. We are more interested in *aggregated* information related to the optimization *performance* of the ES, in detail:

- the dynamics of the average, best, and worst fitness,
- dynamics of the expected distance R to the optimum,
- local performance properties, such as:
  - progress rates,
  - quality gains,
- global performance properties, such as:
  - convergence proofs and orders of convergence,
  - expected running times and time complexities.

In some cases it is possible to obtain such results bypassing the determination of the full dynamics.

As an example, global convergence proofs for elitist ES versions (i.e., those using plus selection) are to be mentioned here (Rechenberg, 1973; Born, 1978). Considering continuous search spaces, the idea is to show that the probability of reaching a certain ϵ-vicinity (level set) of the global optimum $\hat{F} = F(\hat{\mathbf{y}})$ by a sequence of search points $\mathbf{y}^{(g)}$ is one, if the number of generations goes to infinity[8]

$$\forall \epsilon > 0: \quad \lim_{g\to\infty} \Pr\left(|F(\mathbf{y}^{(g)}) - \hat{F}| \le \epsilon\right) = 1. \tag{34}$$

While proofs of equation (34) are of certain mathematical interest and are relatively easy to be accomplished (for a comprehensive collection of such proofs, see Rudolph, 1997a), their relevance with respect to local and global performance properties of the ES is rather limited. This is so because ES users have only a limited amount of time, whereas equation (34) concerns a statement on the *infinite* time behavior.

Local and global performance properties are more important. However, performance measures and the dynamics of aggregated states are difficult to

obtain. In order to keep the mathematics tractable, simple objective functions or simplified algorithms must be considered with the hope that these simplifications still carry the characteristics of EA applied to real-world applications. We shall report some of such results in sections 5.3 and 5.4. The next section, however, will provide some example dynamics.

5.2 *The goal: Predicting the evolutionary dynamics*

5.2.1 *Motivation*

Looking at practical applications of evolutionary algorithms, in most cases the user does not know the structure of the fitness landscape. He typically has to face black-box scenarios and does not know the "hardness" of approximating or finding the optimum. In this typical situation, optimization of real-world problems is rather an *online procedure* than an *offline algorithm* usually considered and investigated in the framework of running time and complexity theory. This is a peculiarity of evolutionary search – optimization becomes *amelioration* and the emphasis is necessarily on the *evolutionary process*, i.e., on the dynamics of the optimum search. By looking at the dynamics of the fitness and other aggregated quantities of the population, the user gets valuable information in order to decide whether to stop or continue the search process. That is why we put such an emphasis on the dynamics of the process.

This does not imply that investigations concerning the time complexity of evolutionary algorithms are of no use, however, such investigations present a rather static view of the performance problem. Wegener (2000) argues that evolutionary algorithms are randomized algorithms (Motwani and Raghavan, 1995) and that they should be analyzed using the paradigms and tools from classical research on efficient algorithms. This reflects the computer scientist's interest in the resources needed by a program in order to solve a *specific* problem of given size N. The most crucial resource is usually the time needed to solve a given problem. Since ES are probabilistic optimization techniques, there is always a positive probability that the optimum will *not* be visited in a given time. Therefore the concept of *expected* (first) hitting time must be considered. This concept has proven especially successful when considering the performance of the ES on pseudo-Boolean functions (see section 5.4; for an overview, e.g., Wegener, 2001).

5.2.2 *Examples*

Let us consider two examples of amelioration processes in a real-valued and in a combinatorial search space. Figure 12 presents typical simple instances of best-so-far fitness dynamics (i.e., fitness function values F over number g of generations).

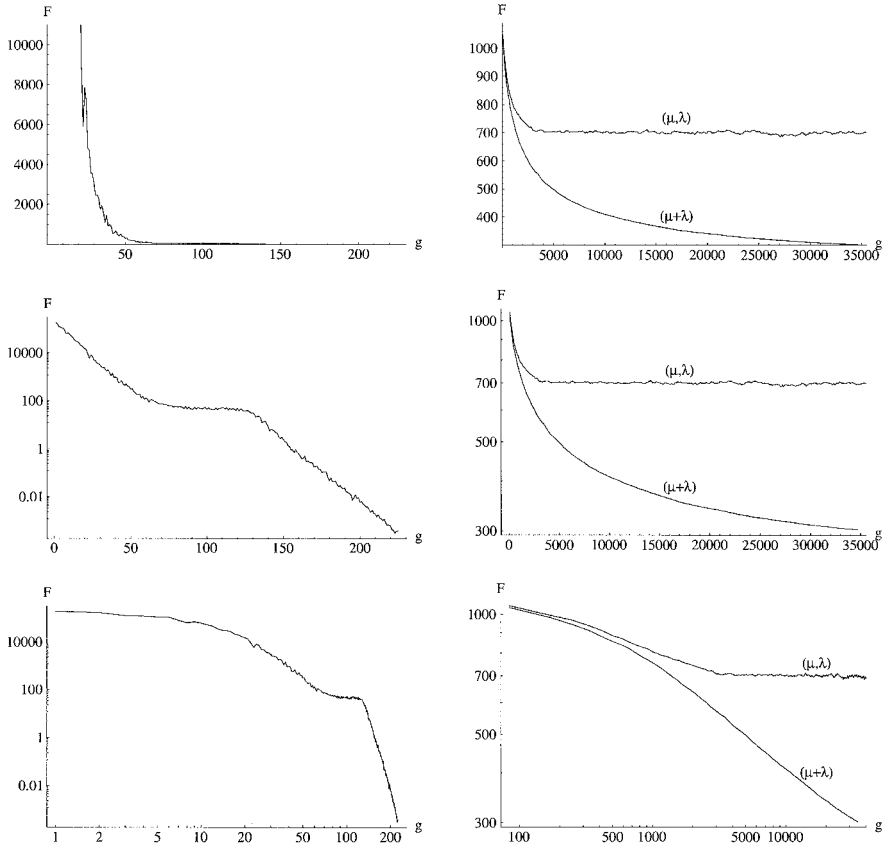


Figure 12. Typical dynamics of best fitness values in the case of an extremely multimodal R^N optimization problem (left-hand side) and in a combinatorial problem (right-hand side).

