Noisy Optimization with Evolution Strategies

Dirk V. Arnold and Hans-Georg Beyer
Department of Computer Science XI
University of Dortmund
44221 Dortmund, Germany
{arnold,beyer}@ls11.cs.uni-dortmund.de

Abstract

Evolution strategies are general, nature-inspired heuristics for search and optimization. Supported both by empirical evidence and by recent theoretical findings, there is a common belief that evolution strategies are robust and reliable, and frequently they are the method of choice if neither derivatives of the objective function are at hand nor differentiability and numerical accuracy can be assumed. However, despite their widespread use, there is little exchange between members of the "classical" optimization community and people working in the field of evolutionary computation. It is our belief that both sides would benefit from such an exchange.

In this paper, we present a brief outline of evolution strategies and discuss some of their properties in the presence of noise. We then empirically demonstrate that for a simple but nonetheless nontrivial noisy objective function, an evolution strategy outperforms other optimization algorithms designed to be able to cope with noise. The environment in which the algorithms are tested is deliberately chosen to afford a transparency of the results that reveals the strengths and shortcomings of the strategies, making it possible to draw conclusions with regard to the design of better optimization algorithms for noisy environments.

1 Introduction

Noise is a common factor in most real-world optimization problems. Sources of noise include, to name but a few, physical measurement limitations, the use of stochastic simulation models, incomplete sampling of large spaces, and human-computer interaction. Many of the search methods designed to be able to cope with noise that are in use today can be traced back to either the approach of response surface methodology or to the field of stochastic approximation. The foundations of response surface methodology were established by Box and Wilson [18] who were concerned with minimizing an unknown quadratic objective function disturbed by random noise of constant strength. They proposed constructing a local linear or quadratic model of the objective function by performing experiments in the neighborhood of the current iterate, and to take a step in the direction of steepest descent as derived from this model. According to Torczon and Trosset [52], the designs employed in these experiments would become the patterns in pattern search methods. Response surface methodology thus is a direct precursor of methods such as those of Hooke and Jeeves [27], Spendley, Hext and Himsworth [50], Nelder and Mead [34], Torczon [51], Humphrey and Wilson [28], and Anderson and Ferris [1]. Stochastic approximation on the other hand dates back to work of Robbins and Monro [40] and Kiefer and Wolfowitz [31]. The latter authors suggested the use of finite differencing for obtaining an approximation to the gradient of an unknown, noisy function, and to proceed in direction of this approximate gradient. The implicit filtering algorithm of Gilmore and Kelley [20] and the simultaneous perturbation stochastic approximation algorithm of Spall [47, 49] are derived from this approach.

Evolutionary algorithms are general, nature-inspired heuristics for search and optimization that have developed in relative isolation from the general optimization community. A notable exception is an early book of Schwefel [45] which contained an empirical comparison of classical and evolutionary optimization strategies on a large number of objective functions. See [46] for an updated and translated version of this text. Only in the wake of the recent reflaring of interest in direct search methods asserted by Wright [54] and witnessed by a number of publications [53, 37, 52] have there been renewed attempts to compare methods from both classical and evolutionary optimization [35, 48]. If carefully crafted, such comparisons can serve to reveal the strengths and weaknesses of the respective strategies.

Characteristic for evolutionary computing is the metaphorical use of concepts, principles, and mechanisms underlying natural systems. The three major variants of evolutionary algorithms distinguished by Bäck [9] — genetic algorithms, evolutionary programming, and evolution strategies — have originated independently and differ in their particulars, but share the same basic paradigm. Starting from an initial set of candidate solutions, in an iterative process new candidate solutions are generated from existing ones by means of variation, and selection serves to drive the set of candidate solutions towards increasingly better regions of the search space. Variation is achieved by means of recombination — the act of combining several candidate solutions to form a new one — and mutation — the random modification of parameter values. Adopting the usual terminology, we refer to the objective function as the *fitness function*, to time steps as *generations*, to the set of candidate solutions as a *population*, and to existing and newly generated candidate solutions as *parents* and *offspring*, respectively. In the design of evolution strategies, special emphasis is put on the aspect of adaptability. That is, mechanisms that dynamically adapt expected properties of the set of offspring candidate solutions to local characteristics of the fitness landscape are intrinsic components of the algorithms.

Industrial applications of evolutionary algorithms date back at least to the 1960s, and areas of application today include management, control, design, scheduling, pattern recognition, and decision making. A host of international conferences and several international journals are devoted to the field. In many instances, evolutionary algorithms have proven to be robust and are frequently employed to solve challenging problems where traditional methods are prone to failure, such as optimization problems with highly discontinuous objective functions or where only unreliable data is available. Major reasons for the widespread use of evolutionary algorithms are their universal applicability and the relative ease with which the underlying paradigm is understood and implemented. While in principle for any optimization problem there is a special-purpose algorithm that uses problem-specific knowledge that makes it more efficient, evolutionary algorithms are intended to be general-purpose, easy-to-use, and usually require very little knowledge of the problem at hand. Moreover, as will be seen below, there is both theoretical and empirical evidence that the very concepts that distinguish them from many other optimization strategies — the use of populations, recombination, and emphasis on adaptability — may give evolutionary algorithms performance advantages especially in the presence of noise.

It is the aim of this paper to contrast the robustness in the presence of noise of evolution strategies with that of other common algorithms. Clearly, there are at least two complementary approaches to learning about the behavior of optimization strategies: theoretical and empirical. Theoretical investigations frequently focus on obtaining proofs of convergence under certain general conditions. Sometimes convergence orders or asymptotic bounds on the expected times required to reach a certain vicinity of a global optimum can be obtained. For evolutionary algorithms a number of such results have been derived by Rudolph [43]. However, while undoubtedly useful, such results are often rather coarse as the assumptions that are made are deliberately as weak as possible, and the results that can be derived offer only limited advice for the practitioner who faces the task of choosing one algorithm or the other. Indeed, Powell [37] states that "there seems to be hardly any correlation between the algorithms that are in regular use for practical applications and the algorithms that enjoy guaranteed convergence in theory." Empirical investigations on the other hand frequently evaluate the performance of optimization strategies on a variety of fairly standard objective functions, including ill-scaled, discontinuous, and multimodal functions. Factors like initialization procedures, termination criteria, and the settings of strategy-specific parameters can have decisive influence on the outcome of such experiments. The results do not necessarily bear significance for even the most closely related problems and strategies, and often they contribute little to the understanding of why one strategy works better than another.

The approach pursued in the present paper is of a different nature. Instead of considering either very general classes of problems or specific complex, "realistic" problems, we seek to learn about the behavior of optimization strategies on very simple but nonetheless nontrivial objective functions. Such investigations can be either theoretical or empirical. In either case, the results that can be obtained are interpretable and ideally lead to an understanding of what makes some strategies work better than others. Sometimes, scaling laws that describe the dependence of the performance of a strategy on parameters of the strategy or of the optimization problem can be derived. We believe that such an approach can constitute a useful complement to the other two approaches as it furthers the understanding not only of under what conditions but also of how and why the strategies work or fail. Therefore, after presenting a brief outline of evolution strategies in Section 2, we introduce and motivate a highly symmetric noisy fitness environment that forms the basis for the empirical investigations in Section 3. In Section 4, we conclude with a summary of the insights we have gained and with directions for future research.

2 Evolution Strategies

The purpose of this section is to outline the $(\mu/\rho \overset{+}{,} \lambda)$-ES with isotropic normal mutations as an evolution strategy for the optimization of real-valued functions $f : \mathbb{R}^N \to \mathbb{R}$. Variants of the basic algorithm are touched on briefly where appropriate, and different mutation strength adaptation schemes are discussed. Finally, a few theoretical results concerning the performance in the presence of noise of the algorithm are summarized.

