Evolution Strategies

Nikolaus Hansen, Dirk Arnold, Anne Auger

Evolution Strategies

Nikolaus Hansen, Dirk V. Arnold and Anne Auger

February 11, 2015

Contents

Abstract

Evolution strategies are evolutionary algorithms that date back to the 1960s and that are most commonly applied to black-box optimization problems in continuous search spaces. Inspired by biological evolution, their original formulation is based on the application of mutation, recombination and selection in populations of candidate solutions. From the algorithmic viewpoint, evolution strategies are optimization methods that sample new candidate solutions stochastically, most commonly from a multivariate normal probability distribution. Their two most prominent design principles are unbiasedness and adaptive control of parameters of the sample distribution. In this overview the important concepts of success based step-size control, self-adaptation and derandomization are covered, as well as more recent developments like covariance matrix adaptation and natural evolution strategies. The latter give new insights into the fundamental mathematical rationale behind evolution strategies. A broad discussion of theoretical results includes progress rate results on various function classes and convergence proofs for evolution strategies.

1 Overview

Evolution Strategies [1, 2, 3, 4], sometimes also referred to as Evolutionary Strategies, and *Evolutionary Programming* [5] are search paradigms inspired by the principles of biological evolution. They belong to the family of evolutionary algorithms that address optimization problems by implementing a repeated process of (small) stochastic variations followed by selection: in each generation (or iteration), new offspring (or candidate solutions) are generated from their parents (candidate solutions already visited), their fitness is evaluated, and the better offspring are selected to become the parents for the next generation.

Evolution strategies most commonly address the problem of *continuous black-box optimization*. The search space is the continuous domain, $\mathbb{R}^n$, and solutions in search space are n-dimensional vectors, denoted as $\boldsymbol{x}$. We consider an objective or fitness function $f : \mathbb{R}^n \to \mathbb{R}, \boldsymbol{x} \mapsto f(\boldsymbol{x})$ to be minimized. We make no specific assumptions on f, other than that f can be evaluated for each $\boldsymbol{x}$, and refer to this search problem as *black-box* optimization. The objective is, loosely speaking, to generate solutions ($\boldsymbol{x}$-vectors) with small f-values while using a small number of f-evaluations.[1]

In this context, we present an overview of methods that sample new offspring, or candidate solutions, from normal distributions. Naturally, such an overview is biased by the authors' viewpoints, and our emphasis will be on important design principles and on contemporary evolution strategies that we consider as most relevant in practice or future research. More comprehensive historical overviews can be found elsewhere [6, 7].

In the next section the main principles are introduced and two *algorithm templates* for an evolution strategy are presented. Section 3 presents six evolution strategies that mark important conceptual and algorithmic developments. Section 4 summarizes important theoretical results.

[1] Formally, we like to "converge" to an essential global optimum of f, in the sense that the best $f(\boldsymbol{x})$ value gets arbitrarily close to the *essential infimum* of f (i.e., the smallest f-value for which all larger, i.e. worse f-values have sublevel sets with positive volume).

Symbols and Abbreviations Throughout this chapter, vectors like $\boldsymbol{z} \in \mathbb{R}^n$ are column vectors, their transpose is denoted as $\boldsymbol{z}^\top$, and transformations like $\exp(\boldsymbol{z})$, $\boldsymbol{z}^2$, or $|\boldsymbol{z}|$ are applied component-wise. Further symbols are

- $|\boldsymbol{z}| = (|z_1|, |z_2|, \dots)^\top$ absolute value taken component wise
- $\|\boldsymbol{z}\| = \sqrt{\sum_i z_i^2}$ Euclidean length of a vector
- $\sim$ equality in distribution
- $\propto$ in the limit proportional to
- $\circ$ binary operator giving the component-wise product of two vectors or matrices (Hadamard product), such that for $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ we have $\boldsymbol{a} \circ \boldsymbol{b} \in \mathbb{R}^n$ and $(\boldsymbol{a} \circ \boldsymbol{b})_i = a_i b_i$.
- $\mathbb{1}_\cdot$ the indicator function, $\mathbb{1}_\alpha = 0$ if α is false or 0 or empty, and $\mathbb{1}_\alpha = 1$ otherwise.
- $\lambda \in \mathbb{N}$ number of offspring, offspring population size
- $\mu \in \mathbb{N}$ number of parents, parental population size
- $\mu_w = \left(\sum_{k=1}^{\mu} |w_k|\right)^2 / \sum_{k=1}^{\mu} w_k^2$, the variance effective selection mass or *effective* number of parents, where always $\mu_w \leq \mu$ and $\mu_w = \mu$ if all recombination weights w_k are equal in absolute value
- (1+1) elitist selection scheme with one parent and one offspring, see Section 2.1
- $(\mu \overset{+}{,} \lambda)$, e.g. (1+1) or $(1, \lambda)$, selection schemes, see Section 2.1
- $(\mu/\rho, \lambda)$ selection scheme with recombination (if $\rho > 1$), see Section 2.1
- $\rho \in \mathbb{N}$ number of parents for recombination
- $\sigma > 0$ a step-size and/or standard deviation
- $\boldsymbol{\sigma} \in \mathbb{R}_+^n$ a vector of step-sizes and/or standard deviations

$\varphi \in \mathbb{R}$ a progress measure, see Definition 2 and Section 4.2

$c_{\mu/\mu,\lambda}$ the progress coefficient for the $(\mu/\mu, \lambda)$-ES [8] equals the expected value of the average of the largest μ order statistics of λ independent standard normally distributed random numbers and is in the order of $\sqrt{2\log(\lambda/\mu)}$.

$\mathbf{C} \in \mathbb{R}^{n\times n}$ a (symmetric and positive definite) covariance matrix

$\mathbf{C}^{1/2} \in \mathbb{R}^{n\times n}$ a matrix that satisfies $\mathbf{C}^{1/2}{\mathbf{C}^{1/2}}^\top = \mathbf{C}$ and is symmetric if not stated otherwise. If $\mathbf{C}^{1/2}$ is symmetric, the eigendecomposition $\mathbf{C}^{1/2} = \mathbf{B}\boldsymbol{\Lambda}\mathbf{B}^\top$ with $\mathbf{B}\mathbf{B}^\top = \mathbf{I}$ and diagonal matrix $\boldsymbol{\Lambda}$ exists and we find $\mathbf{C} = \mathbf{C}^{1/2}\mathbf{C}^{1/2} = \mathbf{B}\boldsymbol{\Lambda}^2\mathbf{B}^\top$ as eigendecomposition of $\mathbf{C}$.

$\boldsymbol{e}_i$ the ith canonical basis vector

$f : \mathbb{R}^n \to \mathbb{R}$ fitness or objective function to be minimized

$\mathbf{I} \in \mathbb{R}^{n\times n}$ the identity matrix (identity transformation)

i.i.d. independent and identically distributed

$\mathcal{N}(\boldsymbol{x}, \mathbf{C})$ a multivariate normal distribution with expectation and modal value $\boldsymbol{x}$ and covariance matrix $\mathbf{C}$, see Section 2.4.

$n \in \mathbb{N}$ search space dimension

$\mathcal{P}$ a multiset of individuals, a population

$\boldsymbol{s}, \boldsymbol{s}_\sigma, \boldsymbol{s}_\mathbf{C} \in \mathbb{R}^n$ a search path or evolution path

s, s_k endogenous strategy parameters (also known as control parameters) of a single parent or the kth offspring; they typically parametrize the mutation, for example with a step-size σ or a covariance matrix $\mathbf{C}$

$t \in \mathbb{N}$ time or iteration index

$w_k \in \mathbb{R}$ recombination weights

$\boldsymbol{x}, \boldsymbol{x}^{(t)}, \boldsymbol{x}_k \in \mathbb{R}^n$ solution or object parameter vector of a single parent (at iteration t) or of the kth offspring; an element of the search space $\mathbb{R}^n$ that serves as argument to the fitness function $f : \mathbb{R}^n \to \mathbb{R}$.

$\operatorname{diag} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ the diagonal matrix from a vector

$\exp^\alpha : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}, \mathbf{A} \mapsto \sum_{i=0}^\infty (\alpha\mathbf{A})^i / i!$ is the matrix exponential for $n > 1$, otherwise the exponential function. If $\mathbf{A}$ is symmetric and $\mathbf{B}\boldsymbol{\Lambda}\mathbf{B}^\top = \mathbf{A}$ is the eigendecomposition of $\mathbf{A}$ with $\mathbf{B}\mathbf{B}^\top = \mathbf{I}$ and $\boldsymbol{\Lambda}$ diagonal, we have $\exp(\mathbf{A}) = \mathbf{B}\exp(\boldsymbol{\Lambda})\mathbf{B}^\top = \mathbf{B}\left(\sum_{i=0}^\infty \boldsymbol{\Lambda}^i/i!\right)\mathbf{B}^\top = \mathbf{I} + \mathbf{B}\boldsymbol{\Lambda}\mathbf{B}^\top + \mathbf{B}\boldsymbol{\Lambda}^2\mathbf{B}^\top/2 + \ldots$. Furthermore we have $\exp^\alpha(\mathbf{A}) = \exp(\mathbf{A})^\alpha = \exp(\alpha\mathbf{A})$ and $\exp^\alpha(x) = (\mathrm{e}^\alpha)^x = \mathrm{e}^{\alpha x}$.

2 Main Principles

Evolution strategies derive inspiration from principles of biological evolution. We assume a *population*, $\mathcal{P}$, of so-called *individuals*. Each individual consists of a solution or object parameter vector $\boldsymbol{x} \in \mathbb{R}^n$ (the visible traits) and further endogenous parameters, s (the hidden traits), and an associated fitness value, $f(\boldsymbol{x})$. In some cases the population contains only one individual. Individuals are also denoted as *parents* or *offspring*, depending on the context. In a generational procedure,

1. one or several parents are picked from the population (mating selection) and new offspring are generated by duplication and recombination of these parents;

2. the new offspring undergo mutation and become new members of the population;

3. environmental selection reduces the population to its original size.

Within this procedure, evolution strategies employ the following main principles that are specified and applied in the operators and algorithms further below.

Environmental Selection is applied as so-called *truncation selection*. Based on the individuals' fitnesses, $f(\boldsymbol{x})$, only the μ best individuals from the population survive. In contrast to roulette wheel selection in *genetic algorithms* [9], only fitness *ranks* are used. In evolution strategies, environmental selection is deterministic. In evolutionary programming, like in many other evolutionary algorithms, environmental selection has a stochastic component. Environmental selection can also remove "overaged" individuals first.

Mating Selection and Recombination. Mating selection picks individuals from the population to become new parents. Recombination generates a single new offspring from these parents. Specifically, we differentiate two common scenarios for mating selection and recombination.

fitness-independent mating selection and recombination do not depend on the fitness values of the individuals and can be either deterministic or stochastic. *Environmental* selection is then essential to drive the evolution toward better solutions.

fitness-based mating selection and recombination, where the recombination operator utilizes the fitness ranking of the parents (in a deterministic way). *Environmental* selection can potentially be omitted in this case.

Mutation and Parameter Control. Mutation introduces small, random and unbiased changes to an individual. These changes typically affect all variables. The average size of these changes depends on endogenous parameters that change over time. These parameters are also called control parameters, or *endogenous strategy parameters*, and define the notion of "small", for example via the *step-size* σ. In contrast, *exogenous strategy parameters* are fixed once and for all, for example parent number μ. Parameter control is not always directly inspired by biological evolution, but is an indispensable and central feature of evolution strategies.

Unbiasedness is a generic design principle of evolution strategies. Variation resulting from mutation or recombination is designed to introduce new, unbiased "information". Selection on the other hand biases this information towards solutions with better fitness. Under neutral selection (i.e., fitness independent mating and environmental selection), all variation operators are desired to be unbiased. Maximum exploration and unbiasedness are in accord. Evolution strategies are unbiased in the following respects.

- The type of mutation distribution, the Gaussian or normal distribution, is chosen in order to have rotational symmetry and maximum entropy (maximum exploration) under the given variances. Decreasing the entropy would introduce prior information and therefore a bias.
- Object parameters and endogenous strategy parameters are unbiased under recombination and unbiased under mutation. Typically, mutation has expectation zero.
- Invariance properties avoid a bias towards a specific representation of the fitness function, e.g. representation in a specific coordinate system or using specific fitness values (invariance to strictly monotonic transformations of the fitness values can be achieved). Parameter control in evolution strategies strives for invariance properties [10].

2.1 $(\mu/\rho \overset{+}{,} \lambda)$ Notation for Selection and Recombination

An evolution strategy is an iterative (generational) procedure. In each generation new individuals (offspring) are created from existing individuals (parents). A mnemonic notation is commonly used to describe some aspects of this iteration. The $(\mu/\rho \overset{+}{,} \lambda)$-ES, where μ, ρ and λ are positive integers, also frequently denoted as $(\mu \overset{+}{,} \lambda)$-ES (where ρ remains unspecified) describes the following.

- The parent population contains μ individuals.
- For recombination, ρ (out of μ) parent individuals are used. We have therefore $\rho \leq \mu$.

- λ denotes the number of offspring generated in each iteration.

- $\dagger$ describes whether or not selection is additionally based on the individuals' age. An evolution strategy applies *either* 'plus'- *or* 'comma'-selection. In 'plus'-selection, age is not taken into account and the μ best of $\mu + \lambda$ individuals are chosen. Selection is elitist and, in effect, the parents are the μ all-time best individuals. In 'comma'-selection, individuals die out after one iteration step and only the offspring (the youngest individuals) survive to the next generation. In that case, environmental selection chooses μ parents from λ offspring.

In a (μ, λ)-ES, $\lambda \geq \mu$ must hold and the case $\lambda = \mu$ requires *fitness-based* mating selection or recombination. In a $(\mu + \lambda)$-ES, $\lambda = 1$ is possible and known as *steady-state* scenario.

Occasionally, a subscript to ρ is used in order to denote the type of recombination, e.g. ρ_I or ρ_W for intermediate or weighted recombination, respectively. Without a subscript we tacitly assume intermediate recombination, if not stated otherwise. The notation has also been expanded to include the maximum age, κ, of individuals as (μ, κ, λ)-ES [11], where 'plus'-selection corresponds to $\kappa = \infty$ and 'comma'-selection corresponds to $\kappa = 1$.

2.2 Two Algorithm Templates

Template 1 gives pseudocode for the evolution strategy. Given is a population, $\mathcal{P}$, of at least μ individuals $(\boldsymbol{x}_k, s_k, f(\boldsymbol{x}_k))$, $k = 1, \ldots, \mu$. Vector $\boldsymbol{x}_k \in \mathbb{R}^n$ is a solution vector and s_k contains the control or endogenous strategy parameters, for example a success counter or a step-size that primarily serves to control the mutation of $\boldsymbol{x}$ (in Line 6). The values of s_k may be identical for all k. In each generation, first λ offspring are generated (Lines 3–6), each by recombination of $\rho \leq \mu$ individuals from $\mathcal{P}$ (Line 4), followed by mutation of s (Line 5) and of $\boldsymbol{x}$ (line 6). The new offspring are added to $\mathcal{P}$ (Line 7). Overaged individuals are removed from $\mathcal{P}$ (Line 8), where individuals from the same generation have, by definition, the same age. Finally, the best μ individuals are retained in $\mathcal{P}$ (Line 9).

Template 1 The $(\mu/\rho\overset{+}{,}\lambda)$-ES

```
0 given n, ρ, μ, λ ∈ ℕ+
1 initialize P = {(x_k, s_k, f(x_k)) | 1 ≤ k ≤ μ}
2 while not happy
3     for k ∈ {1, ..., λ}
4         (x_k, s_k) = recombine(select_mates(ρ, P))
5         s_k ← mutate_s(s_k)
6         x_k ← mutate_x(s_k, x_k) ∈ ℝ^n
7     P ← P ∪ {(x_k, s_k, f(x_k)) | 1 ≤ k ≤ λ}
8     P ← select_by_age(P)          // identity for '+'
9     P ← select_μ_best(μ, P)       // by f-ranking
```

Template 2 The $(\mu/\mu\overset{+}{,}\lambda)$-ES

```
0 given n, λ ∈ ℕ+
1 initialize x ∈ ℝ^n, s, P = {}
2 while not happy
3     for k ∈ {1, ..., λ}
4         s_k = mutate_s(s)
5         x_k = mutate_x(s_k, x)
6         P ← P ∪ {(x_k, s_k, f(x_k))}
7     P ← select_by_age(P)          // identity for '+'
8     (x, s) ← recombine(P, x, s)
```

The mutation of the $\boldsymbol{x}$-vector in Line 6 always involves a stochastic component. Lines 4 and 5 may have stochastic components as well.

When select_mates in Line 4 selects $\rho = \mu$ individuals from $\mathcal{P}$, it reduces to the identity. If $\rho = \mu$ and recombination is deterministic, as is commonly the case, the result of recombine is the same *parental centroid* for all offspring. The computation of the parental centroid can be done once before the **for** loop or as the last step of the **while** loop, simplifying the initialization of the algorithm. Template 2 shows the pseudocode in this case.