The graphs on the left-hand side of Figure 12 are from a *single* $(50/50, 100)$-ES run with isotropic Gaussian mutations, standard self-adaptation (log-normal rule, $\tau = 1/\sqrt{2N}$), and intermediate recombination (for both object and strategy parameters) applied to the so-called Rastrigin test function $F_{\text{Ras}}(\mathbf{y}) = \sum_{k=1}^{N}[y_k^2 + B(1 - \cos 2\pi y_k)]$ with $B = 2$ and $N = 30$. The number of local minima is known to be of the order $(2A + 1)^N$ (for $B = 2$ one can estimate $A = 6$). While finding the global optimum among such many local optima is usually a hopeless endeavor, the structure of F_{Ras} is of such a kind that there is a "certain tendency" toward the global minimum: Far away from the global minimum, the fitness landscape is quadratic. Approaching the global optimizer the landscape becomes increasingly rippled, however, all the local optima are concentrated around the global

attractor. Therefore, there is a certain chance for the ES to pass through the region of deception and locate the global attractor.

The approach to the optimum can be observed by looking at the fitness dynamics of the ES. The graphs on the left-hand side of Figure 12 display all the same (single) ES run, however, using different scales. As one can see, the linear-log-plot is the most informative one, whereas the linear-linear-plot, usually used for displaying the optimization dynamics, as well as the log-log-plot are not suited in this special situation. The linear-log-plot reveals a characteristical dynamics often observed when using self-adaptive ES in real-valued search spaces: *linear convergence order*.[9] The $\log F(g)$ curves are almost piecewise linear according $\log F(g) \approx -ag + b$.[10] That is, the F-dynamics is an exponentially falling function of time $F \propto \mathrm{e}^{-\alpha g}$ ($\alpha > 0$). This holds for the dynamics of the residual distance R to the optimum as well (for an explanation, see section 5.3). In our special example, one can see that the ES provides linear convergence when being far off the optimum and also in the attractor region of the global minimum. Approaching and leaving the region of local optima changes the convergence velocity (i.e., the slope of the $\log F(g)$ curve). Pass through the "sea of ripples" the velocity is considerably reduced. And it can also happen that the ES will be trapped by one of the local attractors (in order to reduce the probability of being trapped, the learning parameter τ should be chosen rather small).

The graphs on the right-hand side of Figure 12 show typical runs of a (μ, λ)-ES and a $(\mu + \lambda)$-ES, $\mu = 5$, $\lambda = 12$, on an $N = 2228$ cities TSP with random uniform distance matrix using the 2-exchange mutation operator from Figure 5. Again, the $F(g)$ plot is not very informative, but in this case, the log-log-plot is the most appropriate one. There are two interesting observations. First, (μ, λ)-ES are not well suited for combinatorial optimization, after a short phase of improvements they approach a steady-state behavior far off the optimum. Second, the fitness dynamics of the $(\mu + \lambda)$-ES appears almost linear as long as the ES has not reached a local attractor (the final convergence phase has not been depicted in Figure 12). That is, in combinatorial optimization problems we typically observe some kind of *sublinear convergence*. In such situations, the F-dynamics can be described by an $F \propto g^{-\alpha}$ law, $\alpha > 0$ (for an explanation, see Nürnberg and Beyer, 1997).

5.3 *Real-valued search spaces*

There is a long tradition in theoretical ES research dating back to the very beginning of ES. Schwefel (1965) presented analyses of (1 + 1)-ES on a discretized parabolic ridge as well as on the hyperplane and sphere model using isotropic Gaussian mutations. This work has been continued

by Rechenberg (1973) and Schwefel (1975) resulting in such concepts like progress rates and success probabilities. In this section we will review the basic concept of progress rate, present some examples and show the connections to the evolutionary dynamics.

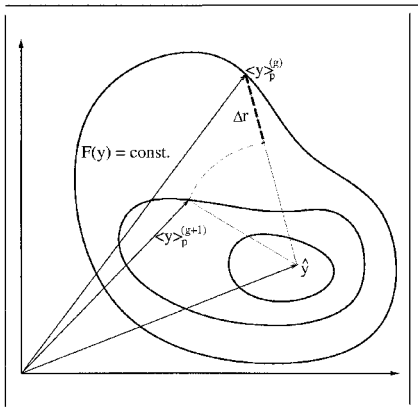
5.3.1 *Local performance measures*
Local performance measures are expected values of (aggregated) population states. They are usually defined in such a manner that they can be used to evaluate the amelioration power of the ES locally. When saying "locally" we mean local in time, i.e., the measures describe the amelioration success between two consecutive generations (g) and $(g+1)$.

5.3.1.1 *The progress rate definition.* The amelioration progress can be measured in the fitness space as well as in the object parameter space. The latter measure is called "progress rate," whereas the former is usually called "quality gain" (for a comprehensive overview, see Beyer, 2001b). Here we will only consider the progress rate φ. It is defined in Figure 13 as the expected distance change Δr of the parental population centroid in the object parameter space (search space) toward the optimizer $\hat{\mathbf{y}}$. Note, we used the lower case letter r in order to express that this (aggregated) quantity is a random variate, whereas the capital R is used to denote its expected value, i.e., $R = \mathrm{E}[r]$.

From the φ definition (35) it becomes clear that the progress rate depends on a number of factors: the ES and fitness function used and also on the parental population state $\mathfrak{P}_{\mathrm{p}}^{(g)}$. Calculating φ is difficult and only tractable for simple objective function models $F(\mathbf{y})$. This is one of the reasons why the investigations were mainly concentrated on the sphere model. Only recently, the so-called ridge function class, defined by $F_{\mathrm{rid}}(\mathbf{y}) = \mathbf{v}'\mathbf{y} - d\|(\mathbf{v}'\mathbf{y})\mathbf{v} - \mathbf{y}\|^{\alpha}$, with $\mathbf{v}'\mathbf{v} = 1$, has attracted intensive research (Oyman, 1999; Oyman et al., 2000; Oyman and Beyer, 2000; Beyer, 2001a).