2.1 The Basic $(\mu/\rho \overset{+}{,} \lambda)$-ES

Evolution strategies strive to drive populations of candidate solutions to an optimization problem towards increasingly better regions of the search space by means of variation and selection. A $(\mu/\rho \overset{+}{,} \lambda)$-ES operates with a population $\mathcal{P}$ of μ candidate solutions. Time proceeds in discrete steps and is indicated by a superscript $^{(t)}$ where necessary. In every time step t, a set $\mathcal{Q}^{(t)}$ of λ candidate solutions is created from $\mathcal{P}^{(t)}$ by means of the variational operators of recombination and mutation. The symbol ρ indicates the number of parental candidate solutions involved in the creation of every single offspring candidate solution. The candidate solutions to form the population $\mathcal{P}^{(t+1)}$ of time step $t + 1$ are selected on the basis of their individual fitness — depending on the selection type — either from $\mathcal{P}^{(t)} \cup \mathcal{Q}^{(t)}$ or from $\mathcal{Q}^{(t)}$. Figure 1 illustrates the basic procedure. While generally, initialization schemes and termination criteria are important components of the algorithm, they are frequently application-dependent and irrelevant for what follows. Rather than considering them here, we refer to Bäck [9] for a discussion. The following paragraphs describe the operators used for variation and selection in greater detail.

Variation is crucial for preventing stagnation of the evolutionary search. It can be considered a

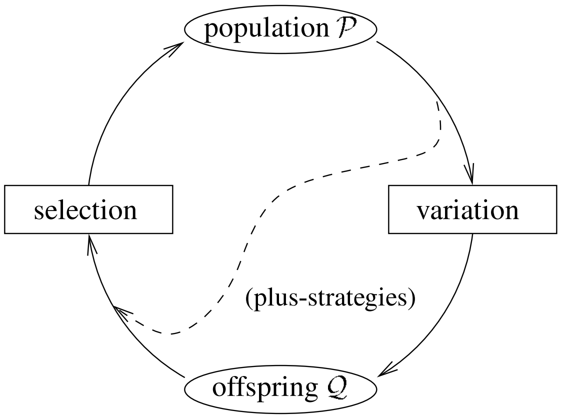

Figure 1: *The basic evolution loop. Variational operators are applied to a population $\mathcal{P}$ of candidate solutions to generate a set $\mathcal{Q}$ of new candidate solutions. Selection is then used to reduce the population to its original size. For comma-strategies, selection is from the set of offspring $\mathcal{Q}$, for plus-strategies it is from the union $\mathcal{P} \cup \mathcal{Q}$ as indicated by the dashed line.*

source of innovation and is usually undirected. The process of creating an offspring candidate solution involves *recombination* and *mutation*. Recombination is a process in which $\rho \leq \mu$ parental candidate solutions are selected at random and their centroid is computed. Mutation consists in adding a random vector drawn from an isotropic normal distribution to that centroid. For a population $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\mu\}$, the set $\mathcal{Q}$ thus consists of offspring candidate solutions

$$\mathbf{y}_i = \frac{1}{\rho} \sum_{j=1}^{\rho} \mathbf{x}_{i_j} + \sigma \mathbf{z}_i \qquad\qquad i = 1, \ldots, \lambda, \tag{1}$$

where the indices i_j are independently drawn with replacement and with equal probability from $\{1, \ldots, \mu\}$, and where the *mutation vectors* $\mathbf{z}_i$ consist of N independent components drawn from a standardized normal distribution. The scalar variable σ determines the expected distance of an offspring candidate solution from the centroid of its parents and is referred to as the *mutation strength*. Only for the special case that $\rho = \mu$, it is commonplace to stipulate that the indices i_j be drawn from $\{1, \ldots, \mu\}$ without replacement. As a consequence, all μ parents are involved in the creation of every single offspring candidate solution and recombination is in fact deterministic. We refer to this case as *global intermediate recombination*.

Selection is the goal-directed component of the evolutionary search and requires the evaluation of the fitness of the candidate solutions. It is deterministic, with the $(\dagger)$ symbolism denoting the two mutually exclusive selection types. Using plus-selection, the μ best of the $\mu + \lambda$ candidate solutions in $\mathcal{P}^{(t)} \cup \mathcal{Q}^{(t)}$ are selected to form $\mathcal{P}^{(t+1)}$. Using comma-selection, the life span of a candidate solution is restricted to a single time step and it is the μ best of the λ candidate solutions in $\mathcal{Q}^{(t)}$ that form $\mathcal{P}^{(t+1)}$. Obviously, comma-strategies require $\lambda \geq \mu$. We refer to the fraction μ/λ as the *truncation ratio*.

The variation and selection operators thus defined are but a small subset of the great number of variants that have been suggested and that are being used. The choice of operators outlined above is motivated both by that they are in widespread use and fairly standard and by their relative mathematical tractability. Among the more common alternatives is *dominant* or *discrete recombination* [13]. Several extensions, such as Cauchy distributed mutations [44], spatially distributed populations [41], or co-evolutionary selection schemes [26] aim at improving global search properties of evolutionary algorithms. Pointers to a great number of such variants can be found in the paper by Bäck, Hammel, and Schwefel [10].

Perhaps the greatest practical shortcoming of the strategy outlined above is its reliance on isotropic mutations. Most objective functions will not exhibit similar scales in different dimensions and will be non-separable. Often, correlated mutations and the use of second-order information will be necessary to achieve satisfactory performance. The scalar variable σ in Equation (1) is to be replaced by an $N \times N$ matrix $\mathbf{S}^{\mathrm{T}}$. Mutation vectors are then normally distributed with positive definite covariance matrix $\mathbf{S}^{\mathrm{T}}\mathbf{S}$. The matrix $\mathbf{S}^{\mathrm{T}}$ can better be adapted to the local structure of the fitness landscape. In the present work, we do not consider correlated mutations as they render the strategies too complicated for mathematical analysis and as they are not useful for the experiments to be conducted in Section 3. However, we also note that the restriction to isotropic mutations may not be as severe as it seems. Ideally, a mutation strength adaptation scheme is able to adapt the matrix $\mathbf{S}^{\mathrm{T}}$ such that locally, fitness functions are rescaled into the spherical function to be introduced in Section 2.3. According to Hansen and Ostermeier [24], using their cumulative mutation strength adaptation mechanism, "Any convex-quadratic function is rescaled into the sphere function". Analyses of the performance of evolution strategies on the sphere can therefore be expected to bear relevance to other fitness functions as well.

2.2 Mutation Strength Adaptation

The mutation strength σ (or, in case of non-isotropic mutations, the matrix $\mathbf{S}^{\mathrm{T}}$) needs to be adapted in the course of the evolutionary search. An ill-adjusted mutation strength can slow down progress by orders of magnitude if it is too low, or lead to divergence if it is too high. A mutation strength adaptation component is therefore an important integral part of evolution strategies. Rather than using fixed schedules as is usual in stochastic approximation, evolutionary algorithms employ dynamic schemes that adjust flexibly to the local characteristics of the fitness landscapes they operate on.

Presumably the first mutation strength adaptation scheme was proposed for the $(1+1)$-ES by Rechenberg [38]. Defining the *success probability* as the probability that an offspring candidate solution is superior to its parent, Rechenberg's scheme relies on the observation that for the fitness functions he investigated, the success probabilities in case of optimally adjusted mutation strength are in a range of values centered at about one fifth, and that generally increasing the mutation strength reduces the success probability and vice versa. Thus, Rechenberg's recommendation was to monitor success probabilities by averaging over a number of time steps, and to increase the mutation strength if the observed estimate of the success probability exceeds 0.2 and to decrease the mutation strength if the success probability is below 0.2.

For multi-parent strategies, at least three different approaches for the adaptation of mutation strengths have been proposed. Nested evolution strategies, propagated by Herdy [25] and Rechenberg [39], adjust strategy parameters such as mutation strengths on a meta level by means of evolutionary optimization. Several populations, each one with their own mutation strength settings, compete with each other for survival. After a number of time steps, the respective progress of the different strategies is determined. The mutation strengths of those populations that have achieved the largest progress are used as a basis for generating mutation strengths for the next round of competition by means of recombination and mutation. Clearly, nested evolution strategies lend themselves well to parallel implementation.