In Template 2, only a single parental centroid $(\boldsymbol{x}, s)$ is initialized. Mutation takes this parental centroid as input (notice that s_k and $\boldsymbol{x}_k$ in Line 4 and 5 are

now *assigned* rather than *updated*) and "recombination" is postponed to the end of the loop, computing in Line 8 the new parental centroid. While $(\boldsymbol{x}_k, s_k)$ can contain all necessary information for this computation, it is often more transparent to use $\boldsymbol{x}$ and s as additional arguments in Line 8. Selection based on f-values is now limited to mating selection in procedure recombine (that is, procedure select_μ_best is omitted and μ is the number of individuals in $\mathcal{P}$ that are actually used by recombine).

Using a single parental centroid has become the most popular approach, because such algorithms are simpler to formalize, easier to analyze and even perform better in various circumstances as they allow for maximum genetic repair (see below). All instances of evolution strategies given in Section 3 are based on Template 2.

2.3 Recombination Operators

In evolution strategies, *recombination* combines information from several parents to generate a *single* new offspring. Often, *multi-recombination* is used, where more than two parents are recombined ($\rho > 2$). In contrast, in *genetic algorithms* often *two* offspring are generated from the recombination of two parents. In evolutionary programming, recombination is generally not used. The most important recombination operators used in evolution strategies are the following.

Discrete or dominant recombination, denoted by $(\mu/\rho_D \,\dot{+}\, \lambda)$, is also known as *uniform crossover* in genetic algorithms. For each variable (component of the $\boldsymbol{x}$-vector), a single parent is drawn uniformly from all ρ parents to inherit the variable value. For ρ parents that all differ in each variable value, the result is uniformly distributed across ρ^n different $\boldsymbol{x}$-values. The result of discrete recombination depends on the given coordinate system.

Intermediate recombination, denoted by $(\mu/\rho_I \,\dot{+}\, \lambda)$, takes the average value of all ρ parents (computes the center of mass, the centroid).

Weighted multi-recombination [12, 10, 13], denoted by $(\mu/\rho_W \,\dot{+}\, \lambda)$, is a generalization of intermediate recombination, usually with $\rho = \mu$. It takes a weighted average of all ρ parents. The weight values depend on the fitness ranking, in that better parents never get smaller weights than inferior ones. With equal weights, intermediate recombination is recovered. By using comma selection and $\rho = \mu = \lambda$, where some of the weights may be zero, weighted recombination can take over the role of fitness-based environmental selection and *negative* weights become a feasible option [12, 13].[2]

In principle, recombination operators from genetic algorithms, like one-point and two-point crossover or line recombination [14] can alternatively be used. However, they have been rarely applied in evolution strategies.

In evolution strategies, the result of selection and recombination is often deterministic (namely, if $\rho = \mu$ and recombination is intermediate or weighted). This means that eventually all offspring are generated by mutation from the same single solution vector (the parental centroid) as in Template 2. This leads, for given variances, to maximum entropy because all offspring are independently drawn from the same normal distribution.[3]

The role of recombination in general is to keep the variation in a population high. Discrete recombination directly introduces variation by generating different solutions. Their distance resembles the distance between the parents. However, discrete recombination, as it depends on the given coordinate system, relies on separability: it can introduce variation *successfully* only if values of disrupted variables do not strongly depend on each other. Solutions resulting from discrete recombination lie on the vertices of an *axis-parallel* box.

[2] The sum of weights must be either one or zero, or recombination must be applied to the vectors $\boldsymbol{x}_k - \boldsymbol{x}$ and the result added to $\boldsymbol{x}$.

[3] With discrete recombination, the offspring distribution is generated from a mixture of normal distributions with different mean values. The resulting distribution has lower entropy unless it has a larger overall variance.

Intermediate and weighted multi-recombination do not lead to variation within the new population as they result in the same single point for all offspring. However, they do allow the mutation operator to introduce *additional* variation by means of genetic repair [15]: recombinative averaging reduces the effective step length taken in unfavorable directions by a factor of $\sqrt{\mu}$ (or $\sqrt{\mu_w}$ in case of weighted recombination), but leaves the step length in favorable directions essentially unchanged, see also Section 4.2. This may allow increased variation by enlarging mutations by a factor of about μ (or μ_w) as revealed in Eq. (16), to achieve maximal progress.

2.4 Mutation Operators

The mutation operator introduces ("small") variations by adding a point symmetric perturbation to the result of recombination, say a solution vector $\boldsymbol{x} \in \mathbb{R}^n$. This perturbation is drawn from a multivariate normal distribution[4], $\mathcal{N}(\mathbf{0}, \mathbf{C})$, with zero mean (expected value) and covariance matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$. We have $\boldsymbol{x} + \mathcal{N}(\mathbf{0}, \mathbf{C}) \sim \mathcal{N}(\boldsymbol{x}, \mathbf{C})$, meaning that $\boldsymbol{x}$ determines the expected value of the new offspring individual. We also have $\boldsymbol{x} + \mathcal{N}(\mathbf{0}, \mathbf{C}) \sim \boldsymbol{x} + \mathbf{C}^{1/2}\mathcal{N}(\mathbf{0}, \mathbf{I})$, meaning that the linear transformation $\mathbf{C}^{1/2}$ generates the desired distribution from the vector $\mathcal{N}(\mathbf{0}, \mathbf{I})$ that has i.i.d. $\mathcal{N}(0, 1)$ components.[5]

Figure 1 shows different normal distributions in dimension $n = 2$. Their lines of equal density are ellipsoids. Any straight section through the 2-dimensional density recovers a 1-dimensional Gaussian bell. Based on multivariate normal distributions, three different mutation operators can be distinguished.

Spherical/isotropic (Figure 1, left) where the covariance matrix is proportional to the identity, i.e., the mutation distribution follows $\sigma\mathcal{N}(\mathbf{0}, \mathbf{I})$ with step-size $\sigma > 0$. The distribution is spherical and invariant under rotations about its mean. Below, Algorithm 1 uses this kind of mutation.

Axis-parallel (Figure 1, middle) where the covariance matrix is a diagonal matrix, i.e., the mutation distribution follows $\mathcal{N}(\mathbf{0}, \mathrm{diag}(\boldsymbol{\sigma})^2)$, where $\boldsymbol{\sigma}$ is a vector of coordinate-wise standard deviations and the diagonal matrix $\mathrm{diag}(\boldsymbol{\sigma})^2$ has eigenvalues σ_i^2 with eigenvectors $\boldsymbol{e}_i$. The principal axes of the ellipsoid are parallel to the coordinate axes. This case includes the previous isotropic case. Below, Algorithms 2, 3, and 4 implement this kind of mutation distribution.

General (Figure 1, right) where the covariance matrix is symmetric and positive definite (i.e. $\boldsymbol{x}^\top \mathbf{C} \boldsymbol{x} > 0$ for all $\boldsymbol{x} \neq \mathbf{0}$), generally non-diagonal and has $(n^2 + n)/2$ degrees of freedom (control parameters). The general case includes the previous axis-parallel and spherical cases. Below, Algorithms 5 and 6 implement general multivariate normally distributed mutations.

In the first and the second cases, the variations of variables are independent of each other, they are uncorrelated. This limits the usefulness of the operator in practice. The third case is "incompatible" with discrete recombination: for a narrow, diagonally oriented ellipsoid (not to be confused with a diagonal covariance matrix), a point resulting from selection and *discrete* recombination lies within this ellipsoid only if each coordinate is taken from the same parent (which happens with probability $1/\rho^{n-1}$) or from a parent with a very similar value in this coordinate. The narrower the ellipsoid the more similar (i.e. correlated) the value needs to be. As another illustration consider sampling, neutral selection and discrete recombination based on Figure 1, right: after discrete recombination the points $(-2, 2)$ and $(2, -2)$ outside the ellipsoid have the same probability as the points $(2, 2)$ and $(-2, -2)$ inside the ellipsoid.

The mutation operators introduced are **unbiased** in several ways. They are all point-symmetrical and have expectation zero. Therefore, mutation alone will

[4] Besides normally distributed mutations, Cauchy mutations [16, 17, 18] have also been proposed in the context of evolution strategies and evolutionary programming.

[5] Using the normal distribution has several advantages. The $\mathcal{N}(\mathbf{0}, \mathbf{I})$ distribution is the most convenient way to implement an *isotropic* perturbation. The normal distribution is stable: sums of independent normally distributed random variables are again normally distributed. This facilitates the design and analysis of algorithms remarkably. Furthermore, the normal distribution has maximum entropy under the given variances.

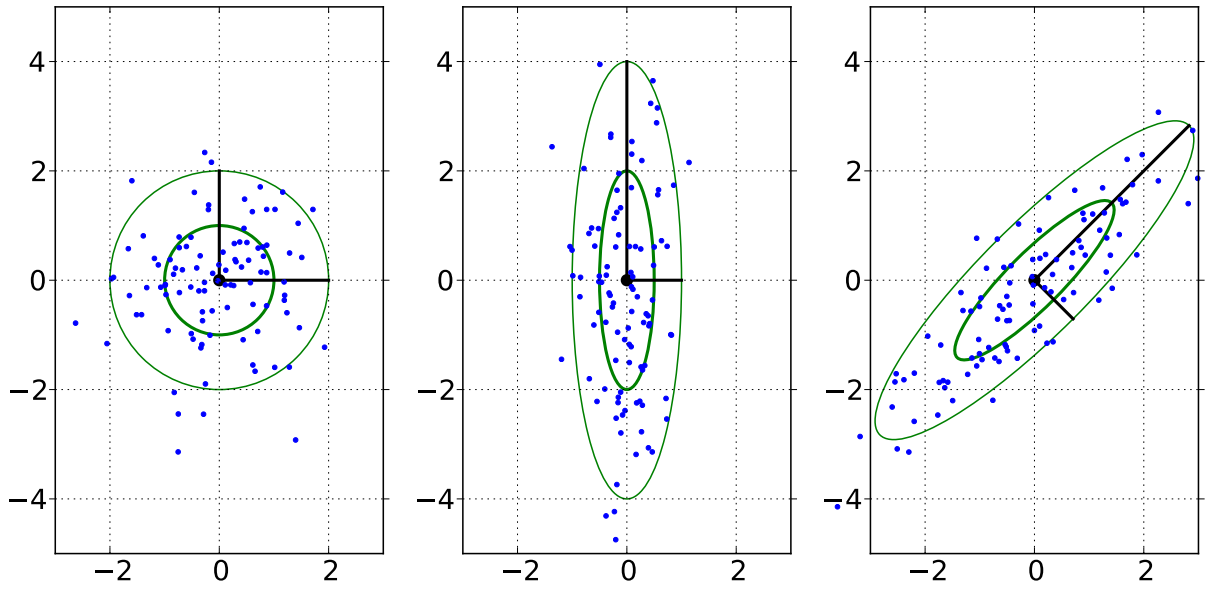

Figure 1: Three 2-dimensional multivariate normal distributions $\mathcal{N}(\mathbf{0}, \mathbf{C}) \sim \mathbf{C}^{1/2}\mathcal{N}(\mathbf{0}, \mathbf{I})$. The covariance matrix $\mathbf{C}$ of the distribution is, from left to right, the identity $\mathbf{I}$ (isotropic distribution), the diagonal matrix $\left(\begin{smallmatrix} 1/4 & 0 \\ 0 & 4 \end{smallmatrix}\right)$ (axis-parallel distribution) and $\left(\begin{smallmatrix} 2.125 & 1.875 \\ 1.875 & 2.125 \end{smallmatrix}\right)$ with the same eigenvalues $(1/4, 4)$ as the diagonal matrix. Shown are in each subfigure the mean at $\mathbf{0}$ as small black dot (a different mean solely changes the axis annotations), two eigenvectors of $\mathbf{C}$ along the principal axes of the ellipsoids (thin black lines), two ellipsoids reflecting the set of points $\{\boldsymbol{x} : (\boldsymbol{x} - \mathbf{0})^\top \mathbf{C}^{-1} (\boldsymbol{x} - \mathbf{0}) \in \{1, 4\}\}$ that represent the 1-σ and 2-σ lines of equal density, and 100 sampled points (however, a few of them are likely to be outside of the area shown).

almost certainly not lead to better fitness values *in expectation*. The isotropic mutation operator features the same distribution along any direction. The general mutation operator is, as long as $\mathbf{C}$ remains unspecified, unbiased towards the choice of a Cartesian coordinate system, i.e. unbiased towards the representation of solutions $\boldsymbol{x}$, which has also been referred to as invariance to affine coordinate system transformations [10]. This however depends on the way how $\mathbf{C}$ is adapted (see below).

3 Parameter Control

Controlling the parameters of the mutation operator is key to the design of evolution strategies. Consider the isotropic operator (Figure 1, left), where the step-size σ is a scaling factor for the random vector perturbation. The step-size controls to a large extent the convergence speed. In situations where larger step-sizes lead to larger expected improvements, a step-size control technique should aim at increasing the step-size (and decreasing it in the opposite scenario).

The importance of step-size control is illustrated with a simple experiment. Consider a spherical function $f(\boldsymbol{x}) = \|\boldsymbol{x}\|^\alpha$, $\alpha > 0$, and a (1+1)-ES with constant step-size equal to $\sigma = 10^{-2}$, i.e. with mutations drawn from $10^{-2}\mathcal{N}(\mathbf{0}, \mathbf{I})$. The convergence of the algorithm is depicted in Fig 2 (constant σ graphs).

We observe, roughly speaking, three stages: up to 600 function evaluations, progress towards the optimum is slow. In this stage the fixed step-size is too small. Between 700 and 800 evaluations, fast progress towards the optimum is observed. In this stage the step-size is close to optimal. Afterwards, the progress

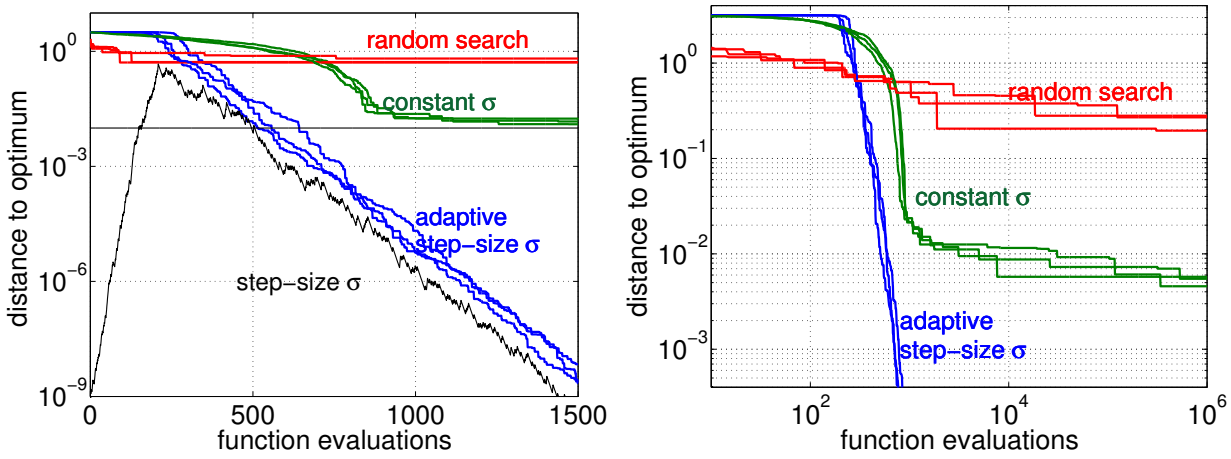

Figure 2: Runs of the (1+1)-ES with constant step-size, of pure random search (uniform in $[-0.2, 1]^{10}$), and of the (1+1)-ES with 1/5th success rule (Algorithm 1) on a spherical function $f(\boldsymbol{x}) = \|\boldsymbol{x}\|^\alpha, \alpha > 0$ (because of invariance to monotonic f-transformation the same graph is observed for any $\alpha > 0$). For each algorithm there are three runs in the left plot and three runs in the right plot. The x-axis is linear in the left and in log-scale in the right hand plot. For the (1+1)-ES with constant step-size, σ equals 10^{-2}. For the (1+1)-ES with 1/5th success rule, the initial step-size is chosen very small to 10^{-9} and the parameter d equals $1 + 10/3$. On the left, also the evolution of the step-size of one of the runs of the (1+1)-ES with 1/5th success rule is shown. All algorithms are initialized at $\mathbf{1}$. Eventually, the (1+1)-ES with 1/5th success rule reveals linear behavior on the left, while the other two algorithms reveal eventually linear behavior in the right hand plot.

decreases and approaches the rate of the pure random search algorithm, well illustrated in the right subfigure. In this stage the fixed step-size is too large and the probability to sample better offspring becomes very small.

The figure also shows runs of the (1+1)-ES with 1/5th success rule step-size control (as described in Section 3.1) and the step-size evolution associated to one of these runs. The initial step-size is far too small and we observe that the adaptation technique increases the step-size in the first iterations. Afterwards, step-size is kept roughly proportional to the distance to the optimum, which is in fact optimal and leads to linear convergence in the left subfigure.

Generally, the goal of parameter control is to drive the endogenous strategy parameters close to their optimal values. These optimal values, as we have seen for the step-size in Figure 2, can significantly change over time or depending on the position in search space. In the most general case, the mutation operator has $(n^2 + n)/2$ degrees of freedom (see Section 2.4). The conjecture is that in the desired scenario lines of equal density of the mutation operator resemble locally the lines of equal fitness [4, p242f]. In case of convex-quadratic fitness functions this resemblance can be perfect and, apart from the step-size, optimal parameters do not change over time (as illustrated in Fig. 3 below).