5.3.1.2 *Example: Sphere model* The most prominent test function is the sphere model, already introduced in equation (16). It represents a highly symmetric function class the function values of which depend only on the distance R of $\mathbf{y}$ to the optimum. That is, R can be used as an aggregated state variable allowing for a reduction of the N-dimensional dynamics to the one-dimensional R-dynamics. Considering ES with isotropic Gaussian mutations of strength σ and given a parental state R, the progress rate φ depends only on the three parameters R, N, and σ. Using the normalizations

$$\varphi^* := \varphi\frac{N}{R} \quad \text{and} \quad \sigma^* := \sigma\frac{N}{R}, \tag{38}$$



Progress Rate:

$$\varphi := \mathrm{E}[\Delta r] = R^{(g)} - R^{(g+1)}, \quad (35)$$

with

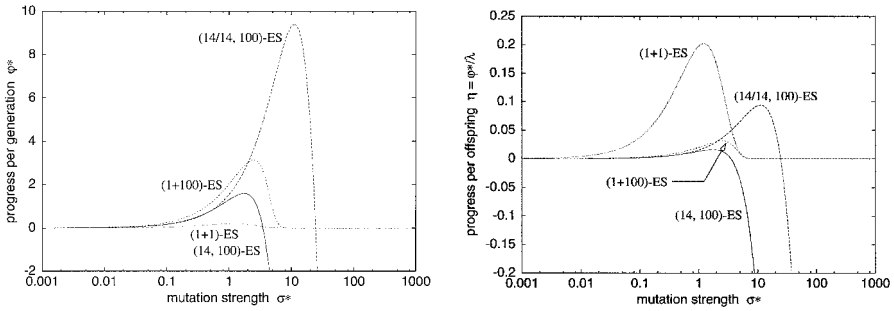
$$\Delta r := r^{(g)} - r^{(g+1)} \quad (36)$$

and

$$r^{(\cdot)} := \left\| \langle \mathbf{y} \rangle_{\mathrm{p}}^{(\cdot)} - \hat{\mathbf{y}} \right\| . \quad (37)$$

Figure 13. On the definition of the progress rate φ.

it can be shown that the normalized progress rate φ^* depends only on the normalized mutation strength σ^* and the search space dimension N. The dependency on N is rather weak such that considering the asymptotic case $N \to \infty$, $\varphi^* = f(\sigma^*)$ often suffices (Beyer, 2001b). Figure 14a shows such progress rate curves.



a) progress rate φ^* b) efficiency

Figure 14. On the local performance of different ES on the sphere model. The performance depends on the mutation strength (note, a logarithmic scale has been used).

Various conclusions concerning the performance of the ES on the sphere model can be drawn from that picture:

1. Plus strategies exhibit always positive progress independent of the mutation strength chosen (for the $(1+1)$-ES, see also Figure 10). That is, they continuously converge to the optimum.

2. Comma strategies can converge provided that the mutation strength is chosen appropriately. If the mutation strength is too large the strategy departs from the optimum, it exhibits divergent behavior.
3. Using a population $\lambda > 1$ increases the performance.
4. A $(\mu + \lambda)$-ES performs always better than a (μ, λ)-ES.
5. Using μ/μ intermediate recombination yields the best performance.
6. Recombinative strategies attain their performance maximum at higher mutation strengths.

In summary one may conclude that given the performance measure at hand, recombinative strategies are the ones that should be chosen.

As can be shown, the $(\mu/\mu, \lambda)$-ES can exhibit a performance speedup of up to λ compared to the $(1 + 1)$-ES. However, this speedup is a *parallel* one. It concerns a full generation step, but not the cost for generating the offspring and the fitness evaluations. Performance on a *serial* computer should be better evaluated by considering the number of fitness evaluations needed to obtain a certain improvement. That is, we have to investigate the progress per offspring. The related progress measure is called (fitness) efficiency and defined by $\eta := \varphi^*/\lambda$. It is depicted in Figure 14b. Under this condition the $(1 + 1)$-ES is the most efficient strategy. As has been shown theoretically, in the limit $N \to \infty$, $\lambda \to \infty$, the $(\mu/\mu, \lambda)$-ES approaches asymptotically the performance maximum of the $(1 + 1)$-ES. The optimum truncation ratio becomes $\mu/\lambda \simeq 0.270\ldots$.

5.3.1.3 *Example: Noisy sphere.* For a long time period it was an open problem to find a simple test function where it can be shown by theory that the $(\mu/\mu, \lambda)$ efficiency can exceed that of the $(1 + 1)$-ES. Recently, two breakthroughs have been made without leaving the realm of spherical models: first, a bimodal sphere model in $\mathbb{B}^N$, to be reported in section 5.4; secondly, the *noisy sphere* model in $\mathbb{R}^N$

$$F_{\text{nsp}}(\mathbf{y}) := c\|\mathbf{y}\|^\alpha + \varepsilon, \quad \text{with} \quad \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2). \tag{39}$$

In equation (39) the fitness $c\|\mathbf{y}\|^\alpha$ is disturbed by Gaussian noise with variance σ_ε^2. The asymptotical progress rate formula ($N \to \infty$) reads (Arnold and Beyer, 2001)

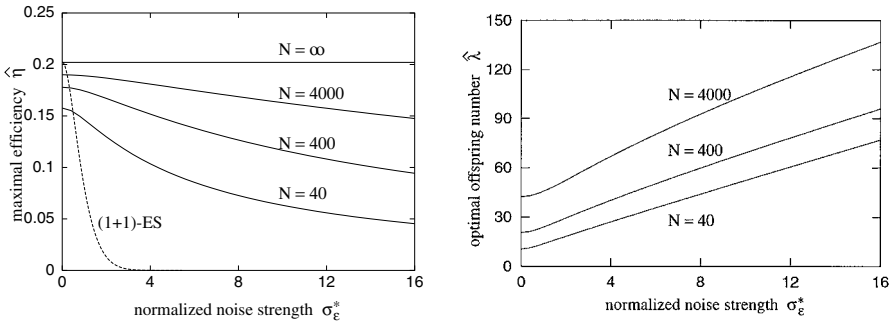
$$\varphi^* = \sigma^{*2}\left[\frac{c_{\mu/\mu,\lambda}}{\sqrt{\sigma^{*2} + \sigma_\varepsilon^{*2}}} - \frac{1}{2\mu}\right], \quad \text{with} \quad \sigma_\varepsilon^* := \sigma_\varepsilon \frac{N}{c\alpha R^\alpha}, \tag{40}$$

where $c_{\mu/\mu,\lambda}$ is the so-called progress coefficient. It can be approximated by the inverse error function erf^{-1}

$$c_{\mu/\mu,\lambda} \simeq \frac{\lambda}{\mu}\frac{1}{\sqrt{2\pi}} \exp\left[-\left(\text{erf}^{-1}\left(1 - 2\frac{\mu}{\lambda}\right)\right)^2\right] \tag{41}$$

and its values are of order $\mathcal{O}(\sqrt{\ln(\lambda/\mu)})$. (Note, formula (40) contains also the noise-free sphere and the $(1, \lambda)$-ES as special cases.)

As can be shown for *any* level $\sigma_\varepsilon^* < \infty$ of noise, the $(\mu/\mu, \lambda)$-ES reaches asymptotically the (maximal) efficiency of the $(1 + 1)$-ES *without* noise, whereas the efficiency of the $(1+1)$-ES with noise degrades increasingly with σ_ε^*. That is, the $(\mu/\mu, \lambda)$-ES outperforms the $(1 + 1)$-ES when the objective function is disturbed by noise. While this is correct for $N \to \infty$, Figure 15a shows that this also holds for $N < \infty$ and sufficiently high noise levels.



a) maximal efficiencies b) optimal number of offspring

Figure 15. Optimal performance of maximally efficient strategies on the noisy sphere.