Mutative self-adaptation, due to Rechenberg [38, 39] and Schwefel [45, 46], includes the mutation strengths into the optimization process at the same hierarchical level as the object parameters of the problem. Different candidate solutions have differing mutation strengths. Assuming that favorable mutation strengths are more likely to generate successful offspring than unfavorable ones, selection of favorable mutation strengths is then a by-product of evolution. Beyer [14] has shown in the absence of noise that mutative self-adaptation can guarantee stochastic linear convergence order and lead to near optimal mutation strengths for the $(1, \lambda)$-ES on spherically symmetric objective functions.

Finally, cumulative mutation strength adaptation, introduced by Hansen and Ostermeier [22, 24], is an attempt to "derandomize" the process of mutation strength adjustment. Unlike the previous two methods, cumulative mutation strength adaptation is deterministic rather than evolutive in that it explicitly analyzes statistical features of the selected offspring to drive the strategy parameter settings towards their optimal values. Instead of having differing strategy parameter settings compete with each other at a single time step, cumulative mutation strength adaptation accumulates and analyzes information from a number of time steps. In the absence of noise, it has been demonstrated empirically to reliably adapt mutation covariance matrices on a variety of fitness landscapes. As we deal with isotropic mutations only, instead of outlining the mechanism in its full generality we restrict ourselves to the variant using a single mutation strength.

Cumulative mutation strength adaptation relies on the presumption that if the mutation strength is below its optimal value, then selected consecutive steps tend to be parallel, and that if the mutation strength is too high, consecutive steps tend to be antiparallel. This is plausible intuitively as several steps in the same direction in search space are ideally replaced by a single longer step in that direction, and as consecutive steps that nullify each other are a sign that the step length is too high. So as to be able to reliably detect parallel or antiparallel correlations of progress vectors, information from a number of time steps needs to be accumulated. For the $(\mu/\mu, \lambda)$-ES, the *accumulated progress vector* $\mathbf{s}$ is defined by $\mathbf{s}^{(0)} = \mathbf{0}$ and the recursive relationship

$$\mathbf{s}^{(t+1)} = (1-c)\mathbf{s}^{(t)} + \sqrt{c(2-c)}\sqrt{\mu}\langle\mathbf{z}\rangle^{(t)}, \tag{2}$$

where c is a constant determining how far back the "memory" of the accumulation process reaches and where $\langle\mathbf{z}\rangle$ is the arithmetic mean of the mutation vectors that correspond to those candidate solutions that are selected for survival and is referred to as the *progress vector*. Thus, the vector $\sigma\langle\mathbf{z}\rangle$ connects consecutive centroids of the population. The mutation strength is updated according to

$$\sigma^{(t+1)} = \sigma^{(t)} \exp\left(\frac{\|\mathbf{s}^{(t+1)}\|^2 - N}{2DN}\right), \tag{3}$$

where D denotes a damping constant. Note that the term N in the numerator of the exponent equals the expected value of the χ^2_N-distribution and thus is the mean squared length of the accumulated progress vector if consecutive progress vectors are stochastically independent of each other. If the squared length of the accumulated progress vector is less than N then the mutation strength is decreased. If it is greater than N then the mutation strength is increased. Also note that the prescription Equation (3) for adapting the mutation strength has been changed slightly from the prescription in the original algorithm given by Hansen [22] in that we perform adaptation on the basis of the squared length of the accumulated progress vector rather than on its length. Pending further investigation, the change has been approved of by Hansen and Ostermeier [24]. The constants c and D are usually set to $1/\sqrt{N}$ and $\sqrt{N}$, respectively, according to recommendations made by Hansen [22].

2.3 Performance Analysis of Evolution Strategies in the Presence of Noise

A considerable amount of effort has gone into the analysis of the local performance of evolution strategies in the presence of noise [15, 3]. The goal of such research is to determine how the performance of the strategy scales with parameters of the problem — such as the dimensionality of the search space or the noise strength — and of the search strategy — such as the population size or the mutation strength. Such scaling laws allow for a comparison of different variants of the strategies, provide guidelines for tuning evolution strategies for maximum performance, and offer insights and an understanding of the behavior of the strategies that goes beyond what can be learned from mere experimentation. The most prominent of the objective functions examined is the quadratic sphere

$$f(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{x}, \qquad\qquad \mathbf{x} \in \mathbb{R}^N. \tag{4}$$

This objective function has also been used by Torczon [51] outside of the realm of evolution strategies to empirically evaluate the performance of optimization algorithms. Studying the performance of search strategies on the sphere can be considered a first step towards a quantitative understanding of the behavior of such algorithms. The sphere is arguably the most simple nontrivial function that can be considered, and it seems plausible that an optimization strategy that fails on the sphere is unlikely to succeed when facing more difficult tasks. According to an argument by Rechenberg [38], the sphere can serve as a model for unconstrained optimization problems at a stage where the population of candidate solutions is already in relatively close vicinity to the optimizer. It derives part of its significance from that, ideally, mechanisms for the adaptation of mutation covariance matrices such as cumulative mutation strength adaptation rescale any convex-quadratic function into the sphere. Other fitness functions such as the ridge analyzed by Oyman and Beyer [36] attempt to model features of fitness landscapes in greater distance from the optimizer. Beyer [16] also explores the use of differential geometric methods for studying general quadratic fitness models. However, such fitness functions have additional degrees of freedom and are therefore more difficult to analyze. At the same time the results are less transparent.

Following common practice, we assume that noise inherent in the evaluation of the fitness function is well modeled by an additive, normally distributed term with mean zero. That is, when evaluating the fitness of a candidate solution $\mathbf{x}$, it is not the *ideal fitness* $f(\mathbf{x})$ that we obtain, but a *measured fitness* that is normally distributed with mean $f(\mathbf{x})$ and with standard deviation $\sigma_\epsilon(\mathbf{x})$. Quite naturally, $\sigma_\epsilon(\mathbf{x})$ is referred to as the *noise strength*. Depending on the dependence of the noise strength on the location in search space, quite different behaviors of evolution strategies can be observed. For example, if the noise strength is constant throughout the search space, it is impossible to accurately determine the optimizer's location in search space. As seen in [15, 16], after much time has passed the fitness values of the population of candidate solutions will fluctuate around a nonzero mean that increases with increasing noise strength and that can be decreased by increasing the population size. In what follows, however we consider fitness-proportionate noise strength. That is, we assume that the noise strength for a candidate solution $\mathbf{x}$ being evaluated is proportional to its ideal fitness $f(\mathbf{x})$. Such relative errors of measurement are of great practical importance as they arise for example in connection with physical measurement devices that are accurate up to a certain percentage of the quantity they measure. We will refer to the sphere in connection with fitness-proportionate noise strength as the *noisy sphere*.

Mathematically, the assumption of fitness-proportionate noise strength leads to perfect scale-invariance of the noisy sphere. Provided that the mutation strength adaptation component functions properly, appropriately normalized quantities are independent of the location in search space and, after initialization effects have faded, have time-invariant probability distributions. When plotting the logarithm of the objective function value of the centroid of the population of candidate solutions over the number of objective function evaluations, the resulting graph is linear with some superimposed fluctuations as illustrated in Figure 2. The convergence behavior could thus be termed *stochastic linear convergence*.