Control parameters like the step-size can be stored on different "levels". Each individual can have its own step-size value (like in Algorithms 2 and 3), or a single step-size is stored and applied to all individuals in the population. In the latter case, sometimes different populations with different parameter values are run in parallel [19].

In the following, six specific evolution strategies are outlined, each of them representing an important achievement in parameter control.

3.1 The 1/5th Success Rule

The 1/5th success rule for step-size control is based on an important discovery made very early in the research of evolution strategies [1]. A similar rule had also been found independently before in [20]. As a control mechanism in practice, the 1/5th success rule has been mostly superseded by more sophisticated methods. However, its conceptual insight remains remarkably valuable.

Consider a linear fitness function, for example $f : \boldsymbol{x} \mapsto x_1$ or $f : \boldsymbol{x} \mapsto \sum_i x_i$. In this case, any point symmetrical mutation operator has a success probability of 1/2: in one half of the cases, the perturbation will improve the original solution, in one half of the cases the solution will deteriorate. Following from Taylors formula, smooth functions are locally linear, that is, they appear to be more and more linear with decreasing neighborhood size. Therefore, the success probability becomes 1/2 for step-size $\sigma \to 0$. On most non-linear functions, the success rate is indeed a monotonously decreasing function in σ and goes to zero for $\sigma \to \infty$. This suggests to control the step-size by increasing it for large success rates and decreasing it for small ones. This mechanism can drive the step-size close to the optimal value.

Rechenberg [1] investigated two simple but quite different functions, the corridor function

$$f : \boldsymbol{x} \mapsto \begin{cases} x_1 & \text{if } |x_i| \leq 1 \text{ for } i = 2, \ldots, n \\ \infty & \text{otherwise} \end{cases}$$

and the sphere function $f : \boldsymbol{x} \mapsto \sum x_i^2$. He found optimal success rates for the (1+1)-ES with isotropic mutation to be $\approx 0.184 > 1/6$ and $\approx 0.270 < 1/3$, respectively (for $n \to \infty$) [1].[6] This leads to approximately 1/5 as being the success value where to switch between decreasing and increasing the step-size.

Algorithm 1 implements the (1+1)-ES with 1/5th success rule in a simple and effective way [21]. Lines 4–6 implement Line 8 from Template 2, including selection in Line 7. Line 4 in Algorithm 1 updates the step-size σ of the single parent. The step-size does not change if and only if the argument of exp is zero.

[6] Optimality here means to achieve the largest expected approach of the optimum in a single generation.

Algorithm 1 The (1+1)-ES with 1/5th Rule

```
0 given n ∈ ℕ+, d ≈ √(n+1)
1 initialize x ∈ ℝ^n, σ > 0
2 while not happy
3     x_1 = x + σ × 𝒩(0, I)                    // mutation
4     σ ← σ × exp^{1/d}(𝟙_{f(x_1)≤f(x)} − 1/5)
5     if f(x_1) ≤ f(x)                         // select if better
6         x = x_1                              // x-value of new parent
```

While this cannot happen in a single generation, we still can find a stationary point for σ: $\log \sigma$ is unbiased if and only if the expected value of the argument of exp is zero. This is the case if $\mathbb{E}\mathbb{1}_{f(\boldsymbol{x}_1) \leq f(\boldsymbol{x})} = 1/5$, in other words, if the probability of an improvement with $f(\boldsymbol{x}_1) \leq f(\boldsymbol{x})$ is 20%. Otherwise, $\log \sigma$ increases in expectation if the success probability is larger than 1/5 and decreases if the success probability is smaller than 1/5. Hence, Algorithm 1 indeed implements the 1/5th success rule.

3.2 Self-Adaptation

A seminal idea in the domain of evolution strategies is parameter control via *self-adaptation* [3]. In self-adaptation, new control parameter settings are generated similar to new $\boldsymbol{x}$-vectors by recombination and mutation. Algorithm 2 presents an example with adaptation of n coordinate-wise standard deviations (individual step-sizes).

First, for conducting the mutation, random events are drawn in Lines 4–6. In Line 7, the step-size vector for each individual undergoes (i) a mutation common for all components, $\exp(\xi_k)$, and (ii) a component-wise mutation with $\exp(\boldsymbol{\xi}_k)$. These mutations are unbiased, in that $\mathbb{E} \log \boldsymbol{\sigma}_k = \log \boldsymbol{\sigma}$. The mutation of $\boldsymbol{x}$ in Line 8 uses the mutated vector $\boldsymbol{\sigma}_k$. After selection in Line 9, intermediate recombination is applied to compute $\boldsymbol{x}$ and $\boldsymbol{\sigma}$ for the next generation. By taking the average over $\boldsymbol{\sigma}_k$ we have $\mathbb{E}\boldsymbol{\sigma} = \mathbb{E}\boldsymbol{\sigma}_k$ in Line 10. However, the application of mutation *and* recombination on $\boldsymbol{\sigma}$ introduces a moderate bias such that $\boldsymbol{\sigma}$ tends to increase under neutral selection [22].

In order to achieve stable behavior of $\boldsymbol{\sigma}$, the num-

Algorithm 2 The $(\mu/\mu, \lambda)$-σSA-ES

0 **given** $n \in \mathbb{N}_+$, $\lambda \geq 5n$, $\mu \approx \lambda/4 \in \mathbb{N}$, $\tau \approx 1/\sqrt{n}$, $\tau_\mathrm{i} \approx 1/n^{1/4}$
1 **initialize** $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{\sigma} \in \mathbb{R}^n_+$
2 **while** not happy
3 **for** $k \in \{1, \ldots, \lambda\}$
// random numbers i.i.d. for all k
4 $\xi_k = \tau\,\mathcal{N}(0,1)$ // global step-size
5 $\boldsymbol{\xi}_k = \tau_\mathrm{i}\,\mathcal{N}(\mathbf{0}, \mathbf{I})$ // coordinate-wise $\boldsymbol{\sigma}$
6 $\boldsymbol{z}_k = \mathcal{N}(\mathbf{0}, \mathbf{I})$ //$\boldsymbol{x}$-vector change
// mutation
7 $\boldsymbol{\sigma}_k = \boldsymbol{\sigma} \circ \exp(\boldsymbol{\xi}_k) \times \exp(\xi_k)$
8 $\boldsymbol{x}_k = \boldsymbol{x} + \boldsymbol{\sigma}_k \circ \boldsymbol{z}_k$
9 $\mathcal{P} = \mathsf{sel\_}\mu\mathsf{\_best}(\{(\boldsymbol{x}_k, \boldsymbol{\sigma}_k, f(\boldsymbol{x}_k)) \,|\, 1 \leq k \leq \lambda\})$
// recombination
10 $\boldsymbol{\sigma} = \dfrac{1}{\mu} \sum_{\boldsymbol{\sigma}_k \in \mathcal{P}} \boldsymbol{\sigma}_k$
11 $\boldsymbol{x} = \dfrac{1}{\mu} \sum_{\boldsymbol{x}_k \in \mathcal{P}} \boldsymbol{x}_k$

ber of parents μ must be large enough, which is reflected in the setting of λ. A setting of $\tau \approx 1/4$ has been proposed in combination with ξ_k being uniformly distributed across the two values in $\{-1, 1\}$ [2].

3.3 Derandomized Self-Adaptation

Derandomized self-adaptation [23] addresses the problem of selection noise that occurs with self-adaptation of $\boldsymbol{\sigma}$ as outlined in Algorithm 2. Selection noise refers to the possibility that very good offspring may be generated with poor strategy parameter settings and vice versa. The problem occurs frequently and has two origins.

- A small/large component in $|\boldsymbol{\sigma}_k \circ \boldsymbol{z}_k|$ (Line 8 in Algorithm 2) does not necessarily imply that the respective component of $\boldsymbol{\sigma}_k$ is small/large. Selection of $\boldsymbol{\sigma}$ is disturbed by the respective realizations of $\boldsymbol{z}$.
- Selection of a small/large component of $|\boldsymbol{\sigma}_k \circ \boldsymbol{z}_k|$ does not imply that this is necessarily a favorable setting: more often than not, the sign of a component is more important than its size and all other components influence the selection as well.

Due to selection noise, poor values are frequently inherited and we observe stochastic fluctuations of $\boldsymbol{\sigma}$. Such fluctuations can in particular lead to very small values (very large values are removed by selection more quickly). The overall magnitude of these fluctuations can be implicitly controlled via the parent number μ, because intermediate recombination (Line 10 in Algorithm 2) effectively reduces the magnitude of $\boldsymbol{\sigma}$-changes and biases $\log \boldsymbol{\sigma}$ to larger values.

For $\mu \ll n$ the stochastic fluctuations become prohibitive and therefore $\mu \approx \lambda/4 \geq 1.25n$ is chosen to make σ-self-adaptation reliable.

Derandomization addresses the problem of selection noise on $\boldsymbol{\sigma}$ directly without resorting to a large parent number. The derandomized $(1, \lambda)$-σSA-ES is outlined in Algorithm 3 and addresses selection noise twofold. Instead of introducing new variations in $\boldsymbol{\sigma}$ by means of $\exp(\boldsymbol{\xi}_k)$, the variations from $\boldsymbol{z}_k$ are directly used for the mutation of $\boldsymbol{\sigma}$ in Line 7. The variations are dampened compared to their use in the mutation of $\boldsymbol{x}$ (Line 6) via d and d_i, thereby mimicking the effect of intermediate recombination on $\boldsymbol{\sigma}$ [23, 24]. The order of the two mutation equations becomes irrelevant.

For Algorithm 3 also a $(\mu/\mu, \lambda)$ variant with recombination is feasible. However, in particular in the $(\mu/\mu_I, \lambda)$-ES, σ-self-adaptation tends to generate too small step-sizes. A remedy for this problem is to use non-local information for step-size control.

3.4 Non-Local Derandomized Step-Size Control (CSA)

When using self-adaptation, step-sizes are associated with individuals and selected based on the fitness of each individual. However, step-sizes that serve individuals well by giving them a high likelihood to be selected are generally not step-sizes that maximize the progress of the entire population or the parental

Algorithm 3 Derandomized $(1, \lambda)$-σSA-ES

0 **given** $n \in \mathbb{N}_+$, $\lambda \approx 10$, $\tau \approx 1/3$, $d \approx \sqrt{n}$, $d_\mathrm{i} \approx n$
1 **initialize** $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{\sigma} \in \mathbb{R}^n_+$
2 **while** not happy
3 **for** $k \in \{1, \ldots, \lambda\}$
// random numbers i.i.d. for all k
4 $\xi_k = \tau \mathcal{N}(0, 1)$
5 $\boldsymbol{z}_k = \mathcal{N}(\mathbf{0}, \mathbf{I})$
// mutation, re-using random events
6 $\boldsymbol{x}_k = \boldsymbol{x} + \exp(\xi_k) \times \boldsymbol{\sigma} \circ \boldsymbol{z}_k$
7 $\boldsymbol{\sigma}_k = \boldsymbol{\sigma} \circ \exp^{1/d_\mathrm{i}}\left(\frac{|\boldsymbol{z}_k|}{\mathbb{E}|\mathcal{N}(0,1)|} - \mathbf{1}\right)$
7b $\times \exp^{1/d}(\xi_k)$
8 $(\boldsymbol{x}_1, \boldsymbol{\sigma}_1, f(\boldsymbol{x}_1)) \leftarrow$ select_single_best$(\{(\boldsymbol{x}_k, \boldsymbol{\sigma}_k, f(\boldsymbol{x}_k)) \mid 1 \le k \le \lambda\})$
// assign new parent
9 $\boldsymbol{\sigma} = \boldsymbol{\sigma}_1$
10 $\boldsymbol{x} = \boldsymbol{x}_1$

Algorithm 4 The $(\mu/\mu, \lambda)$-ES with Search Path

0 **given** $n \in \mathbb{N}_+$, $\lambda \in \mathbb{N}$, $\mu \approx \lambda/4 \in \mathbb{N}$, $c_\sigma \approx \sqrt{\mu/(n+\mu)}$, $d \approx 1 + \sqrt{\mu/n}$, $d_\mathrm{i} \approx 3n$
1 **initialize** $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{\sigma} \in \mathbb{R}^n_+$, $\boldsymbol{s}_\sigma = \mathbf{0}$
2 **while** not happy
3 **for** $k \in \{1, \ldots, \lambda\}$
4 $\boldsymbol{z}_k = \mathcal{N}(\mathbf{0}, \mathbf{I})$ // i.i.d. for each k
5 $\boldsymbol{x}_k = \boldsymbol{x} + \boldsymbol{\sigma} \circ \boldsymbol{z}_k$
6 $\mathcal{P} \leftarrow$ sel_μ_best$(\{(\boldsymbol{x}_k, \boldsymbol{z}_k, f(\boldsymbol{x}_k)) \mid 1 \le k \le \lambda\})$
// recombination and parent update
7 $\boldsymbol{s}_\sigma \leftarrow (1 - c_\sigma)\, \boldsymbol{s}_\sigma +$
7b $\sqrt{c_\sigma\,(2 - c_\sigma)}\frac{\sqrt{\mu}}{\mu} \sum_{\boldsymbol{z}_k \in \mathcal{P}} \boldsymbol{z}_k$
8 $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} \circ \exp^{1/d_\mathrm{i}}\left(\frac{|\boldsymbol{s}_\sigma|}{\mathbb{E}|\mathcal{N}(0,1)|} - \mathbf{1}\right)$
8b $\times \exp^{c_\sigma/d}\left(\frac{\|\boldsymbol{s}_\sigma\|}{\mathbb{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\|} - 1\right)$
9 $\boldsymbol{x} = \frac{1}{\mu} \sum_{\boldsymbol{x}_k \in \mathcal{P}} \boldsymbol{x}_k$

centroid $\boldsymbol{x}$. We will see later that, for example, the optimal step-size may increase linearly with μ (Section 4.2 and Eq. (16)). With self-adaptation on the other hand, the step-size of the μth best offspring is typically even smaller than the step-size of the best offspring. Consequently, Algorithm 3 assumes often too small step-sizes and can be considerably improved by using non-local information about the evolution of the population. Instead of single (local) mutation steps $\boldsymbol{z}$, an exponentially fading record, $\boldsymbol{s}_\sigma$, of mutation steps is taken. This record, referred to as *search path* or *evolution path*, can be pictured as a sequence or sum of consecutive successful $\boldsymbol{z}$-steps that is non-local in time and space. A search path carries information about the interrelation between single steps. This information can improve the adaptation and search procedure remarkably. Algorithm 4 outlines the $(\mu/\mu_I, \lambda)$-ES with *cumulative path length control*, also denoted as *cumulative step-size adaptation* (CSA), and additionally with non-local *individual* step-size adaptation [25, 26].

In the $(\mu/\mu, \lambda)$-ES with search path, Algorithm 4, the factor ξ_k for changing the overall step-size has disappeared (compared to Algorithm 3) and the update of $\boldsymbol{\sigma}$ is postponed until after the **for** loop. Instead of the additional random variate ξ_k, the length of the search path $\|\boldsymbol{s}_\sigma\|$ determines the global step-size change in Line 8b. For the individual step-size change, $|\boldsymbol{z}_k|$ is replaced by $|\boldsymbol{s}_\sigma|$.

Using a search path is justified in two ways. First, it implements a low-pass filter for selected $\boldsymbol{z}$-steps, removing high frequency (most likely noisy) information. Second, and more importantly, it utilizes information that is otherwise lost: even if all single steps have the same length, the length of $\boldsymbol{s}_\sigma$ can vary, because it depends on the correlation between the directions of $\boldsymbol{z}$-steps. If single steps point into similar directions, the path will be up to almost $\sqrt{2/c_\sigma}$ times longer than a single step and the step-size will increase. If they oppose each other the path will be up to almost $\sqrt{c_\sigma/2}$ times shorter and the step-size will decrease. The same is true for single components of $\boldsymbol{s}_\sigma$.

The factors $\sqrt{c_\sigma\,(2 - c_\sigma)}$ and $\sqrt{\mu}$ in Line 7b guaranty unbiasedness of $\boldsymbol{s}_\sigma$ under neutral selection, as

usual.

All evolution strategies described so far are of somewhat limited value, because they feature only isotropic or axis-parallel mutation operators. In the remainder we consider methods that entertain not only an n-dimensional step-size vector $\boldsymbol{\sigma}$, but also correlations between variables for the mutation of $\boldsymbol{x}$.

3.5 Addressing Dependencies Between Variables

The evolution strategies presented so far sample the mutation distribution independently in each component of *the given* coordinate system. The lines of equal density are either spherical or axis-parallel ellipsoids (compare Figure 1). This is a major drawback, because it allows to solve problems with a long or elongated valley efficiently only if the valley is aligned with the coordinate system. In this section we discuss evolution strategies that allow to traverse non-axis-parallel valleys efficiently by sampling distributions with correlations.

Full Covariance Matrix Algorithms that adapt the complete covariance matrix of the mutation distribution (compare Section 2.4) are *correlated mutations* [3], the *generating set adaptation* [26], the *covariance matrix adaptation* (*CMA*) [27], a mutative invariant adaptation [28], and some instances of *natural evolution strategies* [29, 30, 31]. Correlated mutations and some natural evolution strategies are however not invariant under changes of the coordinate system [32, 10, 31]. In the next sections we outline two evolution strategies that adapt the full covariance matrix reliably and are invariant under coordinate system changes: the covariance matrix adaptation evolution strategy (CMA-ES) and the exponential natural evolution strategy (xNES).