Figure 15a compares the result from the $(1 + 1)$-ES performance analysis (Arnold and Beyer, 2000a) with the optimal efficiency of $(\mu/\mu, \lambda)$-ES using the N-dependent progress rate theory (Arnold and Beyer, 2000b). Note, these graphs do *not* correspond to a fixed μ and λ value. As shown in Figure 15b, the offspring size λ depends on N and the normalized noise level σ_ε^*. Interestingly, the optimal μ/λ ratio is again $\mu/\lambda \approx 0.27$.

5.3.2 *Dynamics*

The mean value dynamics of aggregated population states is usually governed by a system of $N + N_s$ difference equations, where N_s is the dimensionality of the strategy parameter set. Under certain conditions and assumptions this system reduces to a single difference equation.

For example, when considering the sphere model, the residual distance R to the optimum and the mutation strength σ^* of the parental centroid can be used as aggregated state variables. Using the φ definition (35) and the normalization (38), one gets

$$R^{(g+1)} = R^{(g)} \left(1 - \frac{\varphi^*(\sigma^{*(g)})}{N}\right). \tag{42}$$

Provided that $|\varphi^*/N| \ll 1$, this difference equation can be approximated by the differential equation

$$\frac{\mathrm{d}R(g)}{\mathrm{d}g} = -R(g)\frac{\varphi^*(\sigma^*(g))}{N}. \tag{43}$$

Its formal solution reads

$$R(g) = R(g_0)\exp\left(-\frac{1}{N}\int_{t=g_0}^{t=g}\varphi^*(\sigma^*(t))\,\mathrm{d}t\right). \tag{44}$$

This solution still depends on the time behavior of the mutation strength σ^*, i.e., the R-dynamics is governed by the dynamics of the σ control algorithm.

Special case: $\sigma^*(g) = \text{const}$. That is, it is assumed that the σ control, e.g., σ self-adaptation, has reached nearly steady-state behavior at time g_0. Under this condition, equation (44) has a simple solution

$$R(g) = R(g_0)\exp\left(-\frac{\varphi^*(\sigma^*)}{N}(g - g_0)\right). \tag{45}$$

Provided that $\varphi^* > 0$, the residual distance decreases exponentially fast. Taking the logarithm in (45) one immediately sees that we have *linear convergence order* (cf. section 5.2.2) that also transfers to the fitness dynamics.

Equation (45) can be used to estimate the time complexity and the number of fitness evaluations needed to reach a certain vicinity of the optimum. To this end, we consider the expected relative amelioration progress over a number G of generations. Using (45), one finds $R^{(g+G)}/R^{(g)} = \exp(-\varphi^* G/N)$ and solving for G yields the expected run time

$$G = N\,\frac{\ln(R^{(g)}/R^{(g+G)})}{\varphi^*}. \tag{46}$$

Given fixed exogenous strategy parameters μ and λ (i.e., $\mu, \lambda \neq f(N)$), the expected running time scales linearly in N. This holds also for the number of functions evaluations ν because $\nu = \lambda G + \mu$.

5.3.2.2 *Evolution criteria and non-convergence.* Convergence toward the optimum is ensured as long as the residual R distance decreases with time. Using equation (43) one sees that this is equivalent to $\varphi^* > 0$. As an example, let us again consider the $(\mu/\mu, \lambda)$-ES on the sphere model. Taking equation (40) into account one immediately finds the *evolution criterion* by which convergence is ensured

$$\sigma^{*2} + \sigma_\varepsilon^{*2} < 4\mu^2 c^2_{\mu/\mu,\lambda}. \tag{47}$$

Considering the noise free case $\sigma_\varepsilon^* = 0$, one finds $\sigma^* < 2\mu c_{\mu/\mu,\lambda}$. What happens when the *non*normalized mutation strength is kept constant? In that situation σ^* increases with decreasing $R^{(g)}$ because of (38). This increase proceeds until the evolution criterion (47) is violated, i.e., in the noise free case $\sigma N/R^{(g)} = \sigma^{*(g)} = 2\mu c_{\mu/\mu,\lambda}$. That is, the ES does not converge to the optimum. The evolution stagnates and the population fluctuates around a specific R, the final localization error

$$R_\infty = \frac{\sigma N}{2\mu c_{\mu/\mu,\lambda}}. \tag{48}$$

This is a typical situation for comma strategies when the mutation strength cannot be down-scaled arbitrarily (cf. Figure 12d, e, f).

A similar behavior can be observed when considering a constant noise strength σ_ε in (39). Criterion (47) can be used to derive a lower bound for the final localization error (for details, see Beyer, 2000).

5.4 *Binary search spaces*

Theoretical investigations concerning the dynamical behavior of ES in $\mathbb{B}^N$ search spaces on pseudo-Boolean functions are still to be performed. However, considerable progress has been made concerning the expected run time and global success probabilities of ES with plus selection. We will review only some of these results referring the reader to the original work by Wegener et al. (see below).

Unlike continuous search spaces, where the number of states is uncountably infinite, the $\mathbb{B}^N$ search space is of finite size 2^N. Therefore, one can consider the event of first visiting the optimum state $\hat{\mathbf{y}}$, i.e., the number t of function evaluations or test states generated until $\mathbf{y}^{(t)} = \hat{\mathbf{y}}$ or $F(\mathbf{y}^{(t)}) = \hat{F}$ is fulfilled. Taking the expectation of t, one gets the expected run time $T := \mathrm{E}[t]$.

5.4.1 *Global performance of the* $(1 + 1)$*-ES*

Most results obtained so far concern the performance of the $(1 + 1)$-ES. For linear functions

$$F_{\mathrm{lin}}(\mathbf{y}) := w_0 + \sum_{k=1}^{N} w_k y_k, \quad \text{with} \quad \forall k: \; w_k \neq 0, \quad w_k \in \mathbb{R}, \tag{49}$$

Droste et al. (1998b) were able to bracket the expected run time T of a $(1 + 1)$-ES with bit mutation rate $1/N$ to

$$T(N) = \Theta(N \log N). \tag{50}$$

The Θ notation indicates that $N \log N$ is both the upper and the lower bound, i.e., $\kappa_1 N \log N \leq T(N) \leq \kappa_2 N \log N$ with $0 < \kappa_1 < \kappa_2 < \infty$.