As both evolution strategies and the optimization algorithms evaluated empirically in Section 3 exhibit stochastic linear convergence, more finely grained performance measures need to be employed for a comparison of the different strategies. Even for strategies whose convergence orders agree, the

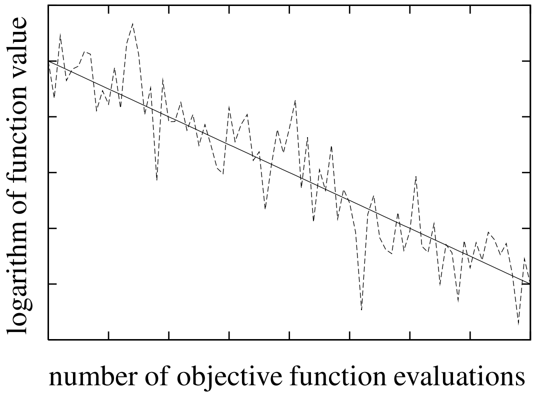

Figure 2: *Typical convergence behavior of direct search strategies on the noisy sphere. The dashed line shows the logarithm of the strategy's function value over the number of objective function evaluations. The efficiency of the search strategy is determined by the negative of the slope of the solid line.*

differences in performance can be significant. Assuming that the computational costs of the search are dominated by the costs involved in the evaluation of the fitness function, the efficiency of a strategy on a serial computer can be considered to be proportional to the negative of the slope of the regression line in Figure 2. For the $(\mu/\rho \stackrel{+}{,} \lambda)$-ES, the x-increment per time step is λ. Denoting the distance from the centroid $\langle \mathbf{x} \rangle$ of the population to the optimizer by R, the fitness of the centroid of the population is $f(\langle \mathbf{x} \rangle) = R^2$ and the y-increment is proportional to the *one-generation gain*

$$\begin{aligned} \Delta_{sphere}^{(t)} &= -\frac{N}{2}\left[\log\left(f(\langle \mathbf{x} \rangle^{(t+1)})\right) - \log\left(f(\langle \mathbf{x} \rangle^{(t)})\right)\right] \\ &= -N \log \frac{R^{(t+1)}}{R^{(t)}}, \end{aligned} \tag{5}$$

where the multiplication with the factor $N/2$ serves the purpose of normalization and reflects the fact the difficulty of numerical search increases with increasing search space dimensionality. For example, the cost of obtaining a gradient estimate by means of finite differencing increases linearly with N. The *efficiency*

$$\eta = \frac{1}{\lambda}\mathrm{E}[\Delta_{sphere}] \tag{6}$$

of a $(\mu/\rho \stackrel{+}{,} \lambda)$-ES is the expected one-generation gain per evaluation of the fitness function. Due to the scale-invariance of the sphere, provided that the mutation strength adaptation functions properly, the efficiency of the $(\mu/\rho \stackrel{+}{,} \lambda)$-ES is constant on the noisy sphere after initialization effects have faded. This is reflected by the fact that the regression line in Figure 2 is straight.

For the $(1+1)$-ES, in [5] systematic overvaluation of the fitness of surviving candidate solutions has been identified as a decisive influence on the performance of the strategy. Those candidate solutions that have a measured fitness that exceeds their ideal fitness are more likely to survive selection than those that are undervalued. In the course of the search, overvaluation builds up and can lead to stagnation of the search due to the failure of the one-fifth-success rule if the noise strength is too high. The same effect can be observed for all optimization strategies where search points can persist indefinitely. Periodic reevaluation of surviving candidate solutions may be required for achieving stochastic linear convergence.

The effect of distributed populations of candidate solutions in the presence of noise has been studied in [7, 3]. It has been seen that the benefit of distributed populations can be traced to a reduction of the noise-to-signal ratio under which the strategy operates. That reduction results from an effective increase in the signal strength that is the sum of a component due to mutation and a component due to the nonzero variance of the population.

Recombination has been seen to reduce the noise-to-signal ratio using a different approach. The analysis presented in [6] relies on a decomposition of mutation vectors into a *central component* that points from the centroid of the parental candidate solutions in direction of the optimizer and a perpendicular *lateral component*. While selection ensures that the central component is responsible for progress in direction of the optimizer, the lateral component represents the "harmful" part of a mutation in that it makes a negative contribution to the fitness of an offspring candidate solution. For the $(\mu/\mu, \lambda)$-ES, introducing normalized quantities

$$\sigma' = \sigma\frac{N}{R} \quad \text{and} \quad \sigma'_\epsilon = \sigma_\epsilon\frac{N}{2R^2},$$

the efficiency law

$$\eta \simeq \frac{1}{\lambda}\left[\frac{\sigma' c_{\mu/\mu,\lambda}}{\sqrt{1+(\sigma'_\epsilon/\sigma')^2}} - \frac{\sigma'^2}{2\mu}\right] \tag{7}$$

derived in [6] is asymptotically exact and provides a good approximation provided that the search space dimensionality N is sufficiently high. The coefficient $c_{\mu/\mu,\lambda}$ equals the expected average of the first μ order statistics out of a sample of λ independent random variates drawn from a standardized normal distribution and can easily be computed numerically. See Arnold, Balakrishnan, and Nagaraja [2] for an introduction to order statistics. The first term in the square brackets of Equation (7) is due to the central components of the mutation vectors, the second term is due to the lateral components. Clearly, the second term places a limit on the mutation strengths that positive efficiency can be achieved with. The presence of the factor μ in the denominator of the second term reflects the presence of for what Beyer [13] has coined the term *genetic repair*. It results from the fact that while the central components of the mutation vectors corresponding to candidate solutions that are selected to survive are correlated, their lateral components are not. The averaging effect implicit in global intermediate recombination thus leads to a reduction in length of the "harmful" lateral components and makes it possible to explore the search space at much higher mutation strengths than would be possible without recombination. In the presence of noise, these increased mutation strengths are especially beneficial as they reduce the noise-to-signal ratio $\vartheta = \sigma'_\epsilon/\sigma'$ under which the $(\mu/\mu, \lambda)$-ES operates that appears in the denominator of the first term. A further result obtained numerically from Equation (7) is that the optimal truncation ratio μ/λ increases from a value of 0.270 in the absence of noise to 0.5 at the point where the noise strength becomes too high for positive efficiency to be possible.

3 Performance of Optimization Strategies on the Noisy Sphere

In this section, a number of common optimization strategies some of which are designed explicitly for optimization in noisy environments are compared by evaluating their respective efficiencies on noisy spheres of several search space dimensionalities. In particular, the strategies considered are the direct pattern search algorithm of Hooke and Jeeves [27], the simplex method of Nelder and Mead [34], the multi-directional search algorithm of Torczon [51], the implicit filtering method of Gilmore and Kelley [20], and a $(\mu/\mu, \lambda)$-ES with isotropic mutations and with cumulative mutation strength adaptation. Clearly, it would be desirable to compare the efficiencies of the various strategies analytically. For the $(\mu/\mu, \lambda)$-ES, analytical results valid for sufficiently large N have been derived in [3]. However, while obtaining such results seems conceivable at least for some of the other strategies, the difficulties involved in such an endeavor can be expected to be considerable, and the analytical comparison remains as a challenge for the future. Meanwhile, we resort to comparing the efficiencies of the strategies on the noisy sphere empirically.

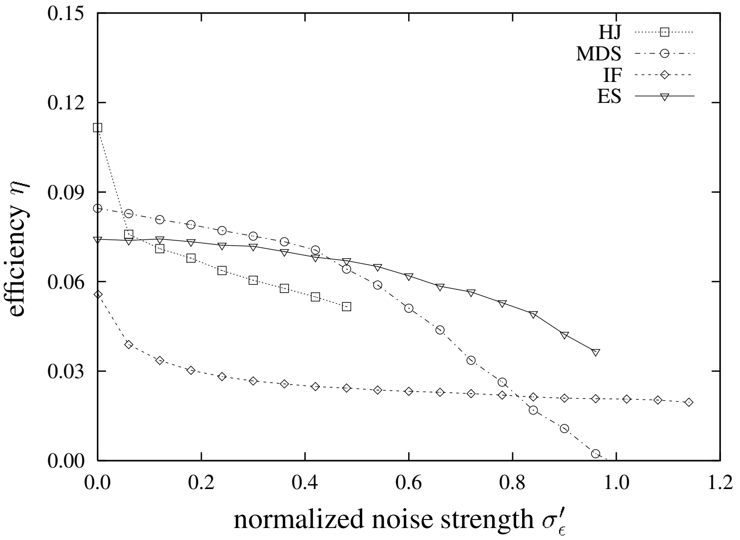

Figure 3: *Efficiencies η of search strategies on the noisy sphere as functions of normalized noise strength σ'_ϵ for search space dimensionality $N = 4$. The curves represent results for direct pattern search (HJ), multi-directional search (MDS), implicit filtering (IF), and a $(2/2, 6)$-ES with cumulative mutation strength adaptation (ES).*

The definition of efficiency from the previous section is easily generalized for arbitrary search strategies. To determine the efficiency of a strategy we average both its one-generation gain and the number of objective function evaluations per time step over $40,000$ time steps and then determine the quotient of the two averages as an estimate for the efficiency. So as to achieve independence of initial conditions, each strategy is run for $2,000$ time steps before the averaging starts. The results of measuring the efficiencies of the search strategies on noisy spheres with search space dimensionalities $N = 4$, 40, and 400 are shown in Figures 3, 4, and 5, respectively, and are described in detail in what follows. Those runs in which stochastic linear convergence was not achieved are excluded from the figures. The corresponding curves thus end abruptly. Due to the highly symmetric nature of the objective function, the reasons for the failure of the various strategies become obvious.