Restricted Covariance Matrix Algorithms that adapt non-diagonal covariance matrices, but are restricted to certain matrices, are the *momentum adaptation* [33], *direction adaptation* [26], *main vector adaptation* [34], and *limited memory CMA-ES* [35]. These variants are limited in their capability to shape the mutation distribution, but they might be advantageous for larger dimensional problems, say larger than a hundred.

3.6 Covariance Matrix Adaptation (CMA)

The *covariance matrix adaptation evolution strategy* (*CMA-ES*) [27, 10, 36] is a de facto standard in continuous domain evolutionary computation. The CMA-ES is a natural generalization of Algorithm 4 in that the mutation ellipsoids are not constrained to be axis-parallel, but can take on a general orientation. The CMA-ES is also a direct successor of the *generating set adaptation* [26], replacing self-adaptation to control the overall step-size with cumulative step-size adaptation.

The $(\mu/\mu_W, \lambda)$-CMA-ES is outlined in Algorithm 5. Two search paths are maintained, $\boldsymbol{s}_\sigma$ and $\boldsymbol{s}_\mathbf{c}$. The first path, $\boldsymbol{s}_\sigma$, accumulates steps in the coordinate system where the mutation distribution is isotropic and which can be derived by scaling in the principal axes of the mutation ellipsoid only. The path generalizes $\boldsymbol{s}_\sigma$ from Algorithm 4 to non-diagonal covariance matrices and is used to implement cumulative step-size adaptation, CSA, in Line 10 (resembling Line 8b in Algorithm 4). Under neutral selection, $\boldsymbol{s}_\sigma \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\log \sigma$ is unbiased.

The second path, $\boldsymbol{s}_\mathbf{c}$, accumulates steps, disregarding σ, in the given coordinate system.[7] The covariance matrix update consists of a rank-one update, based on the search path $\boldsymbol{s}_\mathbf{c}$, and a rank-$\mu$ update with μ nonzero recombination weights w_k. Under neutral selection the expected covariance matrix equals the covariance matrix before the update.

The updates of $\boldsymbol{x}$ and $\mathbf{C}$ follow a common principle. The mean $\boldsymbol{x}$ is updated such that the likelihood of successful offspring to be sampled again is maximized (or increased if $c_\mathrm{m} < 1$). The covariance matrix $\mathbf{C}$ is updated such that the likelihood of successful steps $(\boldsymbol{x}_k - \boldsymbol{x})/\sigma$ to appear again, or the likelihood

[7] Whenever $\boldsymbol{s}_\sigma$ is large and therefore σ is increasing fast, the coefficient h_σ prevents $\boldsymbol{s}_\mathbf{c}$ from getting large and quickly changing the distribution shape via $\mathbf{C}$. Given $h_\sigma \equiv 1$, under neutral selection $\boldsymbol{s}_\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$. The coefficient c_h in line 11 corrects for the bias on $\boldsymbol{s}_\mathbf{c}$ introduced by events $h_\sigma = 0$.

Algorithm 5 The $(\mu/\mu_W, \lambda)$-CMA-ES

0 **given** $n \in \mathbb{N}_+$, $\lambda \geq 5$, $\mu \approx \lambda/2$, $w_k = w'(k)/\sum_k^\mu w'(k)$, $w'(k) = \log(\lambda/2 + 1/2) - \log \mathrm{rank}(f(\boldsymbol{x}_k))$, $\mu_w = 1/\sum_k^\mu w_k^2$, $c_\sigma \approx \mu_w/(n + \mu_w)$, $d \approx 1 + \sqrt{\mu_w/n}$, $c_\mathbf{c} \approx (4 + \mu_w/n)/(n + 4 + 2\mu_w/n)$, $c_1 \approx 2/(n^2 + \mu_w)$, $c_\mu \approx \mu_w/(n^2 + \mu_w)$, $c_\mathrm{m} = 1$

1 **initialize** $\boldsymbol{s}_\sigma = \mathbf{0}$, $\boldsymbol{s}_\mathbf{c} = \mathbf{0}$, $\mathbf{C} = \mathbf{I}$, $\sigma \in \mathbb{R}_+^n$, $\boldsymbol{x} \in \mathbb{R}^n$

2 **while** not happy

3 **for** $k \in \{1, \ldots, \lambda\}$

4 $\boldsymbol{z}_k = \mathcal{N}(\mathbf{0}, \mathbf{I})$ // i.i.d. for all k

5 $\boldsymbol{x}_k = \boldsymbol{x} + \sigma \mathbf{C}^{1/2} \times \boldsymbol{z}_k$

6 $\mathcal{P} = \mathsf{sel\_}\mu\mathsf{\_best}(\{(\boldsymbol{z}_k, f(\boldsymbol{x}_k)) \,|\, 1 \leq k \leq \lambda\})$

7 $\boldsymbol{x} \leftarrow \boldsymbol{x} + c_\mathrm{m} \sigma \, \mathbf{C}^{1/2} \sum_{\boldsymbol{z}_k \in \mathcal{P}} w_k \boldsymbol{z}_k$

8 $\boldsymbol{s}_\sigma \leftarrow (1 - c_\sigma)\, \boldsymbol{s}_\sigma + \sqrt{c_\sigma\,(2 - c_\sigma)}\sqrt{\mu_w} \sum_{\boldsymbol{z}_k \in \mathcal{P}} w_k \boldsymbol{z}_k$ // search path for σ

9 $\boldsymbol{s}_\mathbf{c} \leftarrow (1 - c_\mathbf{c})\, \boldsymbol{s}_\mathbf{c} + h_\sigma \sqrt{c_\mathbf{c}\,(2 - c_\mathbf{c})}\sqrt{\mu_w} \sum_{\boldsymbol{z}_k \in \mathcal{P}} w_k \mathbf{C}^{1/2} \boldsymbol{z}_k$ // search path for $\mathbf{C}$

10 $\sigma \leftarrow \sigma \, \exp^{c_\sigma/d} \left(\dfrac{\|\boldsymbol{s}_\sigma\|}{\mathbb{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1 \right)$

11 $\mathbf{C} \leftarrow (1 - c_1 + c_h - c_\mu)\, \mathbf{C} + c_1 \boldsymbol{s}_\mathbf{c} \boldsymbol{s}_\mathbf{c}^\top + c_\mu \sum_{\boldsymbol{z}_k \in \mathcal{P}} w_k \mathbf{C}^{1/2} \boldsymbol{z}_k (\mathbf{C}^{1/2} \boldsymbol{z}_k)^\top$

where $h_\sigma = \mathbb{1}_{\|\boldsymbol{s}_\sigma\|^2/n < 2 + 4/(n+1)}$, $c_h = c_1(1 - h_\sigma^2) c_\mathbf{c}(2 - c_\mathbf{c})$, and $\mathbf{C}^{1/2}$ is the unique symmetric positive definite matrix obeying $\mathbf{C}^{1/2} \times \mathbf{C}^{1/2} = \mathbf{C}$. All c-coefficients are ≤ 1.

to sample (in direction of) the path $\boldsymbol{s}_\mathbf{c}$, is increased. A more fundamental principle for the equations is given in the next section.

Using not only the μ best but all λ offspring can be particularly useful for the "rank-μ" update of $\mathbf{C}$ in line 11 where negative weights w_k for inferior offspring are advisable. Such an update has been introduced as *active CMA* [37].

The factor c_m in Line 7 can be equally written as a mutation scaling factor $\kappa = 1/c_\mathrm{m}$ in Line 5, compare [38]. This means that the actual mutation steps are larger than the inherited ones, resembling the derandomization technique of damping step-size changes to address selection noise as described in Section 3.3.

An elegant way to replace Line 10 is

$$\sigma \leftarrow \sigma \, \exp^{(c_\sigma/d)/2} \left(\frac{\|\boldsymbol{s}_\sigma\|^2}{n} - 1 \right) \tag{1}$$

and often used in theoretical investigations of this update as those presented in Section 4.2.

A single run of the $(5/5_W, 10)$-CMA-ES on a convex-quadratic function is shown in Fig. 3. For sake of demonstration the initial step-size is chosen far too small (a situation that should be avoided in practice) and increases quickly for the first 400 f-evaluations. After no more than 5500 f-evaluations the adaptation of $\mathbf{C}$ is accomplished. Then the eigenvalues of $\mathbf{C}$ (square roots of which are shown in the lower left) reflect the underlying convex-quadratic function and the convergence speed is the same as on the sphere function and about 60% of the speed of the (1+1)-ES as observed in Fig. 2. The resulting convergence speed is about ten thousand times faster than without adaptation of $\mathbf{C}$ and at least one thousand times faster compared to any of the algorithms from the previous sections.

3.7 Natural Evolution Strategies

The idea of using natural gradient learning [39] in evolution strategies has been proposed in [29] and further pursued in [40, 31]. *Natural evolution strategies* (NES) put forward the idea that the update of all distribution parameters can be based on the same fundamental principle. NES have been proposed as a more principled alternative to CMA-ES and characterized by operating on Cholesky factors of a covariance matrix. Only later was it discovered that also CMA-ES implements the underlying NES principle of natural gradient learning [41, 31].

For simplicity, let the vector θ represent all parameters of the distribution to sample new offspring. In case of a multivariate normal distribution as above, we have a bijective transformation between θ and mean and covariance matrix of the distribution, $\theta \leftrightarrow (\boldsymbol{x}, \sigma^2 \mathbf{C})$.

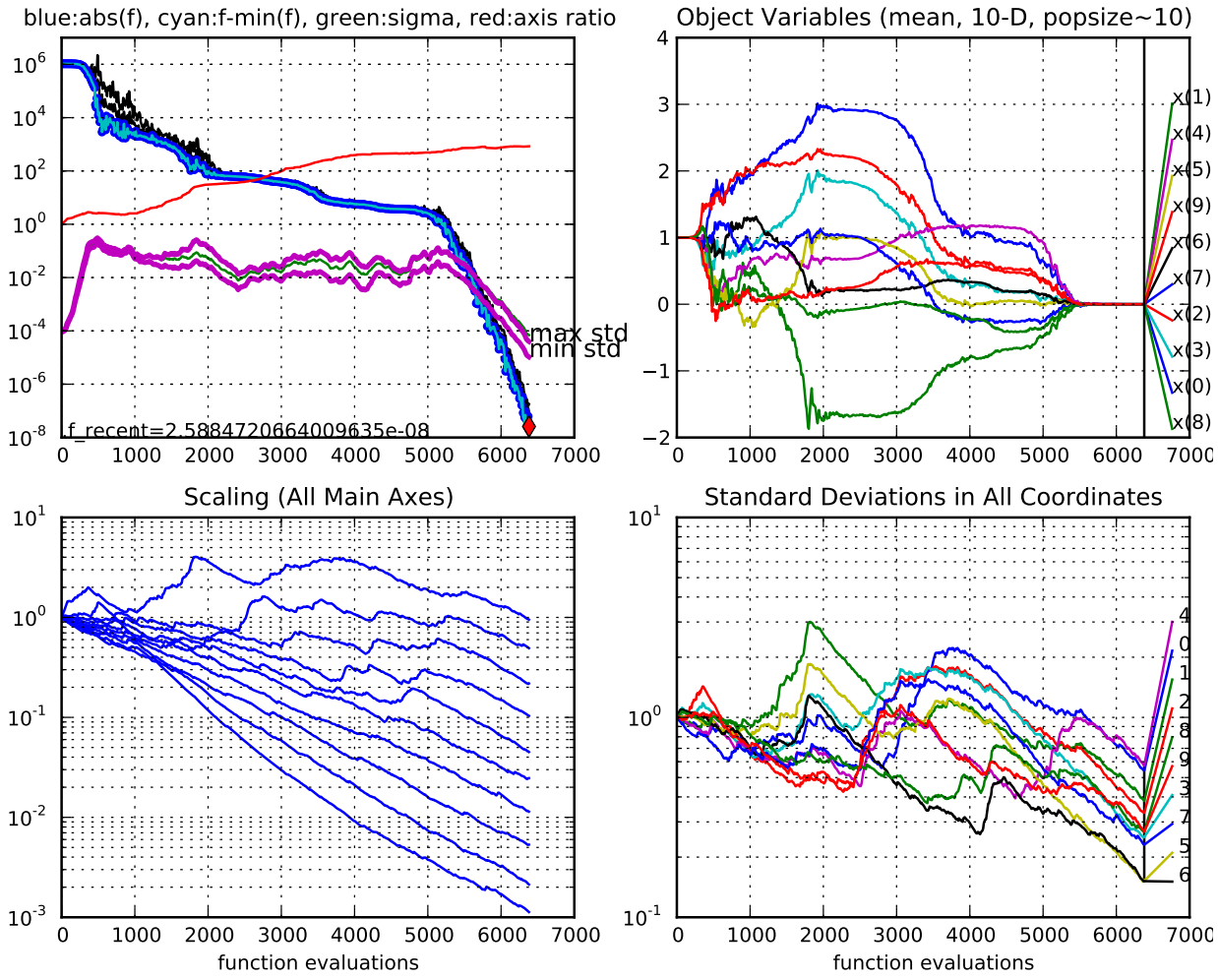

Figure 3: A single run of the $(5/5_W, 10)$-CMA-ES on the rotated ellipsoid function $\boldsymbol{x} \mapsto \sum_{i=1}^{n} \alpha_i^2 y_i^2$ with $\alpha_i = 10^{3(i-1)/(n-1)}$, $\boldsymbol{y} = \mathbf{R}\boldsymbol{x}$, where $\mathbf{R}$ is a random matrix with $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$, for $n = 10$. Shown is the evolution of various parameters against the number of function evaluations. Upper left: best (thick blue line), median and worst fitness value that reveal the final convergence phase after about 5500 function evaluations where the ellipsoid function has been reduced to the simple sphere; minimal and maximal coordinate-wise standard deviation of the mutation distribution and in between (mostly hidden) the step-size σ that is initialized far too small and increases quickly in the beginning, that increases afterwards several times again by up to one order of magnitude and decreases with maximal rate during the last 1000 f-evaluations; axis ratio of the mutation ellipsoid (square root of the condition number of $\mathbf{C}$) that increases from 1 to 1000 where the latter corresponds to α_n/α_1. Lower left: sorted principal axis lengths of the mutation ellipsoid disregarding σ (square roots of the sorted eigenvalues of $\mathbf{C}$, see also Fig. 1) that adapt to the (local) structure of the underlying optimization problem; they finally reflect almost perfectly the factors α_i^{-1} up to a constant factor. Upper right: $\boldsymbol{x}$ (distribution mean) that is initialized with all ones and converges to the global optimum in zero while correlated movements of the variables can be observed. Lower right: standard deviations in the coordinates disregarding σ (square roots of diagonal elements of $\mathbf{C}$) showing the $\mathbf{R}$-dependent projections of the principal axis lengths into the given coordinate system. The straight lines to the right of the vertical line at about 6300 only annotate the coordinates and do not reflect measured data.

We consider a probability density $p(.|\theta)$ over $\mathbb{R}^n$ parametrized by θ, a non-increasing function $W_\theta^f : \mathbb{R} \to \mathbb{R}$,[8] and the expected $W_{\theta'}^f$-transformed fitness [42]

$$J(\theta) = \mathbb{E}(W_{\theta'}^f(f(\boldsymbol{x}))) \qquad \boldsymbol{x} \sim p(.|\theta)$$
$$= \int_{\mathbb{R}^n} W_{\theta'}^f(f(\boldsymbol{x}))\, p(\boldsymbol{x}|\theta) \mathrm{d}\boldsymbol{x} \ , \qquad (2)$$

where the expectation is taken under the given sample distribution. The maximizer of J w.r.t. $p(.|\theta)$ is, for any fixed $W_{\theta'}^f$, a Dirac distribution concentrated on the minimizer of f. A natural way to update θ is therefore a gradient ascent step in $\nabla_\theta J$ direction. However, the "vanilla" gradient $\nabla_\theta J$ depends on the specific parametrization chosen in θ. In contrast, the natural gradient, denoted by $\widetilde{\nabla}_\theta$, is associated to the Fisher metric that is intrinsic to p and independent of the chosen θ-parametrization. Developing $\widetilde{\nabla}_\theta J(\theta)$ under mild assumptions on f and $p(.|\theta)$ by exchanging differentiation and integration, recognizing that the gradient $\widetilde{\nabla}_\theta$ does not act on $W_{\theta'}^f$, using the log-likelihood trick $\widetilde{\nabla}_\theta p(.|\theta) = p(.|\theta)\, \widetilde{\nabla}_\theta \ln p(.|\theta)$ and finally setting $\theta' = \theta$ yields[9]

$$\widetilde{\nabla}_\theta J(\theta) = \mathbb{E}\left(W_\theta^f(f(\boldsymbol{x}))\, \widetilde{\nabla}_\theta \ln p(\boldsymbol{x}|\theta)\right) \ . \qquad (3)$$

A Monte-Carlo approximation of the expected value by the average finally yields the comparatively simple expression

$$\widetilde{\nabla}_\theta J(\theta) \approx \frac{1}{\lambda} \sum_{k=1}^{\lambda} \overbrace{W_\theta^f(f(\boldsymbol{x}_k))}^{\text{preference weight}} \underbrace{\widetilde{\nabla}_\theta \ln p(\boldsymbol{x}_k|\theta)}_{\text{intrinsic candidate direction}} \qquad (4)$$

for a natural gradient update of θ, where $\boldsymbol{x}_k \sim p(.|\theta)$ is sampled from the current distribution. The natural gradient can be computed as $\widetilde{\nabla}_\theta = \mathbf{F}_\theta^{-1} \nabla_\theta$, where $\mathbf{F}_\theta$ is the Fisher information matrix expressed in θ-coordinates. For the multivariate Gaussian distribution, $\widetilde{\nabla}_\theta \ln p(\boldsymbol{x}_k|\theta)$ can indeed be easily expressed and computed efficiently. We find that in CMA-ES (Algorithm 5), the rank-μ update (Line 11 with $c_1 = 0$) and the update in Line 7 are natural gradient updates of $\mathbf{C}$ and $\boldsymbol{x}$, respectively [41, 31], where the kth largest w_k is a consistent estimator for the kth largest $W_\theta^f(f(\boldsymbol{x}_k))$ [42].