Rudolph (1997a) proposed the "LeadingOnes" function

$$F_{\mathrm{LO}}(\mathbf{y}) := \sum_{j=1}^{N} \prod_{k=1}^{j} y_k \tag{51}$$

which counts the number of leading ones in the binary string. He proved the upper bound $\mathcal{O}(N^2)$ for maximizing this function, while Jansen (2000) was able to show

$$T(N) = \Theta(N^2), \tag{52}$$

thus closing the lower bound.

In an attempt to show the superiority of crossover in genetic algorithms against simple mutative EA, Horn et al. (1994) proposed so-called long-path problems where a hillclimber is intended to follow an exponentially long path while a recombinative strategy bypasses the long path. The designers of the original model function oversaw, however, that the standard mutation operator using $p_m = 1/N$ can perform jumps of Hamming distance greater than 1 with a certain probability. Therefore, Rudolph (1997b) was able to prove that $T(N) = \mathcal{O}(N^3)$ holds for the original problem. Furthermore, he presented other versions of long-path problems having a higher time complexity. Droste et al. (1998a) followed this road and presented unimodal model functions with exponential optimization duration.

For a subclass of quadratic functions

$$F_{\mathrm{qua}}(\mathbf{y}) := w_0 + \sum_{k=1}^{N} w_k y_k + \sum_{j=1}^{N} \sum_{k=j+1}^{N} w_{jk} y_j y_k, \tag{53}$$

with $w_k, w_{jk} > 0$ and n as the number of non-zero weights, T can be bounded from above

$$T(N) = \mathcal{O}(nN^2) = \mathcal{O}(N^3), \tag{54}$$

i.e., $T \leq \kappa n N^2$ (Wegener and Witt, 2001). Among others the authors also proved that the square of $F_{\mathrm{lin}}(\mathbf{y})$ is optimized in expected time $T(N) = \mathcal{O}(N \log N)$ with success probability $1/8 - \epsilon$ ($\epsilon > 0$ arbitrarily small). In other words $(F_{\mathrm{lin}}(\mathbf{y}))^2$ can be efficiently treated by a *multistart* version of the $(1 + 1)$-ES.

5.4.2 *Performance increase by recombination in* $(\mu/2 + 1)$*-ES*

After a long period of unsuccessful attempts (Mitchell et al., 1994, Horn et al., 1994) to show performance benefits of recombination in $\mathbb{B}^N$, Jansen and Wegener (1999) presented a test function where discrete ($\rho = 2$) recombination (11) provably decreases the expected run time compared to an ES using mutation with rate $p_m = 1/N$ only. The function they considered, $F_{\text{Jump}}(\mathbf{y})$, is a sphere model using the Hamming distance $\|\mathbf{y}\| = \sum_{k=1}^{N} y_k$ as radius

$$F_{\text{Jump}}(\mathbf{y}) = F(\|\mathbf{y}\|) = \begin{cases} \|\mathbf{y}\|, & \text{if } \|\mathbf{y}\| \leq N - M \text{ or } \|\mathbf{y}\| = N, \\ N - \|\mathbf{y}\|, & \text{otherwise.} \end{cases} \tag{55}$$

Unlike the unimodal sphere models considered in section 5.3, F_{Jump} is a bimodal one having a local (degenerated) maximum at Hamming distance $\|\mathbf{y}\| = N - M$ and the global maximum at $\|\mathbf{y}\| = N$. The function value first increase with $\|\mathbf{y}\|$ up to $N - M$, then it drops down, and finally it jumps to N. That is, there is a gap of size M the ES must bridge.

There is a lower bound for optimizing F_{Jump} using the $(1 + 1)$-ES. It reads

$$(1 + 1)\text{-ES:} \quad T(N) = \Omega(N^M), \tag{56}$$

here the Ω symbol expresses that $T(N) \geq \kappa N^M$. When using $p_m = 1/N$ it can be shown that $T(N) = \Theta(N^M)$.

In order to prove the performance benefits of recombination the authors considered a steady-state ES with discrete two-parent recombination, i.e., $(\mu/2 + 1)$. Using a somehow unusual and very small recombination probability of $p_c = 1/(N \log N)$ (usually $p_c = 1$), they obtained

$$(\mu/2 + 1)\text{-ES:} \quad \begin{cases} T(N) = \mathcal{O}(N^2 \log N), & \text{if } M \neq f(N), \\ T(N) = \mathcal{O}(N^3 \log N), & \text{if } M = \lfloor \log N \rfloor. \end{cases} \tag{57}$$

Comparing the second line in (57) with (56) it becomes clear that the $(1 + 1)$-ES needs superpolynomial time while the recombinative ES has polynomial expected run time. Thus, it has been shown that recombination can help.

While this was a first step in showing a (serial) performance advantage of recombinative strategies, the question remained whether the superpolynomial to polynomial time gap could be extended to an exponential to polynomial time gap. This can be shown indeed, however, up to now only by using a very artificial fitness function (see Jansen and Wegener, 2001).

6. Outlook

Above, we have considered only mainstream ES, i.e., established older instantiations of evolution strategies that mimic just a few basic principles gleaned from nature. Organic evolution has got many more features, some more of which might be worth to be considered as additional sources of inspiration for creating artificial optimum seeking and adaptation procedures. Some proposals of that kind, though neither having gone yet through rigorous analysis nor exhaustive empirical testing, will be sketched below.

The maximal lifespan of an individual is not required to be either unlimited as in the plus version of the standard algorithm or limited to just one iteration (generation) as in the comma version. Any positive integer κ can be used to control the maximal number of iterations (now better reproduction cycles) an individual is allowed to stay within the population. Only after κ cycles it will be deleted from the gene pool even if it cannot be replaced by a better or at least equally fit descendent. This version has been called $(\mu, \kappa, \lambda, \rho)$-ES (Schwefel and Rudolph, 1995), in which ρ, as usual, stands for the number of predecessors an offspring is composed of by recombination. One might expect that certain κ values are better than others to support the self-adaptation of strategy parameters.

Two good reasons led to a rather different type of an evolutionary algorithm without explicit generation transition. One reason was the unavoidable idling of processors of a parallel computer due to the fact that the simulations delivering the fitness values of a generation's offspring (perhaps also testing their feasibilities) often are of very different duration. Lesser synchronization of the selection and the reproduction phases should enhance the efficiency of the evolutionary search substantially in some cases. The second reason for a predator-prey approach (Laumanns et al., 1998; Laumanns et al., 2001) in modeling interwoven exploration and exploitation processes was the easiness with which multiple objectives can be handled just by letting different predators select according to different criteria at the same time. In order to implement such strategy it seems quite natural to let the prey, who bear the object variables as well as the strategy parameters, inhabit distinct places on a (toroidal) grid. The predators perform random hunting walks on the same grid eating the weakest prey in a certain neighborhood and thus opening positions for newborn offspring of the survivors among the prey. Hopefully, the prey finally roam in the vicinity of the Pareto set of nondominated solutions. As always, the variation strength control is crucial for the functioning of this approach (Rudolph and Agapie, 2000).