Direct Pattern Search

The direct pattern search algorithm of Hooke and Jeeves [27] is an early example of a direct search strategy. The state of the strategy at time t is described by a base point $\mathbf{x}^{(t)} \in I\!R^N$, a vector $\mathbf{d}^{(t)} \in I\!R^N$ that equals the most recently taken step, and step length $h^{(t)}$. An iteration of the direct pattern search algorithm consists of a pattern step and a sequence of exploratory steps. The pattern step results in intermediate point $\mathbf{y}^{(t)} = \mathbf{x}^{(t)} + \mathbf{d}^{(t)}$, thus duplicating the most recently taken step in the hope that it may speed up the search. Note that if the most recent step was unsuccessful, $\mathbf{d}^{(t)}$ equals the zero vector and effectively no pattern step is being made. The sequence of exploratory moves starts at $\mathbf{y}^{(t)}$ by successively taking steps of length h along the axes $\mathbf{e}_i$, $i = 1, \ldots, N$, of the coordinate system. If such a step leads to an improved objective function value, it is accepted. Otherwise, taking a step in the opposite direction $-\mathbf{e}_i$ is attempted. In our implementation of the algorithm of Hooke and Jeeves we have made use of the improvement suggested by Bell and Pike [12] that aims at reducing the number of objective function evaluations in the sequence of exploratory steps by remembering the most recently taken of the two possible directions as the more promising one. Only after exploratory

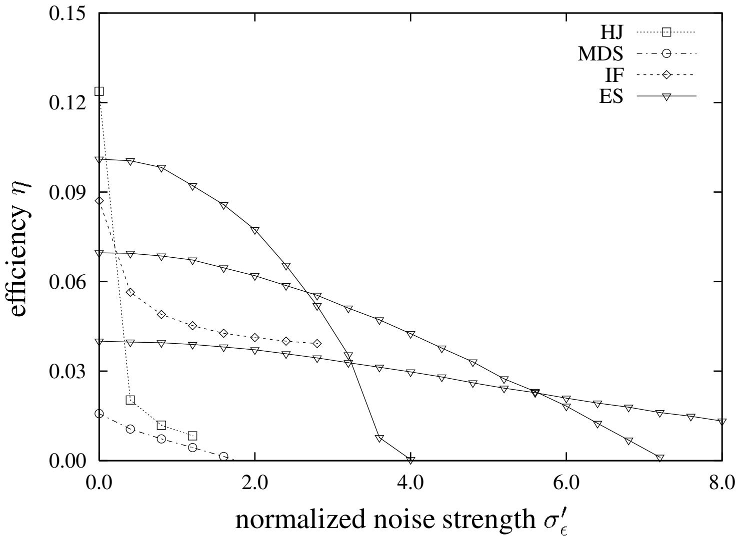

Figure 4: *Efficiencies η of search strategies on the noisy sphere as functions of normalized noise strength σ'_ϵ for search space dimensionality $N = 40$. The curves represent results for direct pattern search (HJ), multi-directional search (MDS), implicit filtering (IF), and, from top-left to bottom-right, a $(3/3, 10)$-ES, a $(6/6, 20)$-ES, and a $(12/12, 40)$-ES with cumulative mutation strength adaptation (ES).*

moves along all axes of the coordinate system have been attempted and a point $\mathbf{z}^{(t)}$ has been reached, the objective function value $f(\mathbf{z}^{(t)})$ is compared with $f(\mathbf{x}^{(t)})$ to decide whether the overall step is to be taken. If it is, then $\mathbf{d}^{(t+1)}$ is set to $\mathbf{z}^{(t)} - \mathbf{x}^{(t)}$ and the new base point $\mathbf{x}^{(t+1)}$ is set to be $\mathbf{z}^{(t)}$. If the overall step is rejected, then $\mathbf{d}^{(t+1)}$ is set to be zero. In addition, if $\mathbf{d}^{(t)}$ already equaled zero before the step, then the step length h is halved.

For the direct pattern search algorithm, as for other search strategies that allow search points to persist indefinitely, the objective function value of the base point needs to be reevaluated periodically so as to achieve stochastic linear convergence. Without reevaluation, the base point is increasingly overvalued and the strategy tends to stagnation. This kind of behavior has been studied analytically for the $(1+1)$-ES in [5]. With reevaluation of the objective function value of the base point in every time step a step of the direct pattern search algorithm involves between $N+1$ and $2N+1$ objective function evaluations. It can be seen from Figures 3, 4, and 5 that the efficiency of the method is quite good in the absence of noise, but that it rather rapidly declines if there is noise present. This is especially true for high-dimensional search spaces. The curves end abruptly as above a certain noise strength stochastic linear convergence is not achieved. Failure of achieving stochastic linear convergence is marked by a rapid decrease in the step length that is easily explained by observing that the "relative step length" h/R determines the "signal strength" under which the strategy operates. Generally, the quotient h/R fluctuates. If it becomes rather small, then the difference in ideal fitness between the old base point and the new candidate base point is small. Thus, the information based on which the decision whether the old base point is to be replaced is almost entirely hidden by noise. As a consequence, in the limit, the old base point is replaced randomly with a probability of one half and the step length is halved with a probability of 25%. This further decrease of the relative step length eventually leads to an exponential decrease of h and to stagnation of the search.

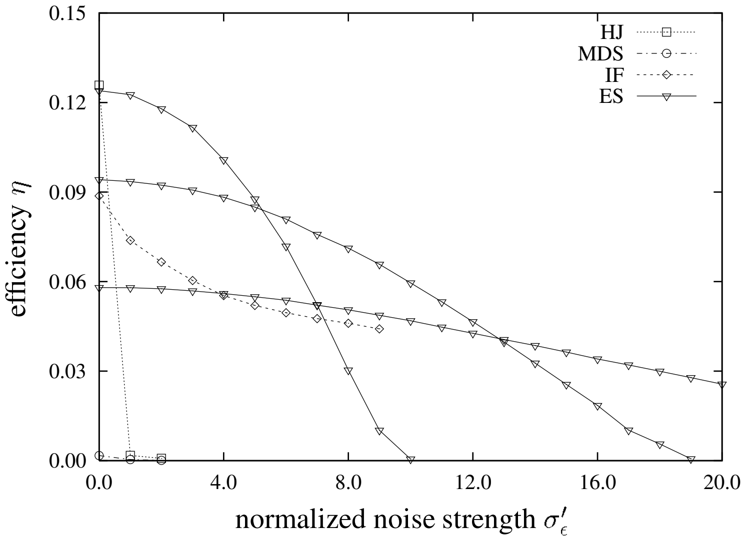

Figure 5: *Efficiencies η of search strategies on the noisy sphere as functions of normalized noise strength σ'_ϵ for search space dimensionality $N = 400$. The curves represent results for direct pattern search (HJ), multi-directional search (MDS), implicit filtering (IF), and, from top-left to bottom-right, a $(6/6, 20)$-ES, a $(12/12, 40)$-ES, and a $(24/24, 80)$-ES with cumulative mutation strength adaptation (ES).*

Simplex Method

Based on an earlier numerical optimization scheme by Spendley, Hext, and Himsworth [50], Nelder and Mead [34] in 1965 devised a simplex method for function minimization. According to Barton and Ivey [11], in 1996 the simplex method of Nelder and Mead was the most popular direct search strategy based on published applications. In 1995, Elster and Neumaier [19] asserted that the simplex method was also the usually recommended and the most frequently used method for noisy function optimization.