While the natural gradient does not depend on the parametrization of the distribution, a finite step taken in the natural gradient direction does. This becomes relevant for the covariance matrix update, where natural evolution strategies take a different parametrization than CMA-ES. Starting from Line 11 in Algorithm 5, we find for $c_1 = c_h = 0$

$$\mathbf{C} \leftarrow (1 - c_\mu)\, \mathbf{C} + c_\mu \sum_{\boldsymbol{z}_k \in \mathcal{P}} w_k \mathbf{C}^{1/2} \boldsymbol{z}_k (\mathbf{C}^{1/2} \boldsymbol{z}_k)^\top$$
$$= \mathbf{C}^{1/2} \left((1 - c_\mu)\, \mathbf{I} + c_\mu \sum_{\boldsymbol{z}_k \in \mathcal{P}} w_k \boldsymbol{z}_k \boldsymbol{z}_k^\top \right) \mathbf{C}^{1/2}$$
$$\overset{\sum w_k = 1}{=} \mathbf{C}^{1/2} \left(\mathbf{I} + c_\mu \sum_{\boldsymbol{z}_k \in \mathcal{P}} w_k \left(\boldsymbol{z}_k \boldsymbol{z}_k^\top - \mathbf{I} \right) \right) \mathbf{C}^{1/2}$$
$$\overset{c_\mu \ll 1}{\approx} \mathbf{C}^{1/2} \exp^{c_\mu} \left(\sum_{\boldsymbol{z}_k \in \mathcal{P}} w_k \left(\boldsymbol{z}_k \boldsymbol{z}_k^\top - \mathbf{I} \right) \right) \mathbf{C}^{1/2} \ . \qquad (5)$$

The term bracketed between the matrices $\mathbf{C}^{1/2}$ in the lower three lines is a multiplicative covariance matrix update expressed in the natural coordinates, where the covariance matrix is the identity and $\mathbf{C}^{1/2}$ serves as coordinate system transformation into the given coordinate system. Only the lower two lines of Eq. (5) do not rely on the constraint $\sum_k w_k = 1$ in order to satisfy a stationarity condition on $\mathbf{C}$.[10] The last line of Eq. (5) is used in the *exponential natural evolution*

[8] More specifically, $W_\theta^f : y \mapsto w(\mathrm{Pr}_{\boldsymbol{z} \sim p(.|\theta)}(f(\boldsymbol{z}) \leq y))$ computes the p_θ-quantile, or cumulative distribution function, of $f(\boldsymbol{z})$ with $\boldsymbol{z} \sim p(.|\theta)$ at point y, composed with a non-increasing predefined weight function $w : [0, 1] \to \mathbb{R}$ (where $w(0) > w(1/2) = 0$ is advisable). The value of $W_\theta^f(f(\boldsymbol{x}))$ is invariant under strictly monotonous transformations of f. For $\boldsymbol{x} \sim p(.|\theta)$ the distribution of $W_\theta^f(f(\boldsymbol{x})) \sim w(\mathcal{U}[0, 1])$ depends only on the predefined w; it is independent of θ and f and therefore also (time-)invariant under θ-updates. Given λ samples $\boldsymbol{x}_k$, we have the rank-based consistent estimator $W_\theta^f(f(\boldsymbol{x}_k)) \approx w\left(\frac{\mathrm{rank}(f(\boldsymbol{x}_k)) - 1/2}{\lambda}\right)$.

[9] We set $\theta' = \theta$ because we will estimate $W_{\theta'}$ using the current samples that are distributed according to $p(.|\theta)$

[10] For a given $\mathbf{C}$ on the right hand side of Eq. (5), we have under neutral selection the stationarity condition $\mathbb{E}(\mathbf{C}_{\mathrm{new}}) = \mathbf{C}$ for the first three lines and $\mathbb{E}(\log(\mathbf{C}_{\mathrm{new}})) = \log(\mathbf{C})$ for the last line, where log is the inverse of the matrix exponential exp.

Algorithm 6 The Exponential NES (xNES)

0 **given** $n \in \mathbb{N}_+$, $\lambda \geq 5$, $w_k = w'(k)/\sum_k |w'(k)|$, $w'(k) \approx \log(\lambda/2 + 1/2) - \log \operatorname{rank}(f(\boldsymbol{x}_k))$, $\eta_c \approx (5 + \lambda)/(5\, n^{1.5}) \leq 1$, $\eta_\sigma \approx \eta_c$, $\eta_x \approx 1$

1 **initialize** $\mathbf{C}^{1/2} = \mathbf{I}$, $\sigma \in \mathbb{R}_+$, $\boldsymbol{x} \in \mathbb{R}^n$

2 **while** not happy

3 **for** $k \in \{1, \ldots, \lambda\}$

4 $\boldsymbol{z}_k = \mathcal{N}(\mathbf{0}, \mathbf{I})$ // i.i.d. for all k

5 $\boldsymbol{x}_k = \boldsymbol{x} + \sigma \mathbf{C}^{1/2} \times \boldsymbol{z}_k$

6 $\mathcal{P} = \{(\boldsymbol{z}_k, f(\boldsymbol{x}_k)) \mid 1 \leq k \leq \lambda\}$

7 $\boldsymbol{x} \leftarrow \boldsymbol{x} + \eta_x \sigma\, \mathbf{C}^{1/2} \sum_{\boldsymbol{z}_k \in \mathcal{P}} w_k \boldsymbol{z}_k$

8 $\sigma \leftarrow \sigma \exp^{\eta_\sigma/2} \left(\sum_{\boldsymbol{z}_k \in \mathcal{P}} w_k \left(\frac{\|\boldsymbol{z}_k\|^2}{n} - 1 \right) \right)$

9 $\mathbf{C}^{1/2} \leftarrow \mathbf{C}^{1/2} \times \exp^{\eta_c/2} \left(\sum_{\boldsymbol{z}_k \in \mathcal{P}} w_k \left(\boldsymbol{z}_k \boldsymbol{z}_k^\top - \frac{\|\boldsymbol{z}_k\|^2}{n} \mathbf{I} \right) \right)$

strategy, xNES [31] and guarantees positive definiteness of $\mathbf{C}$ even with negative weights, independent of c_μ and of the data $\boldsymbol{z}_k$. The xNES is depicted in Algorithm 6.

In xNES, sampling is identical to CMA-ES and environmental selection is omitted entirely. Line 8 resembles the step-size update in (1). Comparing the updates more closely, with $c_\sigma = 1$ Eq. (1) uses

$$\frac{\mu_w \left\| \sum_k w_k \boldsymbol{z}_k \right\|^2}{n} - 1$$

whereas xNES uses

$$\sum_k w_k \left(\frac{\|\boldsymbol{z}_k\|^2}{n} - 1 \right)$$

for updating σ. For $\mu = 1$ the updates are the same. For $\mu > 1$, the latter only depends on the lengths of the $\boldsymbol{z}_k$, while the former depends on their lengths and directions. Finally, xNES expresses the update Eq. (5) in Line 9 on the Cholesky factor $\mathbf{C}^{1/2}$, which does not remain symmetric in this case ($\mathbf{C} = \mathbf{C}^{1/2} \times \mathbf{C}^{1/2\,\top}$ still holds). The term $-\|\boldsymbol{z}_k\|^2/n$ keeps the determinant of $\mathbf{C}^{1/2}$ (and thus the trace of $\log \mathbf{C}^{1/2}$) constant and is of rather cosmetic nature. Omitting the term is equivalent to using $\eta_\sigma + \eta_c$ instead of η_σ in line 8.

The exponential natural evolution strategy is a very elegant algorithm. Like CMA-ES it can be interpreted as an incremental Estimation of Distribution Algorithm [43]. However, it performs generally inferior compared to CMA-ES because it does not use search paths for updating σ and $\mathbf{C}$.

3.8 Further Aspects

Internal Parameters Adaptation and self-adaptation address the control of the most important internal parameters in evolution strategies. Yet, all algorithms presented have hidden and exposed parameters in their implementation. Many of them can be set to reasonable and robust default values. The population size parameters μ and λ however change the search characteristics of an evolution strategy significantly. Larger values, in particular for parent number μ, often help address highly multimodal or noisy problems more successfully.

In practice, several experiments or restarts are advisable, where different initial conditions for $\boldsymbol{x}$ and σ can be employed. For exploring different population sizes, a schedule with increasing population size, IPOP, is advantageous [44, 45, 46], because runs with larger populations take typically more function evaluations. Preceding long runs (large μ and λ) with short runs (small μ and λ) leads to a smaller (relative) impairment of the later runs than vice versa.

Internal computational complexity Algorithms presented in Sections 3.1–3.4 that sample isotropic or axis-parallel mutation distributions have an internal computational complexity linear in the dimension. The internal computational complexity of CMA-ES and xNES is, for constant population size, cubic in the dimension due to the update of $\mathbf{C}^{1/2}$. Typical implementations of the CMA-ES however have quadratic complexity, as they implement a lazy update scheme for $\mathbf{C}^{1/2}$, where $\mathbf{C}$ is decomposed into $\mathbf{C}^{1/2}\mathbf{C}^{1/2}$ only after about n/λ iterations. An exact quadratic update for CMA-ES

has also been proposed [47]. While never considered in the literature, a lazy update for xNES to achieve quadratic complexity seems feasible as well.

Invariance Selection and recombination in evolution strategies are based solely on the ranks of offspring and parent individuals. As a consequence, the behavior of evolution strategies is invariant under order-preserving (strictly monotonous) transformations of the fitness function value. In particular, all spherical unimodal functions belong to the same function class, which the convex-quadratic sphere function is the most pronounced member of. This function is more thoroughly investigated in Section 4.

All algorithms presented are invariant under translations and Algorithms 1, 5 and 6 are invariant under rotations of the coordinate system, provided that the initial $\boldsymbol{x}$ is translated and rotated accordingly.

Parameter control can introduce yet further invariances. All algorithms presented are scale invariant due to step-size adaptation. Furthermore, ellipsoidal functions that are in the reach of the mutation operator of the evolution strategies presented in Sections 3.2 to 3.7 are eventually transformed, effectively, into spherical functions. These evolution strategies are invariant under the respective affine transformations of the search space, given the initial conditions are chosen respectively.

Variants Evolution strategies have been extended and combined with other approaches in various ways. We mention here constraint handling [48, 49], fitness surrogates [50], multi-objective variants [51, 52], and exploitation of fitness values [53].

4 Theory

There is ample *empirical* evidence, that on many unimodal functions evolution strategies with step-size control, as those outlined in the previous section, converge fast and with probability one to the global optimum. Convergence proofs supporting this evidence are discussed in Section 4.3. On multimodal functions on the other hand, the probability to converge to the global optimum (in a single run of the same strategy) is generally smaller than one (but larger than zero), as suggested by observations and theoretical results [54]. Without parameter control on the other hand, elitist strategies always converge to the essential global optimum,[11] however at a much slower rate (compare random search in Figure 2).

In this section we use a time index t to denote iteration and assume, for notational convenience and without loss of generality (due to translation invariance), that the optimum of f is in $\boldsymbol{x}^* = \mathbf{0}$. This simplifies writing $\boldsymbol{x}^{(t)} - \boldsymbol{x}^*$ to simply $\boldsymbol{x}^{(t)}$ and then $\|\boldsymbol{x}^{(t)}\|$ measures the distance to the optimum of the parental centroid in time step t.

Linear convergence plays a central role for evolution strategies. For a *deterministic* sequence $\boldsymbol{x}^{(t)}$ linear convergence (towards zero) takes place if there exists a $c > 0$ such that

$$\lim_{t\to\infty} \frac{\|\boldsymbol{x}^{(t+1)}\|}{\|\boldsymbol{x}^{(t)}\|} = \exp(-c) \tag{6}$$

which means, loosely speaking, that for t large enough, the distance to the optimum decreases in every step by the constant factor $\exp(-c)$. Taking the logarithm of Eq. (6), then exchanging the logarithm and the limit and taking the Cesàro mean yields

$$\lim_{T\to\infty} \underbrace{\frac{1}{T}\sum_{t=0}^{T-1} \log \frac{\|\boldsymbol{x}^{(t+1)}\|}{\|\boldsymbol{x}^{(t)}\|}}_{=\frac{1}{T}\log \|\boldsymbol{x}^{(T)}\|/\|\boldsymbol{x}^{(0)}\|} = -c \ . \tag{7}$$

For a *sequence of random vectors* we define linear convergence based on Eq. (7) as follows.

Definition 1 (linear convergence)**.** *The sequence of random vectors $\boldsymbol{x}^{(t)}$ converges almost surely linearly to $\mathbf{0}$ if there exists a $c > 0$ such that*

$$\begin{aligned} -c &= \lim_{T\to\infty} \frac{1}{T} \log \frac{\|\boldsymbol{x}^{(T)}\|}{\|\boldsymbol{x}^{(0)}\|} \quad a.s. \\ &= \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \log \frac{\|\boldsymbol{x}^{(t+1)}\|}{\|\boldsymbol{x}^{(t)}\|} \quad a.s. \end{aligned} \tag{8}$$

[11] On a bounded domain and with mutation variances bounded away from zero, *non-elitist* strategies generate a *subsequence* of $\boldsymbol{x}$-values converging to the essential global optimum.

The sequence converges in expectation *linearly to* $\mathbf{0}$ *if there exists a* $c > 0$ *such that*

$$-c = \lim_{t\to\infty} \mathbb{E} \log \frac{\|\boldsymbol{x}^{(t+1)}\|}{\|\boldsymbol{x}^{(t)}\|} \quad . \tag{9}$$

The constant c *is the convergence rate of the algorithm.*

Linear convergence hence means that asymptotically in t, the logarithm of the distance to the optimum decreases linearly in t like $-ct$. This behavior has been observed in Figure 2 for the (1+1)-ES with 1/5th success rule on a unimodal spherical function.

Note that λ function evaluations are performed per iteration and it is then often useful to consider a convergence rate *per function evaluation*, i.e. to normalize the convergence rate by λ.

The *progress rate* measures the reduction of the distance to optimum within a single generation [1].

Definition 2 (progress rate)**.** *The normalized progress rate is defined as the expected relative reduction of* $\|\boldsymbol{x}^{(t)}\|$

$$\begin{aligned}\varphi^* &= n\,\mathbb{E}\left(\frac{\|\boldsymbol{x}^{(t)}\| - \|\boldsymbol{x}^{(t+1)}\|}{\|\boldsymbol{x}^{(t)}\|} \,\middle|\, \boldsymbol{x}^{(t)}, s^{(t)}\right) \\ &= n\left(1 - \mathbb{E}\left(\frac{\|\boldsymbol{x}^{(t+1)}\|}{\|\boldsymbol{x}^{(t)}\|} \,\middle|\, \boldsymbol{x}^{(t)}, s^{(t)}\right)\right) \quad , \end{aligned} \tag{10}$$

where the expectation is taken over $\boldsymbol{x}^{(t+1)}$*, given* $(\boldsymbol{x}^{(t)}, s^{(t)})$*. In situations commonly considered in theoretical analyses,* φ^* *does not depend on* $\boldsymbol{x}^{(t)}$ *and is expressed as a function of strategy parameters* $s^{(t)}$*.*

Definitions 1 and 2 are related, in that for a given $\boldsymbol{x}^{(t)}$

$$\varphi^* \le -n \log \mathbb{E} \frac{\|\boldsymbol{x}^{(t+1)}\|}{\|\boldsymbol{x}^{(t)}\|} \tag{11}$$

$$\le -n\,\mathbb{E} \log \frac{\|\boldsymbol{x}^{(t+1)}\|}{\|\boldsymbol{x}^{(t)}\|} = nc \quad . \tag{12}$$

Therefore, progress rate φ^* and convergence rate nc do not agree and we might observe convergence ($c > 0$) while $\varphi^* < 0$. However for $n \to \infty$, we typically have $\varphi^* = nc$ [55].

The normalized progress rate φ^* for evolution strategies has been extensively studied in various situations, see Section 4.2. Scale-invariance and (sometimes artificial) assumptions on the step-size typically ensure that the progress rates do not depend on t.

Another way to describe how fast an algorithm approaches the optimum is to count the number of function evaluations needed to reduce the distance to the optimum by a given factor $1/\epsilon$ or, similarly, the runtime to hit a ball of radius ϵ around the optimum, starting, e.g., from distance one.