Further still mostly unexplored options of enhanced ES/EA are the introduction of diploidy or even polyploidy (Kursawe, 1992) and of multicellularity together with somatic mutations (Schwefel and Kursawe, 1998). More

advances in these and other directions can be expected in future issues of the handful journals in the field of evolutionary computation as well as in the proceedings of the nearly two dozens of relevant conferences being held annually all around the world.

Acknowledgements

The author acknowledges support as Heisenberg Fellow of the Deutsche Forschungsgemeinschaft under grant Be 1578/4-2.

Notes

[1] In order to achieve a mutation with zero mean change from the parent position, the difference of two binomially distributed random samplings was used, actually – with probability parameters $1/2$ and repetition numbers according to the wanted variance.
[2] This kind of performance measure does not take into account the cost of *parallel* computation. If cost considerations become important, the benefit of an earlier result must, however, also be taken into account.
[3] While optimization is the main application domain of ES, it has been demonstrated that ES can also be applied to the simulation of dissipative systems (Beyer, 1990).
[4] Note, in many papers on ES the notations (μ, λ) and $(\mu + \lambda)$ always included some kind of recombination implicitly. The $(\mu/\rho \stackrel{+}{,} \lambda)$ notation has been introduced in (Rechenberg, 1978) first.
[5] Note, this does not mean that recombination is of secondary importance. It only means that recombination usually serves a different purpose. If applicable, the use of recombination is strongly recommended (see sections 3.4 and 5.3).
[6] The term "mutation strength" is used here intuitively, in real-valued search spaces it refers to the standard deviation of a single component of the mutation vector.
[7] Note, without selection the components would be isotropically distributed (i.e., symmetrically with respect to the improvement direction) because of the isotropy of the $\mathbf{z}$ mutations used.
[8] For discrete search spaces (34) simplifies to $\lim_{g\to\infty} \Pr\left(F(\mathbf{y}^{(g)}) = \hat{F}\right) = 1$.
[9] Analyzing $(1 + 1)$-ES like random search strategies on convex optimization problems, Rappl (1989) was the first to introduce the concept of convergence order in this field.
[10] This assumes the global minimum value $\check{F} = 0$. If $\check{F} \neq 0$, $\log(F(g) - \check{F})$ must be considered instead. Alternatively, one can consider the residual distance R to the optimum (measured in the object parameter search space).

References

Altenberg L (1994) The evolution of evolvability in genetic programming. In: Kinnear K (ed) Advances in Genetic Programming, pp. 47–74. MIT Press, Cambridge, MA
Arnold BC, Balakrishnan N and Nagaraja HN (1992) A First Course in Order Statistics. Wiley, New York

Arnold DV and Beyer H-G (2000a) Local performance of the (1 + 1)-ES in a noisy environment. IEEE Transactions on Evolutionary Computation. accepted for publication

Arnold DV and Beyer H-G (2000b) Performance analysis of evolution strategies with multi-recombination in high-dimensional $\mathbb{R}^N$-search spaces disturbed by noise. *Theoretical Computer Science*. In print

Arnold DV and Beyer H-G (2001) Local performance of the $(\mu/\mu_I, \lambda)$-ES in a noisy environment. In: Martin W and Spears W (eds) Foundations of Genetic Algorithms, 6, pp. 127–141. Morgan Kaufmann, San Francisco, CA

Bäck T (1996) Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York, NY

Bäck T, Fogel D and Michalewicz Z (eds) (1997) Handbook of evolutionary computation. IOP Publishing and Oxford University Press, New York

Beyer H-G (1990) Simulation of steady states in dissipative systems by darwin's paradigm of evolution. J. Non-Equilib. Thermodyn. 15: 45–58

Beyer H-G (1992) Some aspects of the 'evolution strategy' for solving tsp-like optimization problems. In: Männer R and Manderick B (eds) Parallel Problem Solving from Nature, 2, pp. 361–370. Elsevier, Amsterdam

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

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

Beyer H-G (1997) An alternative explanation for the manner in which genetic algorithms operate. BioSystems 41: 1–15

Beyer H-G (2000) Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice. Computer Methods in Applied Mechanics and Engineering 186(2–4): 239–267

Beyer H-G (2001a) On the performance of $(1, \lambda)$-evolution strategies for the ridge function class. IEEE Transactions on Evolutionary Computation 5(3): 218–235

Beyer H-G (2001b) The Theory of Evolution Strategies. Natural Computing Series. Springer, Heidelberg

Beyer H-G and Deb K (2001) On self-adaptive features in real-parameter evolutionary algorithms. IEEE Transactions on Evolutionary Computation 5(3): 250–270

Born J (1978) Evolutionsstrategien zur numerischen Lösung von Adaptationsaufgaben. Dissertation A. Humboldt-Universität, Berlin

De Jong K, David D, Fogel B and Schwefel H-P (1997) A history of evolutionary computation. In: Bäck T, Fogel DB and Michalewicz Z (eds) Handbook of Evolutionary Computation, pp. A2.3:1–12. Oxford University Press, New York, and Institute of Physics Publishing, Bristol

Droste S, Jansen T and Wegener I (1998a) On the optimization of unimodal functions with the (1 + 1) evolutionary algorithm. In: Eiben A, Bäck T, Schoenauer M and Schwefel H-P (eds) Parallel Problem Solving from Nature, 5, pp. 13–22. Springer-Verlag, Heidelberg

Droste S, Jansen T and Wegener I (1998b) A rigorous complexity analysis of the $(1+1)$ evolutionary algorithm for separable functions with Boolean inputs. Evolutionary Computation 6(2): 185–196

Droste S and Wiesmann D (2000) Metric based evolutionary algorithms. In: Poli R, Banzhaf W, Langdon W, Miller J, Nordin P and Fogarty T (eds) Proc. of the Third European Conference on Genetic Programming, EuroGP 2000, pp. 29–43. Springer, Berlin

Eiben AE, Hinterding R and Michalewicz Z (1999) Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation 3(2): 124–141