A simplex is the convex hull of $N + 1$ points in $\mathbb{R}^N$, where the points satisfy the nondegeneracy condition that the volume of the hull is nonzero. The simplex method attempts to replace the current worst vertex by a new vertex that is generated by a reflection, by an expansion, or by a contraction. Only in case this fails a shrink step is carried out. According to Nelder and Mead, the purpose of these operations is that "the simplex adapts itself to the local landscape, elongating down inclined planes, changing direction on encountering a valley at an angle, and contracting in the neighborhood of a minimum". Depending on the quality of the new points that are generated, the method requires either 1, 2, or $N + 2$ objective function evaluations per time step. A good description of the method along with a discussion of its properties has been published by Wright [54].

In spite of its widespread use, it is well known that the performance of the simplex method frequently is unsatisfactory. A reason for the unsatisfactory performance of the method has been identified in the tendency of the simplices to collapse into a subspace of the search space or to become extremely elongated and distorted in shape even if the local structure of the objective function does not demand that. McKinnon [33] constructed a two-dimensional, strictly convex objective function that has continuous second derivatives where the simplex method converges to a nonoptimal point. The method repeatedly contracts the simplex with the best vertex remaining fixed. The simplices tend to a straight line which is orthogonal to the steepest descent direction. Moreover, Torczon [51] has shown experimentally that the algorithm of Nelder and Mead fails even on the sphere unless the dimensionality N of the search space is very small, and that the presence of noise even worsens the tendency of the method to stagnate at nonoptimal points.

Our observations of the performance of the simplex method of Nelder and Mead on the noisy sphere agree with those of Torczon [51]. For $N = 4$ and zero noise strength, we have obtained an efficiency of about 0.26, thus exceeding that of all other search strategies we have tested but implicit filtering. However, for nonzero noise strength or for $N = 40$ or even $N = 400$, not a single run of the strategy resulted in stochastic linear convergence. We have tested a number of reevaluation strategies and have followed the recommendations of Barton and Ivey [11] with regard to the setting of parameters, but to no discernible effect. There is little use in averaging over multiple samples as any nonzero level of noise can lead to stagnation. Restart strategies to be applied if the simplex becomes too degenerate have been suggested for example by Humphrey and Wilson [28] and by Kelley [29]. We have not employed any such strategies as the breakdowns are so frequent that the behavior of the algorithm would be determined by the restart strategy rather than by the simplex search.

Multi-Directional Search

The multi-directional search method of Torczon [51] is a simplex-based strategy that attempts to overcome the shortcomings of the algorithm of Nelder and Mead. A primary motivation for the new method was the desire for efficiency in a parallel computing environment. An empirical comparison of multi-directional search with the simplex method of Nelder and Mead led Torczon to suggest that "the multi-directional search algorithm may prove to be most useful when the function evaluations are subject to error". A related method has recently been suggested by Anderson and Ferris [1].

Realizing that degenerate simplices are a frequent source of failure of the simplex method of Nelder and Mead, Torczon insisted that for the multi-directional search method the shape of the simplices does not change but that merely their size varies. In our implementation we employ a regular simplex, i.e. one for which all edges have the same length. In contrast to the Nelder-Mead method, not single vertices but the entire simplex is reflected, expanded, and contracted in one time step. An iteration succeeds when it finds a point of strict improvement over the *best* vertex, in contrast to the much weaker condition in a Nelder-Mead iteration of finding a strict improvement compared to the worst point.

In order to achieve stochastic linear convergence, it is necessary to reevaluate the objective function value of the best vertex in every time step. An iteration of the multi-directional search requires $2N + 1$ evaluations of the objective function. It can be seen from Figures 3, 4, and 5 that overall the efficiency of the strategy on the noisy sphere is satisfactory only for $N = 4$. In contrast to most of the other methods considered, the efficiency markedly declines with increasing search space dimensionality and is virtually zero for $N = 400$ even in the absence of noise. Moreover, in contrast to most of the other search strategies, the multi-directional search method never stagnates but rather diverges if the noise strength is too high. This is plausible from the contraction/expansion strategy of the multi-directional search method. An expansion step is made if at least one of the N vertices of both the reflected and the expanded simplices improve on the best vertex of the current simplex. For high noise strength and large N, the possibility of that happening simply due to noise is high. Thus, the strategy expands the simplex more often than it contracts it. While in the absence of noise and for low noise strengths the strategy's suitability for implementation on a parallel computer may be an asset, parallelization would merely lead to faster divergence in the range of noise strengths in which the efficiency is negative. Overall, multi-directional search may be useful in extended, "flat" fitness landscapes; it is of little use in "deep" fitness landscapes like the sphere that require the continuous adjustment of the resolution of the search over several orders of magnitude.

Implicit Filtering

In contrast to the search methods introduced so far, implicit filtering as devised by Gilmore and Kelley [20, 30] relies on explicitly approximating the local gradient of the objective function by means of finite differencing. A brief summary of the algorithm due to Kelley [30] states that "In its simplest unconstrained form, implicit filtering is the steepest descent algorithm with difference gradients, where the difference increment varies as the iteration progresses. Because the gradient is only an approximation, the computed steepest descent direction may fail to be a descent direction and the line search may fail. In this event, the difference increment is reduced." The name "implicit filtering" has been chosen because the method uses the differencing to "step over" the noise at varying levels of resolution, hence implicitly filtering the objective. It is worth noting that Kelley [30] has enhanced the basic algorithm to be described below by a quasi-Newton component that attempts to accumulate second-order information on the objective function in the course of the search. As the sphere is the only objective function we attempt to minimize, we do not consider this extension here. Kelley [30] also provides pointers to optimization problems that implicit filtering has been applied to.

The state of the implicit filtering algorithm at time t is described by a base point $\mathbf{x}$ and a difference increment h. Writing $\mathbf{e}_i$ for the ith unit vector, a central finite difference gradient $\nabla_h f(\mathbf{x})$ with ith component

$$(\nabla_h f(\mathbf{x}))_i = \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} - h\mathbf{e}_i)}{2h}$$

is computed by taking steps of length h in both the positive and negative directions parallel to the axes of the coordinate system. Clearly, this step involves $2N$ evaluations of the objective function. Instead of using central differences, forward differences could be employed; however, Kelley [30] states that the performance of implicit filtering with central difference gradients is far superior to that with forward difference gradients.

Subsequently, a line search in the negative direction of the approximate gradient thus obtained is carried out. In implicit filtering, the algorithm of Armijo is the usual choice for a line search method. That is, starting with $\lambda = \lambda_0$, it is tested whether the condition

$$f(\mathbf{x}) - f(\mathbf{x} - \lambda \nabla_h f(\mathbf{x})) \geq \frac{1}{2}\lambda \|\nabla_h f(\mathbf{x})\|^2$$

holds. If it does hold, then the base point is replaced by $\mathbf{x} - \lambda \nabla_h f(\mathbf{x})$ and the implicit filtering algorithm proceeds to the next iteration. Otherwise, λ is halved and the condition is tested again. If the value of λ has been halved i_{max} times, then the line search is aborted, the difference increment h is halved, and a new iteration of the algorithm is started with the base point left unchanged. The line search maximally requires a number of objective function evaluations that depends on i_{max}, but not on the dimensionality N of the search space.

While the parameter λ_0 that determines the maximum step length and that was set to unity in our experiments is relatively uncritical as long as it is chosen large enough, the maximum number of iterations per line search i_{max} does have a decisive influence on the performance of the algorithm. In our experiments, we have used $i_{max} = 8$. In general, the size h of the difference increments is above its optimal value. Decreasing i_{max} leads to it being decreased faster and therefore to improved efficiency, but at the price of decreased stability.