Definition 3 (runtime)**.** *The runtime is the first hitting time of a ball around the optimum. Specifically, the runtime in number of function evaluations as a function of* ϵ *reads*

$$\begin{aligned}&\lambda \times \min\left\{t : \|\boldsymbol{x}^{(t)}\| \le \epsilon \times \|\boldsymbol{x}^{(0)}\|\right\} \\ &\quad = \lambda \times \min\left\{t : \frac{\|\boldsymbol{x}^{(t)}\|}{\|\boldsymbol{x}^{(0)}\|} \le \epsilon\right\} \quad . \end{aligned} \tag{13}$$

Linear convergence with rate c as given in Eq. (9) implies that, for $\epsilon \to 0$, the expected runtime divided by $\log(1/\epsilon)$ goes to the constant λ/c.

4.1 Lower Runtime Bounds

Evolution strategies with a fixed number of parent and offspring individuals cannot converge faster than linearly and with a convergence rate of $\mathcal{O}(1/n)$. This means that their runtime is lower bounded by a constant times $\log(1/\epsilon^n) = n \log(1/\epsilon)$ [56, 57, 58, 59, 60]. This result can be obtained by analyzing the branching factor of the tree of possible paths the algorithm can take. It therefore holds for any optimization algorithm taking decisions based solely on a bounded number of comparisons between fitness values [56, 57, 58]. More specifically, the runtime of any $(1 \overset{+}{,} \lambda)$-ES with isotropic mutations cannot be asymptotically faster than $\propto n \log(1/\epsilon)\, \lambda / \log(\lambda)$ [61]. Considering more restrictive classes of algorithms can provide more precise non-asymptotic bounds [59, 60]. Different approaches address in particular the (1+1)- and $(1, \lambda)$-ES and precisely characterize the fastest convergence rate that can be obtained with isotropic

normal distributions on any objective function with any step-size adaptation mechanism [62, 55, 63, 64].

Considering the sphere function, the optimal convergence rate is attained with *distance proportional step-size*, that is, a step-size proportional to the distance of the parental centroid to the optimum, $\sigma = \text{const} \times \|\boldsymbol{x}\| = \sigma^* \|\boldsymbol{x}\|/n$. Optimal step-size and optimal convergence rate according to Eqs. (8) and (9) can be expressed in terms of expectation of some random variables that are easily simulated numerically. The convergence rate of the (1+1)-ES with distance proportional step-size is shown in Figure 4 as a function of the normalized step-size $\sigma^* = n\sigma/\|\boldsymbol{x}\|$. The peak of each curve is the upper bound for the convergence rate that can be achieved on any function with any form of step-size adaptation. As for the general bound, the evolution strategy converges linearly and the convergence rate c decreases to zero like $1/n$ for $n \to \infty$ [55, 65, 64], which is equivalent to linear scaling of the runtime in the dimension. The asymptotic limit for the convergence rate of the (1+1)-ES, as shown in the lowest curve in Figure 4, coincides with the progress rate expression given in the next section.

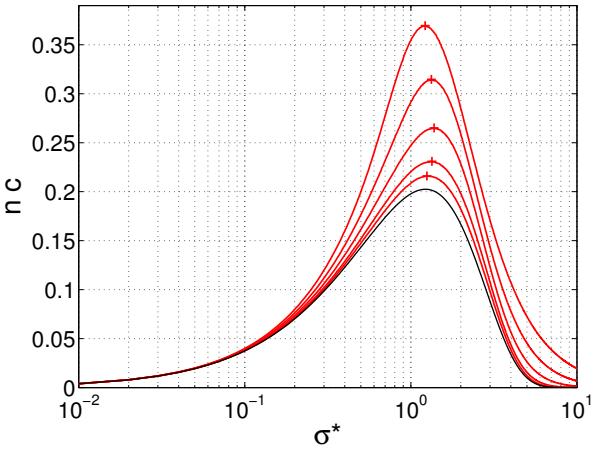

Figure 4: Normalized convergence rate nc versus normalized step-size $n\sigma/\|\boldsymbol{x}\|$ of the (1+1)-ES with distance proportional step-size for $n = 2, 3, 5, 10, 20, \infty$ (top to bottom). The peaks of the graphs represent the upper bound for the convergence rate of the (1+1)-ES with isotropic mutation (corresponding to the lower runtime bound). The limit curve for n to infinity (lower black curve) reveals the optimal normalized progress rate of $\varphi^* \approx 0.202$ of the (1+1)-ES on sphere functions for $n \to \infty$.

4.2 Progress Rates

This section presents analytical approximations to progress rates of evolution strategies for sphere, ridge, and cigar functions in the limit $n \to \infty$. Both one-generation results and those that consider multiple time steps and cumulative step-size adaptation are considered.

The first analytical progress rate results date back to the early work of Rechenberg [1] and Schwefel [3], who considered the sphere and corridor models and very simple strategy variants. Further results have since been derived for various ridge functions, several classes of convex quadratic functions, and more general constrained linear problems. The strategies that results are available for have increased in complexity as well and today include multi-parent strategies employing recombination as well as several step-size adaptation mechanisms. Only strategy variants with isotropic mutation distributions have been considered up to this point. However, parameter control strategies that successfully adapt the shape of the mutation distribution (such as CMA-ES) effectively transform ellipsoidal functions into (almost) spherical ones, thus lending extra relevance to the analysis of sphere and sphere-like functions.

The simplest convex quadratic functions to be optimized are variants of the sphere function (see also the discussion of invariance in Section 3.8)

$$f(\boldsymbol{x}) = \|\boldsymbol{x}\|^2 = \sum_{i=1}^{n} x_i^2 = R^2 \,,$$

where R denotes the distance from the optimal solution. Expressions for the progress rate of evolution strategies on sphere functions can be computed by decomposing mutation vectors into two components $\boldsymbol{z}_\odot$ and $\boldsymbol{z}_\ominus$ as illustrated in Fig. 5. Component $\boldsymbol{z}_\odot$ is the projection of $\boldsymbol{z}$ onto the negative of the gradient vec-

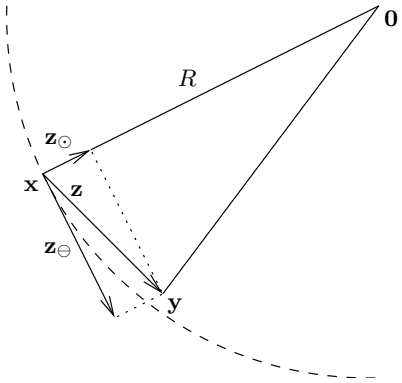

Figure 5: Decomposition of mutation vector $\boldsymbol{z}$ into a component $\boldsymbol{z}_\odot$ in the direction of the negative of the gradient vector of the objective function and a perpendicular component $\boldsymbol{z}_\ominus$.

tor ∇f of the objective function. It contributes positively to the fitness of offspring candidate solution $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{z}$ if and only if $-\nabla f(\boldsymbol{x}) \cdot \boldsymbol{z} > 0$. Component $\boldsymbol{z}_\ominus = \boldsymbol{z} - \boldsymbol{z}_\odot$ is perpendicular to the gradient direction and contributes negatively to the offspring fitness. Its expected squared length exceeds that of $\boldsymbol{z}_\odot$ by a factor of $n-1$. Considering normalized quantities $\sigma^* = \sigma n/R$ and $\varphi^* = \varphi n/R$ allows giving concise mathematical representations of the scaling properties of various evolution strategies on spherical functions as shown below. Constant σ^* corresponds to the distance proportional step-size from Section 4.1.

4.2.1 (1+1)-ES on Sphere Functions

The normalized progress rate of the $(1+1)$-ES on sphere functions is

$$\varphi^* = \frac{\sigma^*}{\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{8}\sigma^{*2}} - \frac{\sigma^{*2}}{4}\left[1 - \mathrm{erf}\left(\frac{\sigma^*}{\sqrt{8}}\right)\right] \quad (14)$$

in the limit of $n \to \infty$ [1]. The expression in square brackets is the success probability (i.e., the probability that the offspring candidate solution is superior to its parent and thus replaces it). The first term in Eq. (14) is the contribution to the normalized progress rate from the component $\boldsymbol{z}_\odot$ of the mutation vector that is parallel to the gradient vector. The second term results from the component $\boldsymbol{z}_\ominus$ that is perpendicular to the gradient direction.

The black curve in Figure 4 illustrates how the normalized progress rate of the (1+1)-ES on sphere functions in the limit $n \to \infty$ depends on the normalized mutation strength. For small normalized mutation strengths, the normalized progress rate is small as the short steps that are made do not yield significant progress. The success probability is nearly one half. For large normalized mutation strengths, progress is near zero as the overwhelming majority of steps result in poor offspring that are rejected. The normalized progress rate assumes a maximum value of $\varphi^* = 0.202$ at normalized mutation strength $\sigma^* = 1.224$. The range of step-sizes for which close to optimal progress is achieved is referred to as the evolution window [1]. In the runs of the (1+1)-ES with constant step-size shown in Fig. 2, the normalized step-size initially is to the left of the evolution window (large relative distance to the optimal solution) and in the end to its right (small relative distance to the optimal solution), achieving maximal progress at a point in between.

4.2.2 $(\mu/\mu, \lambda)$-ES on Sphere Functions

The normalized progress rate of the $(\mu/\mu, \lambda)$-ES on sphere functions is described by

$$\varphi^* = \sigma^* c_{\mu/\mu,\lambda} - \frac{\sigma^{*2}}{2\mu} \quad (15)$$

in the limit $n \to \infty$ [2]. The term $c_{\mu/\mu,\lambda}$ is the expected value of the average of the μ largest order statistics of λ independent standard normally distributed random numbers. For λ fixed, $c_{\mu/\mu,\lambda}$ decreases with increasing μ. For fixed truncation ratio μ/λ, $c_{\mu/\mu,\lambda}$ approaches a finite limit value as λ and μ increase [15, 8].

It is easily seen from Eq. (15) that the normalized progress rate of the $(\mu/\mu, \lambda)$-ES is maximized by normalized mutation strength

$$\sigma^* = \mu c_{\mu/\mu,\lambda} \ . \quad (16)$$

The normalized progress rate achieved with that setting is

$$\varphi^* = \frac{\mu c_{\mu/\mu,\lambda}^2}{2} \quad . \tag{17}$$

The progress rate is negative if $\sigma^* > 2\mu c_{\mu/\mu,\lambda}$. Figure 6 illustrates how the optimal normalized progress rate per offspring depends on the population size parameters μ and λ. Two interesting observations can be made from the figure.

- For all but the smallest values of λ, the $(\mu/\mu, \lambda)$-ES with $\mu > 1$ is capable of significantly more rapid progress per offspring than the $(1, \lambda)$-ES. This contrasts with findings for the $(\mu/1, \lambda)$-ES, the performance of which on sphere functions for $n \to \infty$ monotonically deteriorates with increasing μ [8].
- For large λ, the optimal truncation ratio is $\mu/\lambda = 0.27$, and the corresponding progress per offspring is 0.202. Those values are identical to the optimal success probability and resulting normalized progress rate of the (1+1)-ES. Beyer [8] shows that the correspondence is no coincidence and indeed exact. The step-sizes that the two strategies employ differ widely, however. The optimal step-size of the (1+1)-ES is 1.224; that of the $(\mu/\mu, \lambda)$-ES is $\mu c_{\mu/\mu,\lambda}$ and for fixed truncation ratio μ/λ increases (slightly superlinearly) with the population size. For example, optimal step-sizes of $(\mu/\mu, 4\mu)$-ES for $\mu \in \{1, 2, 3\}$ are 1.029, 2.276, and 3.538, respectively. If offspring candidate solutions can be evaluated in parallel, the $(\mu/\mu, \lambda)$-ES is preferable to the (1+1)-ES, which does not benefit from the availability of parallel computational resources.

Equation (15) holds in the limit $n \to \infty$ for any finite value of λ. In finite but high dimensional search spaces, it can serve as an approximation to the normalized progress rate of the $(\mu/\mu, \lambda)$-ES on sphere functions in the vicinity of the optimal step-size provided that λ is not too large. A better approximation for finite n is derived in [15, 8] (however compare also [55]).

The improved performance of the $(\mu/\mu, \lambda)$-ES for $\mu > 1$ compared to the strategy that uses $\mu = 1$ is

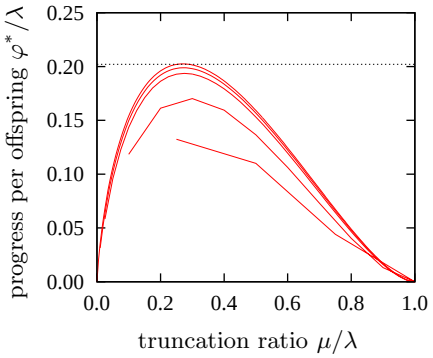

Figure 6: Maximal normalized progress per offspring of the $(\mu/\mu, \lambda)$-ES on sphere functions for $n \to \infty$ plotted against the truncation ratio. The curves correspond to, from bottom to top, $\lambda = 4, 10, 40, 100, \infty$. The dotted line represents the maximal progress rate of the (1+1)-ES.

a consequence of the factor μ in the denominator of the term in Eq. (15) that contributes negatively to the normalized progress rate. The components $\boldsymbol{z}_\odot$ of mutation vectors selected for survival are correlated and likely to point in the direction opposite to the gradient vector. The perpendicular components $\boldsymbol{z}_\ominus$ in the limit $n \to \infty$ have no influence on whether a candidate solution is selected for survival and are thus uncorrelated. The recombinative averaging of mutation vectors results in a length of the $\boldsymbol{z}_\odot$-component similar to those of individual mutation vectors. However, the squared length of the components perpendicular to the gradient direction is reduced by a factor of μ, resulting in the reduction of the negative term in Eq. (15) by a factor of μ. Beyer [15] has coined the term *genetic repair* for this phenomenon.

Weighted recombination (compare Algorithms 5 and 6) can significantly increase the progress rate of $(\mu/\mu, \lambda)$-ES on sphere functions. If n is large, the kth best candidate solution is optimally associated with a weight proportional to the expected value of the kth largest order statistic of a sample of λ independent standard normally distributed random numbers. The resulting optimal normalized progress rate per offspring candidate solution for large values of λ then approaches a value of 0.5, exceeding that of

optimal unweighted recombination by a factor of almost two and a half [13]. The weights are symmetric about zero. If only positive weights are employed and $\mu = \lfloor \lambda/2 \rfloor$, the optimal normalized progress rate per offspring with increasing λ approaches a value of 0.25. The weights in Algorithms 5 and 6 closely resemble those positive weights.

4.2.3 $(\mu/\mu, \lambda)$-ES on Noisy Sphere Functions

Noise in the objective function is most commonly modeled as being Gaussian. If evaluation of a candidate solution $\boldsymbol{x}$ yields a noisy objective function value $f(\boldsymbol{x}) + \sigma_\epsilon \mathcal{N}(0,1)$, then inferior candidate solutions will sometimes be selected for survival and superior ones discarded. As a result, progress rates decrease with increasing noise strength σ_ϵ. Introducing normalized noise strength $\sigma^*_\epsilon = \sigma_\epsilon n/(2R^2)$, in the limit $n \to \infty$, the normalized progress rate of the $(\mu/\mu, \lambda)$-ES on noisy sphere functions is

$$\varphi^* = \frac{\sigma^* c_{\mu/\mu,\lambda}}{\sqrt{1+\vartheta^2}} - \frac{\sigma^{*2}}{2\mu} \tag{18}$$

where $\vartheta = \sigma^*_\epsilon/\sigma^*$ is the noise-to-signal ratio that the strategy operates under [66]. Noise does not impact the term that contributes negatively to the strategy's progress. However, it acts to reduce the magnitude of the positive term stemming from the contributions of mutation vectors parallel to the gradient direction. Notice that unless the noise scales such that σ^*_ϵ is independent of the location in search space (i.e., the standard deviation of the noise term increases in direct proportion to $f(\boldsymbol{x})$, such as in a multiplicative noise model with constant noise strength), Eq. (18) describes progress in single time steps only rather than a rate of convergence.

Figure 7 illustrates for different offspring population sizes λ how the optimal progress rate per offspring depends on the noise strength. The curves have been obtained from Eq. (18) for optimal values of σ^* and μ. As the averaging of mutation vectors results in a vector of reduced length, increasing λ (and μ along with it) allows the strategy to operate using larger and larger step-sizes. Increasing the step-size reduces the noise-to-signal ratio ϑ that the strategy operates under and thereby reduces the impact of noise on selection for survival. Through genetic repair, the $(\mu/\mu, \lambda)$-ES thus implicitly implements the rescaling of mutation vectors proposed in [2] for the $(1, \lambda)$-ES in the presence of noise. (Compare c_m and η_x in Algorithms 5 and 6 that, for values smaller than one, implement the explicit rescaling). It needs to be emphasized though that in finite-dimensional search spaces, the ability to increase λ without violating the assumptions made in the derivation of Eq. (18) is severely limited. Nonetheless, the benefits resulting from genetic repair are significant, and the performance of the $(\mu/\mu, \lambda)$-ES is much more robust in the presence of noise than that of the (1+1)-ES.

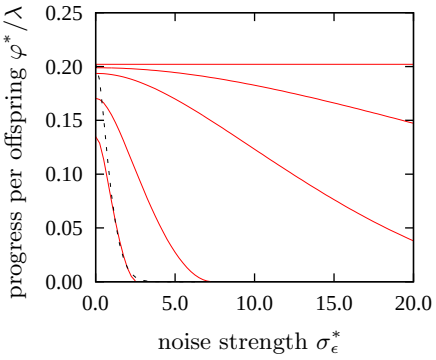

Figure 7: Optimal normalized progress rate per offspring of the $(\mu/\mu, \lambda)$-ES on noisy sphere functions for $n \to \infty$ plotted against the normalized noise strength. The solid lines depict results for, from bottom to top, $\lambda = 4, 10, 40, 100, \infty$ and optimally chosen μ. The dashed line represents the optimal progress rate of the (1+1)-ES [67].