Fisz M (1971) Wahrscheinlichkeitsrechnung und mathematische Statistik. VEB Deutscher Verlag der Wissenschaften, Berlin

Fogel D (ed) (1998) Evolutionary Computation: The Fossil Record. IEEE Press, Piscataway, NJ

Fogel DB, Schwefel H-P, Bäck T and Yao X (eds) (1998) Proc. Second IEEE World Congress on Computational Intelligence (WCCI'98) with Fifth IEEE Conf. Evolutionary Computation (IEEE/ICEC'98) Anchorage AK, May 4–9, 1998 IEEE Press, Piscataway, NJ

Fogel LJ (1962) Autonomous automata. Industrial Research 4: 14–19

Fogel LJ, Owens AJ and Walsh MJ (1966) Artificial Intelligence through Simulated Evolution. Wiley, New York

Goldberg D (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, Reading, MA

Grünz L and Beyer H-G (1999) Some observations on the interaction of recombination and self-adaptation in evolution strategies. In: Angeline P (ed) Proceedings of the CEC'99 Conference, pp. 639–645. IEEE, Piscataway, NJ

Hansen N and Ostermeier A (1996) Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: Proceedings of 1996 IEEE Int'l Conf. on Evolutionary Computation (ICEC '96), pp. 312–317. NY, IEEE Press

Hansen N and Ostermeier A (1997) Convergence properties of evolution strategies with the derandomized covariance matrix adaptation: The $(\mu/\mu_I, \lambda)$-CMA-ES. In: Zimmermann HJ (ed) 5th European Congress on Intelligent Techniques and Soft Computing (EUFIT'97), pp. 650–654. Aachen, Germany, Verlag Mainz

Hansen N and A Ostermeier (2001) Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation 9(2): 159–195

Hartmann D (1974) Optimierung balkenartiger Zylinderschalen aus Stahlbeton mit elastischem und plastischem Werkstoffverhalten. Dr.-Ing. Dissertation, University of Dortmund, Dortmund

Herdy M (1990) Application of the 'evolutionsstrategie' to discrete optimization problems. In: Schwefel H-P and Männer R (eds) Parallel Problem Solving from Nature, 1, pp. 188–192. Springer-Verlag, Berlin.

Herdy M (1992) Reproductive isolation as strategy parameter in hierarchically organized evolution strategies. In: Männer R and Manderick B (eds) Parallel Problem Solving from Nature, 2, pp. 207–217. Elsevier, Amsterdam.

Holland JH (1962) Outline for a logical theory of adaptive systems. JACM 9: 297–314

Holland JH (1975) Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor

Holland JH (1995) Hidden Order: How Adaptation Builds Complexity. Addison-Wesley, Reading, MA

Horn J, Goldberg D and Deb K (1994) Long path problems. In: Davidor Y, Männer R and Schwefel H-P (eds) Parallel Problem Solving from Nature, 3, pp. 149–158. Springer-Verlag, Heidelberg

Jansen T (2000) Theoretische analyse evolutionärer Algorithmen unter dem Aspekt der Optimierung in diskreten Suchräumen. Phd thesis, Univ. of Dortmund, Dortmund, Germany (in German).

Jansen T and Wegener I (1999) On the analysis of evolutionary algorithms – a proof that crossover really can help. In: Nesetril J (ed) Proceedings of the 7th Annual European Symposium on Algorithms (ESA '99), pp. 184–193. Berlin, Germany, LNCS 1643, Springer

Jansen T and Wegener I (2000) On the choice of the mutation probability for the (1 + 1) EA. In: M Schoenauer (ed) Parallel Problem Solving from Nature, 6, pp. 89–98. Springer, Heidelberg

Jansen T and Wegener I (2001) Real royal road functions – where crossover provably is essential. In: Spector L (ed) GECCO'01: Proceedings of the Genetic and Evolutionary Computation Conference. Morgan Kaufmann, San Francisco, CA

Jaynes ET (1979) Where do we stand on maximum entropy? In: Levine R and Tribus M (eds) The Maximum Entropy Formalism, pp. 15–118

Kappler C, Bäck T, Heistermann J, Van de Velde A and Zamparelli M (1996) Refueling of a nuclear power plant: comparison of a naive and a specialized mutation operator. In: Voigt H-M, Ebeling W, Rechenberg I and Schwefel H-P (eds) Parallel Problem Solving from Nature – PPSN IV, Int'l Conf. Evolutionary Computation, pp. 829–838. Springer, Berlin

Klockgether J and Schwefel H-P (1970) Two-phase nozzle and hollow core jet experiments. In: Elliott DG (ed) Proc. 11th Symp. Engineering Aspects of Magnetohydrodynamics, pp. 141–148. California Institute of Technology, Pasadena CA

Kursawe F (1992) Evolution strategies for vector optimization. In: Tzeng G-H and Yu P-L (eds) Preliminary Proc. Tenth Int'l Conf. Multiple Criteria Decision Making, pp. 187–193. National Chiao Tung University, Taipei

Kursawe F (1999) Grundlegende empirische Untersuchungen der Parameter von Evolutionsstrategien – Metastrategien. Dr. rer. nat.–Dissertation, University of Dortmund, Department of Computer Science, Chair of Systems Analysis. Schwefel.

Laumanns M, Rudolph G and Schwefel H-P (1998) A spatial predator-prey approach to multi-objective optimization. In: Eiben AE, Bäck T, Schoenauer M and Schwefel H-P (eds) Parallel Problem Solving from Nature – PPSN V, Fifth Int'l Conf., Amsterdam, The Netherlands, September 27–30, 1998, Proc., pp. 241–249. Springer, Berlin

Laumanns M, Rudolph G and Schwefel H-P (2001) Mutation control and convergence in evolutionary multi-objective optimization. In: Matousek R and Osmera P (eds) Proc. Seventh Int'l Conf. Soft Computing (MENDEL'01), pp. 24–29. Brno University of Technology, Brno, Czech Republic

Lin S and Kernighan BW (1973) An effective heuristic algorithm for the traveling salesman problem. Oper. Res. 21: 498–516

Lohmann R (1992) Structure evolution and incomplete induction. In: Männer R and Manderick B (eds) Parallel Problem Solving from Nature, 2, pp. 175–185. Elsevier, Amsterdam

Michalewicz Z, Schaffer JD, Schwefel H-P, Fogel DB and Kitano H (eds) (1994) Proc. First IEEE Conf. Evolutionary Computation, Vol. I (oral presentations) and II (posters) of IEEE World Congress on Computational Intelligence. Orlando FL. The Institute of Electrical and Electronics Engineers, IEEE-Press, Piscataway NJ