In the absence of noise, implicit filtering converges much faster than any of the other methods as the exact gradient direction is obtained. Whether the exact optimizer is attained in a single time step thus only depends on the outcome of the line search. Rather than measuring the performance of the implicit filtering algorithm for zero noise strength, we have used the very low noise strength $\sigma'_\epsilon = 0.001$ instead. For nonzero noise strength, Figures 3, 4, and 5 illustrate that implicit filtering does quite well on the sphere, but that above a certain noise strength the strategy fails to achieve stochastic linear convergence. This failure is due to the fact that if h is too small, then the gradient approximation is so unreliable that the line searches fail frequently and h is decreased further, leading to an exponential decrease of h and to stagnation of the search.

Evolution Strategy

As described in Section 2, the state of the $(\mu/\mu, \lambda)$-ES is described by a population $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\mu\}$ of candidate solutions, an accumulated progress vector $\mathbf{s}$, and a mutation strength σ. In every time step, offspring candidate solutions $\mathbf{y}_i = \sum_{j=1}^{\mu} \mathbf{x}_j/\mu + \sigma\mathbf{z}_i$, $i = 1, \ldots, \lambda$, are generated by independently sampling the components of the mutation vectors $\mathbf{z}_i$ from a standardized normal distribution. The fitness function values of the newly generated candidate solutions are evaluated, thus requiring λ fitness function evaluations per time step. Letting $i;\lambda$ denote the index of the offspring candidate solution with the ith highest measured fitness, the iteration of the algorithm is completed by setting $\mathbf{x}_i = \mathbf{y}_{i;\lambda}$ for $i = 1, \ldots, \mu$, updating the accumulated progress vector by multiplication with $(1 - c)$ and subsequent addition of $\sqrt{c(2-c)/\mu}\sum_{i=1}^{\mu} \mathbf{z}_{i;\lambda}$ according to Equation (2), and updating the mutation strength by multiplication with $\exp((\|\mathbf{s}\|^2 - N)/(2DN))$ according to Equation (3).

Figures 3, 4, and 5 show that the $(\mu/\mu, \lambda)$-ES with cumulative mutation strength adaptation is that strategy in the lineup that is the most robust with regard to the effects of noise. For $N = 40$ and especially for $N = 400$, it is the only search strategy that converges reliably for higher levels of noise. As for choosing appropriate population sizes, theoretical results derived in [3] provide useful orientation. Generally, increased population sizes decrease the efficiency for low noise strengths, but afford better robustness for high noise strengths. It can be seen from Figures 4 and 5 that overall the choice is not very critical, and that satisfactory performance can usually be achieved for a range of population size parameter settings.

4 Discussion and Conclusions

In this paper, we have compared empirically the efficiency of a number of optimization strategies on a simple, spherically symmetric objective function. It was assumed that gradient information is not available, and objective function evaluations were subject to Gaussian noise of constant relative strength. Despite its apparent simplicity, due to the presence of noise the environment presented a real challenge to the strategies. Except for the implicit filtering method in the absence of noise, all strategies we considered at best exhibited a linear decrease over time of logarithmic function values.

The strategies studied employ quite different approaches to generating new search points. While most strategies place new search points using deterministic patterns, evolution strategies employ stochastic rules for that task. A further difference consists in the number of new search points that are generated per time step. Some strategies, such as implicit filtering and the multi-directional search method, utilize a number of objective function evaluations that increases linearly with the search space dimensionality N. Other strategies, such as the simplex method of Nelder and Mead (except when taking a shrink step) or evolution strategies, conduct only a typically rather small number of objective function evaluations before taking a step. In general, by increasing the number of objective function evaluations per time step, improved accuracy of the approximation to the gradient direction can be obtained at the price of increased computational costs. For the sphere in the absence of noise, conducting $N+1$ objective function evaluations so as to obtain the exact gradient is optimal. However, if even only little noise is present, the efficiencies of the various strategies are of about the same order of magnitude irrespective of the number of search points they generate per time step. The objective of methods that rely on a large number of objective function evaluations per time step is to be able to make large steps. The objective of strategies like the $(\mu/\mu, \lambda)$-ES or the simultaneous perturbation stochastic approximation approach of Spall [47, 49] on the other hand is to make more steps, possibly in directions that differ considerably from the gradient direction, but that are beneficial in their accumulation. In the presence of noise, the possibilities of making large steps are limited, and approaches that naturally rely on making smaller steps may have an advantage.

Perhaps more crucial for the performance in the presence of noise of optimization strategies than the placement of new search points is the step length control mechanism that determines at what resolution the search space is explored. The step length determines the signal strength under which a strategy operates. If the steps that are made are very small, then the ideal fitness values of search points that are to be compared or that are used to determine an approximation to the gradient direction are minor and noise can dominate the search process. It can frequently be read that finite-difference gradient-based methods exhibit poor performance in the presence of noise. This is true if the difference increments are so small that the differences in function values are hidden by noise. Implicit filtering recognizes this and uses difference increments that are large enough to afford good performance. Similarly, evolution strategies benefit from genetic repair that makes it possible to employ comparatively high mutation strengths. It has been seen that the usual road to failure in the presence of noise for the direct pattern search method as well as for implicit filtering consists in the unwarranted decrease of difference increments or step lengths. Both strategies react with a further decrease of the step length if the noise strength is too high and thus stagnate. Especially for high search space dimensionalities, the $(\mu/\mu, \lambda)$-ES with cumulative mutation strength adaptation fared best of all of the strategies we have tried. At the same time, the cumulative mutation strength adaptation scheme is the only method considered here that explicitly accumulates information for the adaptation of step lengths over a number of time steps. The averaging that is inherent in the accumulation of the progress vectors seems to make cumulative mutation strength adaptation relatively robust with regard to the effects of noise.

Evolutionary algorithms — and in fact randomized algorithms in general — are sometimes referred to as methods of last resort, to be applied only if everything else fails. Our results indicate that if gradients are not available and especially in the presence of noise, they may be more than just that. We have seen that due to their use of populations of candidate solutions, to the benefits of genetic repair resulting from recombination, and to their robust schemes for the adaptation of mutation strengths, evolutionary algorithms are quite rightfully the method of choice in many technical disciplines. Future research should include both the empirical and the theoretical investigation of the behaviors of different search strategies in other simple fitness environments.

Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under grants Be1578/4-2 and Be1578/6-3. The publication of the work was also supported by the DFG as part of the Collaborative Research Center “Computational Intelligence” (SFB 531). Hans-Georg Beyer is a Heisenberg Fellow of the DFG.

References

[1] E. J. Anderson and M. C. Ferris, "A Direct Search Algorithm for Optimization with Noisy Function Evaluations", *SIAM Journal on Optimization*, 11(3), pp. 837-857, (2001).

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

[3] D. V. Arnold, *Local Performance of Evolution Strategies in the Presence of Noise*, Ph.D. thesis, Department of Computer Science, University of Dortmund, (2001).

[4] D. V. Arnold and H.-G. Beyer, "Efficiency and Self-Adaptation of the $(\mu/\mu_I, \lambda)$-ES in a Noisy Environment", in M. Schoenauer et al. (eds.), *Parallel Problem Solving from Nature – PPSN VI*, pp. 39-48, (Springer, Heidelberg, 2000).

[5] D. V. Arnold and H.-G. Beyer, "Local Performance of the $(1 + 1)$-ES in a Noisy Environment", *IEEE Transactions on Evolutionary Computation*, to appear, (2001).

[6] D. V. Arnold and H.-G. Beyer, "Local Performance of the $(\mu/\mu_I, \lambda)$-ES in a Noisy Environment", in W. Martin and W. M. Spears (eds.), *Foundations of Genetic Algorithms* 6, pp. 127-141, (Morgan-Kaufmann, San Francisco, 2001).

[7] D. V. Arnold and H.-G. Beyer, "Investigation of the (μ, λ)-ES in the Presence of Noise", *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, pp. 332-339, (2001).

[8] D. V. Arnold and H.-G. Beyer, "Performance Analysis of Evolution Strategies with Multi-Recombination in High-Dimensional $\mathbb{R}^N$-Search Spaces Disturbed by Noise", *Theoretical Computer Science*, to appear, (2001).