4.2.4 Cumulative Step-Size Adaptation

All progress rate results discussed up to this point consider single time steps of the respective evolution strategies only. Analyses of the behavior of evolution strategies that include some form of step-size adaptation are considerably more difficult. Even for objective functions as simple as sphere functions, the state of the strategy is described by several variables with nonlinear, stochastic dynamics, and simplifying assumptions need to be made in order to arrive at

quantitative results.

In the following we consider the $(\mu/\mu, \lambda)$-ES with cumulative step-size adaptation (Algorithm 4 with Eq. (1) in place of Lines 8 and 8b for mathematical convenience) and parameters set such that $c_\sigma \to 0$ as $n \to \infty$ and $d = \Theta(1)$. The state of the strategy on noisy sphere functions with $\sigma^*_\epsilon = \text{const}$ (i.e., noise that decreases in strength as the optimal solution is approached) is described by the distance R of the parental centroid from the optimal solution, normalized step-size σ^*, the length of the search path s parallel to the direction of the gradient vector of the objective function, and that path's overall squared length. After initialization effects have faded, the distribution of the latter three quantities is time invariant. Mean values of the time invariant distribution can be approximated by computing expected values of the variables after a single iteration of the strategy in the limit $n \to \infty$ and imposing the condition that those be equal to the respective values before that iteration. Solving the resulting system of equations for $\sigma^*_\epsilon \leq \sqrt{2}\mu c_{\mu/\mu,\lambda}$ yields

$$\sigma^* = \mu c_{\mu/\mu,\lambda} \sqrt{2 - \left(\frac{\sigma^*_\epsilon}{\mu c_{\mu/\mu,\lambda}}\right)^2} \tag{19}$$

for the average normalized mutation strength assumed by the strategy [68, 69]. The corresponding normalized progress rate

$$\varphi^* = \frac{\sqrt{2}-1}{2} \mu c^2_{\mu/\mu,\lambda} \left[2 - \left(\frac{\sigma^*_\epsilon}{\mu c_{\mu/\mu,\lambda}}\right)^2\right] \tag{20}$$

is obtained from Eq. (18). Both the average mutation strength and the resulting progress rate are plotted against the noise strength in Fig. 8. For small noise strengths cumulative step-size adaptation generates mutation strengths that are larger than optimal. The evolution window continually shifts toward smaller values of the step-size, and adaptation remains behind its target. However, the resulting mutation strengths achieve progress rates within 20 percent of optimal ones. For large noise strengths the situation is reversed and the mutation strengths generated by cumulative step-size adaptation are smaller than optimal. However, increasing the population size parameters μ and λ allows shifting the operating regime of the strategy toward the left hand side of the graphs in Fig. 8, where step-sizes are near optimal. As above, it is important to keep in mind the limitations of the results derived in the limit $n \to \infty$. In finite dimensional search spaces the ability to compensate for large amounts of noise by increasing the population size is more limited than Eqs. (19) and (20) suggest.

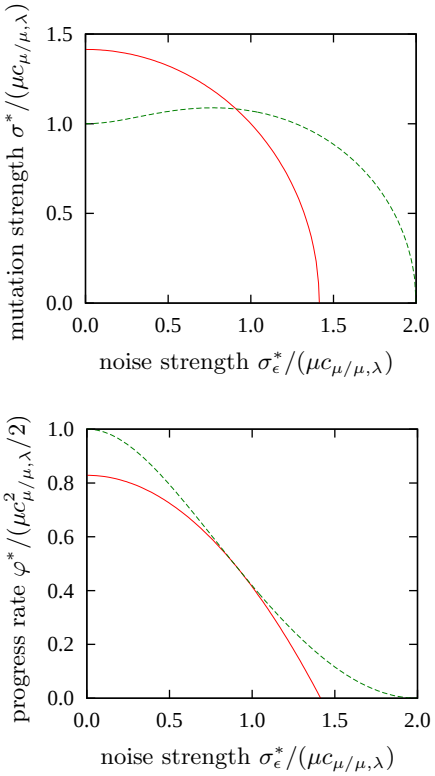

Figure 8: Normalized mutation strength and normalized progress rate of the $(\mu/\mu, \lambda)$-ES with cumulative step size adaptation on noisy sphere functions for $n \to \infty$ plotted against the normalized noise strength. The dashed lines depict optimal values.

4.2.5 Parabolic Ridge Functions

A class of test functions that poses difficulties very different from those encountered in connection with

sphere functions are ridge functions,

$$f(\boldsymbol{x}) = x_1 + \xi \left(\sum_{i=2}^{n} x_i^2 \right)^{\alpha/2} = x_1 + \xi R^\alpha \ ,$$

which include the parabolic ridge for $\alpha = 2$. The x_1-axis is referred to as the ridge axis, and R denotes the distance from that axis. Progress can be made by minimizing the distance from the ridge axis or by proceeding along it. The former requires decreasing step-sizes and is limited in its effect as $R \geq 0$. The latter allows indefinite progress and requires that the step-size does not decrease to zero. Short and long term goals may thus be conflicting, and inappropriate step-size adaptation may lead to stagnation.

As an optimal solution to the ridge problem does not exist, the progress rate φ of the $(\mu/\mu, \lambda)$-ES on ridge functions is defined as the expectation of the step made in the direction of the negative ridge axis. For constant step-size, the distance R of the parental centroid from the ridge axis assumes a time-invariant limit distribution. An approximation to the mean value of that distribution can be obtained by identifying that value of R for which the expected change is zero. Using this value yields

$$\varphi = \frac{2\mu c_{\mu/\mu,\lambda}^2}{n\xi(1 + \sqrt{1 + (2\mu c_{\mu/\mu,\lambda}/(n\xi\sigma))^2})} \tag{21}$$

for the progress rate of the $(\mu/\mu, \lambda)$-ES on parabolic ridge functions [70]. The strictly monotonic behavior of the progress rate, increasing from a value of zero for $\sigma = 0$ to $\varphi = \mu c_{\mu/\mu,\lambda}^2/(n\xi)$ for $\sigma \to \infty$, is fundamentally different from that observed on sphere functions. However, the derivative of the progress rate with regard to the step-size for large values of σ tends to zero. The limited time horizon of any search as well as the intent of using ridge functions as local rather than global models of practically relevant objective functions both suggest that it may be unwise to increase the step-size without bounds.

The performance of cumulative step-size adaptation on parabolic ridge functions can be studied using the same approach as described above for sphere functions, yielding

$$\sigma = \frac{\mu c_{\mu/\mu,\lambda}}{\sqrt{2}n\xi} \tag{22}$$

for the (finite) average mutation strength [71]. From Eq. (21), the corresponding progress rate

$$\varphi = \frac{\mu c_{\mu/\mu,\lambda}^2}{2n\xi} \tag{23}$$

is greater than half of the progress rate attained with any finite step size.

4.2.6 Cigar Functions

While parabolic ridge functions provide an environment for evaluating whether step-size adaptation mechanisms are able to avoid stagnation, the ability to make continual meaningful positive progress with some constant nonzero step-size is of course atypical for practical optimization problems. A class of ridge-like functions that requires continual adaptation of the mutation strength and is thus a more realistic model of problems requiring ridge following are cigar functions

$$f(\boldsymbol{x}) = x_1^2 + \xi \sum_{i=2}^{n} x_i^2 = x_1^2 + \xi R^2$$

with parameter $\xi \geq 1$ being the condition number of the Hessian matrix. Small values of ξ result in sphere-like characteristics, large values in ridge-like ones. As above, R measures the distance from the x_1-axis.

Assuming successful adaptation of the step-size, evolution strategies exhibit linear convergence on cigar functions. The expected relative per iteration change in objective function value of the population centroid is referred to as the quality gain Δ and determines the rate of convergence. In the limit $n \to \infty$ it is described by

$$\Delta^* = \begin{cases} \dfrac{\sigma^{*2}}{2\mu(\xi - 1)} & \text{if } \sigma^* < 2\mu c_{\mu/\mu,\lambda} \dfrac{\xi - 1}{\xi} \\ c_{\mu/\mu,\lambda}\sigma^* - \dfrac{\sigma^{*2}}{2\mu} & \text{otherwise} \end{cases}$$

where $\sigma^* = \sigma n/R$ and $\Delta^* = \Delta n/2$ [72]. That relationship is illustrated in Fig. 9 for several values of the conditioning parameter. The parabola for $\xi = 1$ reflects the simple quadratic relationship for sphere

functions seen in Eq. (15). (For the case of sphere functions, normalized progress rate and normalized quality gain are the same.) For cigar functions with large values of ξ, two separate regimes can be identified. For small step-sizes, the quality gain of the strategy is limited by the size of the steps that can be made in the direction of the x_1-axis. The x_1-component of the population centroid virtually never changes sign. The search process resembles one of ridge following, and we refer to the regime as the ridge regime. In the other regime, the step-size is such that the quality gain of the strategy is effectively limited by the ability to approach the optimal solution in the subspace spanned by the $x_2, \ldots, x_n$-axes. The x_1-component of the population centroid changes sign much more frequently than in the ridge regime, as is the case on sphere functions. We thus refer to the regime as the sphere regime.

The approach to the analysis of the behavior of cumulative step-size adaptation explained above for sphere and parabolic ridge functions can be applied to cigar functions as well, yielding

$$\sigma^* = \sqrt{2}\mu c_{\mu/\mu,\lambda}$$

for the average normalized mutation strength generated by cumulative step-size adaptation [72]. The corresponding normalized quality gain is

$$\Delta^* = \begin{cases} (\sqrt{2}-1)\mu c^2_{\mu/\mu,\lambda} & \text{if } \xi < \dfrac{\sqrt{2}}{\sqrt{2}-1} \\ \dfrac{\mu c^2_{\mu/\mu,\lambda}}{\xi - 1} & \text{otherwise .} \end{cases}$$

Both are compared with optimal values in Fig. 10. For small condition numbers, $(\mu/\mu, \lambda)$-ES operate in the sphere regime and are within 20 percent of the optimal quality gain as seen above. For large condition numbers, the strategy operates in the ridge regime and achieves a quality gain within a factor of two of the optimal one, in accordance with the findings for the parabolic ridge above.

4.2.7 Further Work

Further research regarding the progress rate of evolution strategies in different test environments includes work analyzing the behavior of mutative self-adaptation for linear [22], spherical [73], and ridge functions [74]. Hierarchically organized evolution strategies have been studied when applied to both parabolic ridge and sphere functions [75, 76]. Several step-size adaptation techniques have been compared for ridge functions, including, but not limited to, parabolic ones [77]. A further class of convex quadratic functions for which quality gain results have been derived is characterized by the occurrence of only two distinct eigenvalues of the Hessian, both of which occur with high multiplicity [78, 79].

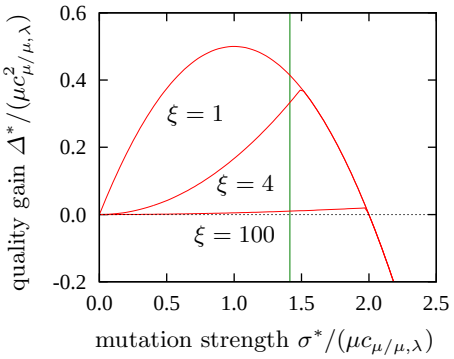

Figure 9: Normalized quality gain of $(\mu/\mu, \lambda)$-ES on cigar functions for $n \to \infty$ plotted against the normalized mutation strength for $\xi \in \{1, 4, 100\}$. The vertical line represents the average normalized mutation strength generated by cumulative step-size adaptation.

An analytical investigation of the behavior of the (1+1)-ES on noisy sphere functions finds that failure to reevaluate the parental candidate solution results in the systematic overvaluation of the parent and thus in potentially long periods of stagnation [67]. Contrary to what might be expected, the increased difficulty of replacing parental candidate solutions can have a positive effect on progress rates as it tends to prevent the selection for survival of offspring candidate solutions solely due to favorable noise values. The convergence behavior of the (1+1)-ES on finite dimensional sphere functions is studied by Jebalia et al. [80] who show that the additive noise model is inappropriate in finite dimensions unless the parental

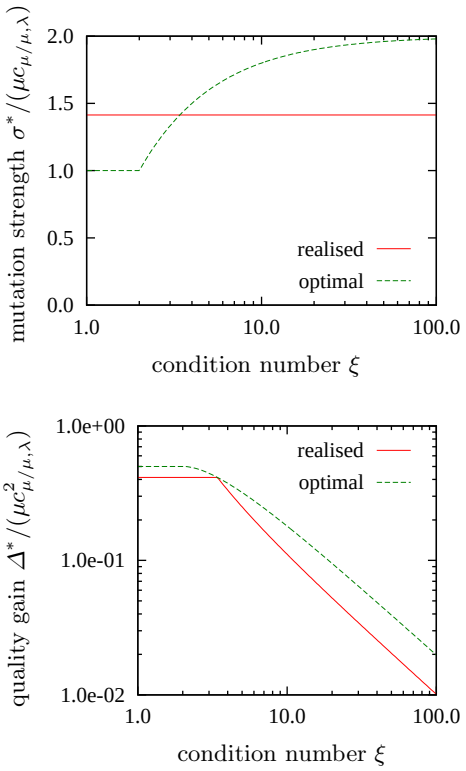

Figure 10: Normalized mutation strength and normalized quality gain of the $(\mu/\mu, \lambda)$-ES with cumulative step-size adaptation on cigar functions for $n \to \infty$ plotted against the condition number of the cigar. The dashed curves represent optimal values.

candidate solution is reevaluated, and who suggest a multiplicative noise model instead. An analysis of the behavior of (μ, λ)-ES (without recombination) for noisy sphere functions finds that in contrast to the situation in the absence of noise, strategies with $\mu > 1$ can outperform $(1, \lambda)$-ES if there is noise present [81]. The use of non-singleton populations increases the signal-to-noise ratio and thus allows for more effective selection of good candidate solutions. The effects of non-Gaussian forms of noise on the performance of $(\mu/\mu, \lambda)$-ES applied to the optimization of sphere functions have also been investigated [82].

Finally, there are some results regarding the optimization of time-varying objectives [83] as well as analyses of simple constraint handling techniques [84, 85, 86].

4.3 Convergence Proofs

In the previous section we have described theoretical results that involve approximations in their derivation and consider the limit for $n \to \infty$. In this section, exact results are discussed.

Convergence proofs with only mild assumptions on the objective function are easy to obtain for evolution strategies with a step-size that is effectively bounded from below and above (and, for non-elitist strategies, when additionally the search space is bounded) [63, 12]. In this case, the expected runtime to reach an ϵ-ball around the global optimum (see also Definition 3) cannot be faster than $\propto 1/\epsilon^n$, as obtained with pure random search for $\epsilon \to 0$ or $n \to \infty$.[12] Similarly, convergence proofs can be obtained for adaptive strategies that include provisions for using a fixed step-size and covariance matrix with some constant probability.

Convergence proofs for strategy variants that do not explicitly ensure that long steps are sampled for a sufficiently long time typically require much stronger restrictions on the set of objective functions that they hold for. Such proofs however have the potential to reveal much faster, namely linear convergence. Evolution strategies with the *artificial* distance proportional step-size, $\sigma = \text{const} \times \|\boldsymbol{x}\|$, exhibit, as shown above, linear convergence on the sphere function with an associated runtime proportional to $\log(1/\epsilon)$ [88, 62, 80, 64]. This result can be easily proved by using a law of large numbers, because $\|\boldsymbol{x}^{(t+1)}\|/\|\boldsymbol{x}^{(t)}\|$ are independent and identically distributed for all t.

Without the artificial choice of step-size, $\sigma/\|\boldsymbol{x}\|$ becomes a random variable. If this random variable is a homogeneous Markov chain and stable enough to satisfy the law of large numbers, linear convergence is maintained [88, 63]. The stability of the Markov chain associated to the self-adaptive $(1, \lambda)$-ES on the

[12] If the mutation distribution is not normal and exhibits a singularity in zero, convergence can be much faster than with random search even when the step-size is bounded away from zero [87].

sphere function has been shown in dimension $n = 1$ [89] providing thus a proof of linear convergence of this algorithm. The extension of this proof to higher dimensions is straightforward.

Proofs that are formalized by upper bounds on the time to reduce the distance to the optimum by a given factor can also associate the linear dependency of the convergence rate in the dimension n. The $(1 + \lambda)$- and the $(1, \lambda)$-ES with common variants of the 1/5th success rule converge linearly on the sphere function with a runtime of $\mathcal{O}(n \log(1/\epsilon)\, \lambda/\sqrt{\log \lambda})$ [90, 61]. When λ is smaller than $\mathcal{O}(n)$ the $(1 + \lambda)$-ES with a modified success rule is even $\sqrt{\log \lambda}$ times faster and therefore matches the general lower runtime bound $\Omega(n \log(1/\epsilon)\, \lambda/\log(\lambda))$ [61, Theorem 5]. On convex-quadratic functions, the asymptotic runtime of the (1+1)-ES is the same as on the sphere function and, at least in some cases, proportional to the condition number of the problem [91].