Mitchell M, Holland J and Forrest S (1994) When will a genetic algorithm outperform hill climbing. In: Cowan J, Tesauro G and Alspector J (eds) Advances in Neural Information Processing Systems, pp. 51–58. Morgan Kaufmann, San Mateo, CA

Motwani R and Raghavan P (1995) Randomized Algorithms. Cambridge University Press, New York, NY

Mühlenbein H and Mahnig T (1999) FDA a scalable evolutionary algorithm for the optimization of additively decomposed functions. Evolutionary Computation 7(4): 353–376

Nürnberg H-T and Beyer H-G (1997) The dynamics of evolution strategies in the optimization of traveling salesman problems. In: Angeline P, Reynolds R, McDonnell J and Eberhart R (eds) Evolutionary Programming VI: Proceedings of the Sixth Annual Conference on Evolutionary Programming, pp. 349–359. Springer-Verlag, Heidelberg

Ostermeier A, Gawelczyk A and Hansen N (1994) Step-size adaptation based on non-local use of selection information. In: Davidor Y, Männer R and Schwefel H-P (eds) Parallel Problem Solving from Nature, 3, pp. 189–198. Springer-Verlag, Heidelberg

Oyman AI (1999) Convergence behavior of evolution strategies on ridge functions. Ph.D. Thesis, University of Dortmund, Department of Computer Science

Oyman AI and Beyer H-G (2000) Analysis of the $(\mu/\mu, \lambda)$-ES on the parabolic ridge. Evolutionary Computation 8(3): 267–289

Oyman AI, Beyer H-G and Schwefel H-P (2000) Analysis of a simple ES on the "parabolic ridge". Evolutionary Computation 8(3): 249–265

Pelikan M, Goldberg D and Cantu-Paz E (1999) BOA: the bayesian optimization algorithm. In: Banzhaf W, Daida J, Eiben A, Garzon M, Honavar V, Jakiela M and Smith R (eds) GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 525–532. Morgan Kaufmann, San Francisco, CA

Rappl G (1989), On linear convergence of a class of random search algorithms. Zeitschrift f. angew. Math. Mech. (ZAMM) 69(1): 37–45

Rechenberg I (1965) Cybernetic solution path of an experimental problem. Royal Aircraft Establishment, Farnborough p. Library Translation 1122

Rechenberg I (1971) Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Dr.-Ing. Thesis, Technical University of Berlin, Department of Process Engineering

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

Rechenberg I (1978) Evolutionsstrategien. In: Schneider B and Ranft U (eds) Simulationsmethoden in der Medizin und Biologie, pp. 83–114. Springer-Verlag, Berlin

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

Rudolph G (1992) On correlated mutations in evolution strategies. In: Männer R and Manderick B (eds) Parallel Problem Solving from Nature – Proc. Second Conf. PPSN, pp. 105–114. Free University of Brussels, Elsevier, Amsterdam

Rudolph G (1994) An evolutionary algorithm for integer programming. In: Davidor Y, Männer R and Schwefel H-P (eds) Parallel Problem Solving from Nature, 3, pp. 139–148. Springer-Verlag, Heidelberg

Rudolph G (1997a) Convergence Properties of Evolutionary Algorithms. Verlag Dr. Kovač, Hamburg. PhD-Thesis

Rudolph G (1997b) How mutation and selection solve long-path problems in polynomial expected time. Evolutionary Computation 4(2): 195–205

Rudolph G and Agapie A (2000) Convergence properties of some multi-objective evolutionary algorithms. In: Zalzala A and Eberhart R (eds) Proc. 2000 Congress on Evolutionary Computation (CEC'00), Vol. 2. La Jolla CA, pp. 1010–1016. IEEE Press, Piscataway NJ

Schwefel H-P (1965) Kybernetische Evolution als Strategie der exprimentellen Forschung in der Strömungstechnik. Master's thesis, Technical University of Berlin

Schwefel H-P (1968) Experimentelle Optimierung einer Zweiphasendüse Teil I. Technical Report No. 35 of the Project MHD–Staustrahlrohr 11.034/68, AEG Research Institute, Berlin

Schwefel H-P (1975) Evolutionsstrategie und numerische Optimierung. Dissertation, TU Berlin, Germany

Schwefel H-P (1977) Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie, Interdisciplinary systems research; 26. Birkhäuser, Basel

Schwefel H-P (1981) Numerical Optimization of Computer Models. Wiley, Chichester

Schwefel H-P (1987) Collective phenomena in evolutionary systems. In: Checkland P and Kiss I (eds) 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, Vol. 2. Budapest, pp. 1025–1033. Int'l Soc. for General System Research

Schwefel H-P (1995) Evolution and Optimum Seeking. Wiley, New York, NY

Schwefel H-P and Kursawe F (1998) On natural life's tricks to survive and evolve. In: Fogel DB, Schwefel H-P, Bäck T and Yao X (eds) Proc. Second IEEE World Congress on Computational Intelligence (WCCI'98) with Fifth IEEE Conf. Evolutionary Computation (IEEE/ICEC'98), pp. 1–8. Anchorage AK, IEEE Press, Piscataway NJ

Schwefel H-P and Rudolph G (1995) Contemporary evolution strategies. In: Morana F, Moreno A, Merelo JJ and Chacon P (eds) Advances in Artificial Life. Third ECAL Proceedings, pp. 893–907. Springer-Verlag, Berlin

Sendhoff B, Kreuz M and von Seelen W (1997) A condition for the genotype-phenotype mapping: Causality. In: Bäck T (ed) Proc. 7th Int'l Conf. on Genetic Algorithms, pp. 73–80. Morgan Kaufmann Publishers, Inc., Francisco, CA

Syswerda G (1989) Uniform crossover in genetic algorithms. In: Schaffer JD (ed) Proc. 3rd Int'l Conf. on Genetic Algorithms, pp. 2–9. Morgan Kaufmann, San Mateo, CA.

Wegener I (2000) On the design and analysis of evolutionary algorithms. In: Workshop on Algorithm Engineering as a New Paradigm. Kyoto, pp. 36–47. Research Institute for Mathematical Science, Kyoto Univ.

Wegener I (2001) Theoretical aspects of evolutionary algorithms. In: Spirakis P (ed), Proc. 28th International Colloquium on Automata, Languages and Programming. Springer-Verlag, Berlin

Wegener I and Witt C (2001) On the analysis of a simple evolutionary algorithm on quadratic pseudo-Boolean functions. Submitted to Journal of Discrete Algorithms

Yao X (1999) Evolving artificial neural networks. Proceedings of the IEEE 87(9): 1423–1447