[9] T. Bäck, *Evolutionary Algorithms in Theory and Practice*, (Oxford University Press, New York, 1996).

[10] T. Bäck, U. Hammel, and H.-P. Schwefel, "Evolutionary Computation: Comments on the History and Current State", *IEEE Transactions on Evolutionary Computation*, 1(1), pp. 3-17, (1997).

[11] R. R. Barton and J. S. Ivey, "Nelder-Mead Simplex Modifications for Simulation Optimization", *Management Science*, 42(7), pp. 954-973, (1996).

[12] M. Bell and M. C. Pike, "Remark on Algorithm 178", *Communications of the ACM*, 9, pp. 684-685, (1966).

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

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

[15] H.-G. Beyer, "Evolutionary Algorithms in Noisy Environments: Theoretical Issues and Guidelines for Practice", *Computer Methods in Mechanics and Applied Engineering*, 186, pp. 239-267, (2000).

[16] H.-G. Beyer, *The Theory of Evolution Strategies*, (Springer, Heidelberg, 2001).

[17] D. M. Bortz and C. T. Kelley, "The Simplex Gradient and Noisy Optimization Problems", in J. T. Borggaard et al. (eds.), *Computational Methods in Optimal Design and Control*, pp. 77-90, (Birkhäuser, Boston, 1998).

[18] G. E. P. Box and K. B. Wilson, "On the Experimental Attainment of Optimal Conditions", *Journal of the Royal Statistical Society, Series B*, XIII(1), pp. 1-45, (1951).

[19] C. Elster and A. Neumaier, "A Grid Algorithm for Bound Constrained Optimization of Noisy Functions", *IMA Journal of Numerical Analysis*, 15, pp. 585-608, (1995).

[20] P. Gilmore and C. T. Kelley, "An Implicit Filtering Algorithm for Optimization of Functions with Many Local Minima", *SIAM Journal on Optimization*, 5, pp. 269-285, (1995).

[21] L. S. Gurin and L. A. Rastrigin, "Convergence of the Random Search Method in the Presence of Noise", *ARC*, 26, pp. 1505-1511, (1965).

[22] N. Hansen, *Verallgemeinerte individuelle Schrittweitenregelung in der Evolutionsstrategie*, (Mensch & Buch, Berlin, 1998).

[23] N. Hansen, "Invariance, Self-Adaptation and Correlated Mutations in Evolution Strategies", in M. Schoenauer et al. (eds.), *Parallel Problem Solving from Nature – PPSN VI*, pp. 355-364, (Springer, Heidelberg, 2000).

[24] N. Hansen and A. Ostermeier, "Completely Derandomized Self-Adaptation in Evolution Strategies", *Evolutionary Computation*, 9(2), pp. 159-195, (2001).

[25] M. Herdy, "Reproductive Isolation as Strategy Parameter in Hierarchically Organized Evolution Strategies", in R. Männer and B. Manderick (eds.), *Parallel Problem Solving from Nature – PPSN II*, pp. 207-217, (Elsevier, Amsterdam, 1992).

[26] W. D. Hillis, "Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure", in C. G. Langton et al. (eds.), *Artificial Life* II, pp. 313-324, (Addison-Wesley, Redwood City, 1992).

[27] R. Hooke and T. A. Jeeves, " 'Direct Search' Solution of Numerical and Statistical Problems", *Journal of the ACM*, 8, pp. 212-229, (1961).

[28] D. G. Humphrey and J. R. Wilson, "A Revised Simplex Search Procedure for Stochastic Simulation Response Surface Optimization", *INFORMS Journal on Computing*, 12(4), pp. 272-283, (2000).

[29] C. T. Kelley, "Detection and Remediation of Stagnation in the Nelder-Mead Algorithm Using a Sufficient Decrease Condition", *SIAM Journal on Optimization*, 10(1), pp. 43-55, (1999).

[30] C. T. Kelley, *Iterative Methods for Optimization*, (SIAM, Philadelphia, 1999).

[31] J. Kiefer and J. Wolfowitz, "Stochastic Estimation of a Regression Function", *Annals of Mathematical Statistics*, 23, pp. 462-466, (1952).

[32] R. M. Lewis, V. Torczon, and M. W. Trosset, "Direct Search Methods: Then and Now", *Journal of Computational and Applied Mathematics*, 124, pp. 191-207, (2000).

[33] K. I. M. McKinnon, "Convergence of the Nelder-Mead Simplex Method to a Nonstationary Point", *SIAM Journal on Optimization*, 9(1), pp. 148-158, (1998).

[34] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization", *Computer Journal*, 7, pp. 308-313, (1965).

[35] V. Nissen and J. Propach, "On the Robustness of Population-Based Versus Point-Based Optimization in the Presence of Noise", *IEEE Transactions on Evolutionary Computation*, 2(3), pp. 107-119, (1998).

[36] A. I. Oyman and H.-G. Beyer, "Analysis of the $(\mu/\mu, \lambda)$-ES on the Parabolic Ridge", *Evolutionary Computation*, 8(3), pp. 267-289, (2000).

[37] M. J. D. Powell, "Direct Search Algorithms for Optimization Calculations", *Acta Numerica*, 7, pp. 287-336, (1998).

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

[39] I. Rechenberg, *Evolutionsstrategie '94*, (Frommann-Holzboog, Stuttgart, 1994).

[40] H. Robbins and S. Monro, "A Stochastic Approximation Method", *Annals of Mathematical Statistics*, 29, pp. 400-407, (1951).

[41] G. Rudolph, "Global Optimization by Means of Distributed Evolution Strategies", in H.-P. Schwefel and R. Männer (eds.), *Parallel Problem Solving from Nature – PPSN I*, pp. 209-213, (Springer, Berlin, 1990).

[42] G. Rudolph, "On Correlated Mutations in Evolution Strategies", in R. Männer and B. Manderick (eds.), *Parallel Problem Solving from Nature – PPSN II*, pp. 105-114, (Elsevier, Amsterdam, 1992).

[43] G. Rudolph, *Convergence Properties of Evolutionary Algorithms*, (Dr. Kovač, Hamburg, 1997).

[44] G. Rudolph, "Local Convergence rates of Simple Evolutionary Algorithms with Cauchy Mutations", *IEEE Transactions on Evolutionary Computation*, 1(4), pp. 249-258, (1997).

[45] H.-P. Schwefel, *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, (Birkhäuser, Basel, 1977).

[46] H.-P. Schwefel, *Evolution and Optimum Seeking*, (Wiley, New York, 1995).

[47] J. C. Spall, "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation", *IEEE Transactions on Automatic Control*, 37, pp. 332-341, (1992).

[48] J. C. Spall, S. D. Hill, and D. R. Stark, "Theoretical Comparisons of Evolutionary Computation and Other Optimization Approaches", *Proceedings of the 1999 IEEE Congress on Evolutionary Computation*, pp. 1398-1405, (1998).

[49] J. C. Spall, "Adaptive Stochastic Approximation by the Simultaneous Perturbation Method", *IEEE Transactions on Automatic Control*, 45(10), pp. 1839-1853, (2000).

[50] W. Spendley, G. R. Hext, and F. R. Himsworth, "Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation", *Technometrics*, 4, pp. 441-461, (1962).

[51] V. J. Torczon, *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*, Ph.D. thesis, Department of Mathematical Sciences, Rice University, Houston, (1989).

[52] V. Torczon and M. W. Trosset, "From Evolutionary Operation to Parallel Direct Search: Pattern Search Algorithms for Numerical Optimization", *Computing Science and Statistics*, 29, pp. 396-401, (1998).

[53] M. W. Trosset, "I Know It When I See It: Toward a Definition of Direct Search Methods", *SIAG/OPT Views and News*, 9, pp. 7-10, (1997).

[54] M. H. Wright, "Direct Search Methods: Once Scorned, Now Respectable", in D. F. Griffiths and G. A. Watson (eds.), *Numerical Analysis*, pp. 191-208, (Addison Wesley, Redwood City, 1995).