Convergence proofs of modern evolution strategies with recombination, of CSA-ES, CMA-ES or xNES are not yet available, however we believe that some of them are likely to be achieved in the coming decade.

References

[1] I. Rechenberg: *Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution* (Frommann-Holzboog Verlag, 1973)

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

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

[4] H.-P. Schwefel: *Evolution and Optimum Seeking* (Wiley, 1995)

[5] L. J. Fogel, A. J. Owens, M. J. Walsh: *Artificial Intelligence through Simulated Evolution* (Wiley, 1966)

[6] H.-G. Beyer, H.-P. Schwefel: Evolution strategies — A comprehensive introduction, Natural computing **1**(1), 3 – 52 (2002)

[7] D. B. Fogel: *Evolutionary Computation: The Fossil Record* (Wiley – IEEE Press, 1998)

[8] H.-G. Beyer: *The Theory of Evolution Strategies* (Springer Verlag, 2001)

[9] D. E. Goldberg: *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison Wesley, 1989)

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

[11] H.-P. Schwefel, G. Rudolph: Contemporary evolution strategies, Advances in Artificial Life, ed. by F. Morán et al. (Springer Verlag 1995) 891 – 907

[12] G. Rudolph: *Convergence Properties of Evolutionary Algorithms* (Verlag Dr. Kovač, 1997)

[13] D. V. Arnold: Weighted multirecombination evolution strategies, Theoretical Computer Science **361**(1), 18 – 37 (2006)

[14] H. Mühlenbein, D. Schlierkamp-Voosen: Predictive models for the breeder genetic algorithm I. Continuous parameter optimization, Evolutionary Computation **1**(1), 25 – 49 (1993)

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

[16] C. Kappler: Are evolutionary algorithms improved by large mutations?, Parallel Problem Solving from Nature (PPSN IV), ed. by H.-M. Voigt et al. (Springer Verlag 1996) 346 – 355

[17] G. Rudolph: Local convergence rates of simple evolutionary algorithms with Cauchy mutations, IEEE Transactions on Evolutionary Computation **1**(4), 249 – 258 (1997)

[18] X. Yao, Y. Liu, G. Lin: Evolutionary programming made faster, IEEE Transactions on Evolutionary Computation **3**(2), 82 – 102 (1999)

[19] M. Herdy: The number of offspring as strategy parameter in hierarchically organized evolution strategies, ACM SIGBIO Newsletter **13**(2), 2 – 9 (1993)

[20] M. Schumer, K. Steiglitz: Adaptive step size random search, IEEE Transactions on Automatic Control **13**(3), 270 – 276 (1968)

[21] S. Kern, S. D. Müller, N. Hansen, D. Büche, J. Ocenasek, P. Koumoutsakos: Learning probability distributions in continuous evolutionary algorithms — A comparative review, Natural Computing **3**(1), 77 – 112 (2004)

[22] N. Hansen: An analysis of mutative σ-self-adaptation on linear fitness functions, Evolutionary Computation **14**(3), 255 – 275 (2006)

[23] A. Ostermeier, A. Gawelczyk, N. Hansen: A derandomized approach to self-adaptation of evolution strategies, Evolutionary Computation **2**(4), 369 – 380 (1994)

[24] T. Runarsson: Reducing random fluctuations in mutative self-adaptation, Parallel Problem Solving from Nature (PPSN VII), ed. by J. J. Merelo Guervós et al. (Springer Verlag 2002) 194 – 203

[25] A. Ostermeier, A. Gawelczyk, N. Hansen: Step-size adaptation based on non-local use of selection information, Parallel Problem Solving from Nature (PPSN III), ed. by Y. Davidor et al. (Springer Verlag 1994) 189 – 198

[26] N. Hansen, A. Ostermeier, A. Gawelczyk: On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation, International Conference on Genetic Algorithms (ICGA '95), ed. by L. J. Eshelman (Morgan Kaufmann 1995) 57 – 64

[27] N. Hansen, A. Ostermeier: Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation, International Conference on Evolutionary Computation (ICEC '96) (IEEE Press 1996) 312 – 317

[28] A. Ostermeier, N. Hansen: An evolution strategy with coordinate system invariant adaptation of arbitrary normal mutation distributions within the concept of mutative strategy parameter control, Genetic and Evolutionary Computation Conference (GECCO 1999), ed. by W. Banzhaf et al. (Morgan Kaufmann 1999) 902 – 909

[29] D. Wierstra, T. Schaul, J. Peters, J. Schmidhuber: Natural evolution strategies, IEEE Congress on Evolutionary Computation (CEC 2008) (IEEE Press 2008) 3381 – 3387

[30] Y. Sun, D. Wierstra, T. Schaul, J. Schmidhuber: Efficient natural evolution strategies, Genetic and Evolutionary Computation Conference (GECCO 2009) (ACM Press 2009) 539 – 546

[31] T. Glasmachers, T. Schaul, Y. Sun, D. Wierstra, J. Schmidhuber: Exponential natural evolution strategies, Genetic and Evolutionary Computation Conference (GECCO 2010) (ACM Press 2010) 393 – 400

[32] N. Hansen: Invariance, self-adaptation and correlated mutations and evolution strategies, Parallel Problem Solving from Nature (PPSN VI), ed. by M. Schoenauer et al. (Springer Verlag 2000) 355 – 364

[33] A. Ostermeier: An evolution strategy with momentum adaptation of the random number distribution, Parallel Problem Solving from Nature (PPSN II) 1992, ed. by R. Männer, B. Manderick (Elsevier 1992) 199 – 208

[34] J. Poland, A. Zell: Main vector adaptation: A CMA variant with linear time and space complexity, Genetic and Evolutionary Computation Conference (GECCO 2001), ed. by L. Spector et al. (Morgan Kaufmann 2001) 1050 – 1055

[35] J. N. Knight, M. Lunacek: Reducing the space-time complexity of the CMA-ES, Genetic and Evolutionary Computation Conference (GECCO 2007) (ACM Press 2007) 658 – 665

[36] N. Hansen, S. Kern: Evaluating the CMA evolution strategy on multimodal test functions, Parallel Problem Solving from Nature (PPSN VIII), ed. by X. Yao et al. (Springer Verlag 2004) 282 – 291

[37] G. A. Jastrebski, D. V. Arnold: Improving evolution strategies through active covariance matrix adaptation, IEEE Congress on Evolutionary Computation (CEC 2006) (IEEE Press 2006) 2814 – 2821

[38] H.-G. Beyer: Mutate large, but inherit small! On the analysis of rescaled mutations in $(\tilde{1}, \tilde{\lambda})$-ES with noisy fitness data, Parallel Problem Solving from Nature (PPSN V), ed. by A. E. Eiben et al. (Springer Verlag 1998) 109 – 118

[39] S. I. Amari: Natural gradient works efficiently in learning, Neural Computation **10**(2), 251 – 276 (1998)

[40] Y. Sun, D. Wierstra, T. Schaul, J. Schmidhuber: Stochastic search using the natural gradient, International Conference on Machine Learning (ICML '09), ed. by A. P. Danyluk et al. (ACM Press 2009) 1161 – 1168

[41] Y. Akimoto, Y. Nagata, I. Ono, S. Kobayashi: Bidirectional relation between CMA evolution strategies and natural evolution strategies, Parallel Problem Solving from Nature (PPSN XI), ed. by R. Schaefer et al. (Springer Verlag 2010) 154 – 163

[42] L. Arnold, A. Auger, N. Hansen, Y. Ollivier: Information-geometric optimization algorithms: A unifying picture via invariance principles, ArXiv e-prints (2011)

[43] M. Pelikan, M. W. Hausschild, F. G. Lobo: Introduction to estimation of distribution algorithms. In: *Handbook of Computational Intelligence*, ed. by J. Kacprzyk, W. Pedrycz (Springer Verlag 2013)

[44] G. R. Harik, F. G. Lobo: A parameter-less genetic algorithm, Genetic and Evolutionary Computation Conference (GECCO 1999), ed. by W. Banzhaf et al. (Morgan Kaufmann 1999) 258 – 265

[45] F. G. Lobo, D. E. Goldberg: The parameter-less genetic algorithm in practice, Information Sciences **167**(1), 217 – 232 (2004)

[46] A. Auger, N. Hansen: A restart CMA evolution strategy with increasing population size, IEEE Congress on Evolutionary Computation (CEC 2005) (IEEE Press 2005) 1769 – 1776

[47] T. Suttorp, N. Hansen, C. Igel: Efficient covariance matrix update for variable metric evolution strategies, Machine Learning **75**(2), 167 – 197 (2009)

[48] Z. Michalewicz, M. Schoenauer: Evolutionary algorithms for constrained parameter optimizationproblems, Evolutionary Computation **4**(1), 1 – 32 (1996)

[49] E. Mezura-Montes, C. A. Coello Coello: Constraint-handling in nature-inspired numerical optimization: Past, present, and future, Swarm and Evolutionary Computation **1**(4), 173 – 194 (2011)

[50] M. Emmerich, A. Giotis, M. Özdemir, T. Bäck, K. Giannakoglou: Metamodel-assisted evolution strategies, Parallel Problem Solving from Nature (PPSN VII), ed. by J. J. Merelo Guervós et al. (Springer Verlag 2002) 361 – 370

[51] C. Igel, N. Hansen, S. Roth: Covariance matrix adaptation for multi-objective optimization, Evolutionary Computation **15**(1), 1 – 28 (2007)

[52] N. Hansen T. Voß, C. Igel: Improved step size adaptation for the MO-CMA-ES, Genetic and Evolutionary Computation Conference (GECCO 2010) (ACM Press 2010) 487 – 494

[53] R. Salomon: Evolutionary algorithms and gradient search: Similarities and differences, IEEE Transactions on Evolutionary Computation **2**(2), 45 – 55 (1998)

[54] G. Rudolph: Self-adaptive mutations may lead to premature convergence, IEEE Transactions on Evolutionary Computation **5**(4), 410 – 414 (2001)

[55] A. Auger, N. Hansen: Reconsidering the progress rate theory for evolution strategies in finite dimensions, Genetic and Evolutionary Computation Conference (GECCO 2006) (ACM Press 2006) 445 – 452

[56] O. Teytaud, S. Gelly: General lower bounds for evolutionary algorithms, Parallel Problem Solving from Nature (PPSN IX), ed. by T. P. Runarsson et al. (Springer Verlag 2006) 21 – 31

[57] H. Fournier, O. Teytaud: Lower bounds for comparison based evolution strategies using VC-dimension and sign patterns, Algorithmica **59**(3), 387 – 408 (2011)

[58] O. Teytaud: Lower bounds for evolution strategies. In: *Theory of Randomized Search Heuristics: Foundations and Recent Developments*, ed. by A. Auger, B. Doerr (World Scientific Publishing 2011) Chap. 11, pp. 327 – 354

[59] J. Jägersküpper: Lower bounds for hit-and-run direct search, International Symposium on Stochastic Algorithms: Foundations and Applications (SAGA 2007), ed. by J. Hromkovic et al. (Springer Verlag 2007) 118 – 129

[60] J. Jägersküpper: Lower bounds for randomized direct search with isotropic sampling, Operations Research Letters **36**(3), 327 – 332 (2008)

[61] J. Jägersküpper: Probabilistic runtime analysis of $(1 \overset{+}{,} \lambda)$ evolution strategies using isotropic mutations, Genetic and Evolutionary Computation Conference (GECCO 2006) (ACM Press 2006) 461 – 468

[62] M. Jebalia, A. Auger, P. Liardet: Log-linear convergence and optimal bounds for the (1+1)-ES, Evolution Artificielle (EA '07), ed. by N. Monmarché et al. (Springer Verlag 2008) 207 – 218

[63] A. Auger, N. Hansen: Theory of evolution strategies: A new perspective. In: *Theory of Randomized Search Heuristics: Foundations and Recent Developments*, ed. by A. Auger, B. Doerr (World Scientific Publishing 2011) Chap. 10, pp. 289 – 325

[64] A. Auger, D. Brockhoff, N. Hansen: Analyzing the impact of mirrored sampling and sequential selection in elitist evolution strategies, Foundations of Genetic Algorithms (FOGA 11) (ACM Press 2011) 127 – 138

[65] A. Auger, D. Brockhoff, N. Hansen: Mirrored sampling in evolution strategies with weighted recombination, Genetic and Evolutionary Computation Conference (GECCO 2011) (ACM Press 2011) 861 – 868

[66] D. V. Arnold, H.-G. Beyer: Local performance of the $(\mu/\mu_I, \lambda)$-ES in a noisy environment, Foundations of Genetic Algorithms (FOGA 6), ed. by W. N. Martin, W. M. Spears (Morgan Kaufmann 2001) 127 – 141

[67] D. V. Arnold, H.-G. Beyer: Local performance of the (1 + 1)-ES in a noisy environment, IEEE Transactions on Evolutionary Computation **6**(1), 30 – 41 (2002)

[68] D. V. Arnold: *Noisy Optimization with Evolution Strategies* (Kluwer Academic Publishers, 2002)

[69] D. V. Arnold, H.-G. Beyer: Performance analysis of evolutionary optimization with cumulative step length adaptation, IEEE Transactions on Automatic Control **49**(4), 617 – 622 (2004)

[70] A. I. Oyman, H.-G. Beyer: Analysis of the $(\mu/\mu, \lambda)$-ES on the parabolic ridge, Evolutionary Computation **8**(3), 267 – 289 (2000)

[71] D. V. Arnold, H.-G. Beyer: Evolution strategies with cumulative step length adaptation on the noisy parabolic ridge, Natural Computing **7**(4), 555 – 587 (2008)

[72] D. V. Arnold, H.-G. Beyer: On the behaviour of evolution strategies optimising cigar functions, Evolutionary Computation **18**(4), 661 – 682 (2010)

[73] H.-G. Beyer: Towards a theory of evolution strategies: Self-adaptation, Evolutionary Computation **3**(3) (1995)

[74] S. Meyer-Nieberg, H.-G. Beyer: Mutative self-adaptation on the sharp and parabolic ridge, Foundations of Genetic Algorithms (FOGA 9), ed. by C. R. Stephens et al. (Springer Verlag 2007) 70 – 96

[75] D. V. Arnold, A. MacLeod: Hierarchically organised evolution strategies on the parabolic ridge, Genetic and Evolutionary Computation Conference (GECCO 2006) (ACM Press 2006) 437 – 444

[76] H.-G. Beyer, M. Dobler, C. Hämmerle, P. Masser: On strategy parameter control by meta-ES, Genetic and Evolutionary Computation Conference (GECCO 2009) (ACM Press 2009) 499 – 506

[77] D. V. Arnold, A. MacLeod: Step length adaptation on ridge functions, Evolutionary Computation **16**(2), 151 – 184 (2008)

[78] D. V. Arnold: On the use of evolution strategies for optimising certain positive definite quadratic forms, Genetic and Evolutionary Computation Conference (GECCO 2007) (ACM Press 2007) 634 – 641

[79] H.-G. Beyer, S. Finck: Performance of the $(\mu/\mu_I, \lambda)$-σSA-ES on a class of PDQFs, IEEE Transactions on Evolutionary Computation **14**(3), 400 – 418 (2010)

[80] M. Jebalia, A. Auger, N. Hansen: Log-linear convergence and divergence of the scale-invariant (1+1)-ES in noisy environments, Algorithmica **59**(3), 425 – 460 (2011)

[81] D. V. Arnold, H.-G. Beyer: On the benefits of populations for noisy optimization, Evolutionary Computation **11**(2), 111 – 127 (2003)

[82] D. V. Arnold, H.-G. Beyer: A general noise model and its effects on evolution strategy performance, IEEE Transactions on Evolutionary Computation **10**(4), 380 – 391 (2006)

[83] D. V. Arnold, H.-G. Beyer: Optimum tracking with evolution strategies, Evolutionary Computation **14**(3), 291 – 308 (2006)

[84] D. V. Arnold, D. Brauer: On the behaviour of the (1 + 1)-ES for a simple constrained problem, Parallel Problem Solving from Nature (PPSN X), ed. by G. Rudolph et al. (Springer Verlag 2008) 1 – 10

[85] D. V. Arnold: On the behaviour of the $(1, \lambda)$-ES for a simple constrained problem, Foundations of Genetic Algorithms (FOGA 11) (ACM Press 2011) 15 – 24

[86] D. V. Arnold: Analysis of a repair mechanism for the $(1, \lambda)$-ES applied to a simple constrained problem, Genetic and Evolutionary Computation Conference (GECCO 2011) (ACM Press 2011) 853 – 860

[87] Anatoly A. Zhigljavsky: *Theory of Global Random Search* (Kluwer Academic Publishers, 1991)

[88] A. Bienvenüe, O. François: Global convergence for evolution strategies in spherical problems: Some simple proofs and difficulties, Theoretical Computer Science **306**(1-3), 269 – 289 (2003)

[89] A. Auger: Convergence results for (1,λ)-SA-ES using the theory of φ-irreducible Markov chains, Theoretical Computer Science **334**(1-3), 35 – 69 (2005)

[90] J. Jägersküpper: Algorithmic analysis of a basic evolutionary algorithm for continuous optimization, Theoretical Computer Science **379**(3), 329 – 347 (2007)

[91] J. Jägersküpper: How the (1+1) ES using isotropic mutations minimizes positive definite quadratic forms, Theoretical Computer Science **361**(1), 38 – 56 (2006)