Chapter 2
Evolution Strategies

Prior to introducing the particular algorithms in Sect. 2.2, the more general foundations of evolution strategies are introduced in Sect. 2.1. To start with, the definition of an optimization task as used throughout this book is given in Sect. 2.1.1. Following [58], Sect. 2.1.2 presents a discussion of evolution strategy metaheuristics as a special case of evolutionary algorithms. In particular, the components of such a metaheuristic—namely recombination, mutation, evaluation and selection—are described in a general way. Due to the particular importance[1] of the mutation operator for evolution strategies (in $\mathbb{R}^n$), it is discussed in quite some detail in Sect. 2.1.3.

2.1 Introduction

2.1.1 Optimization

Evolution strategies are particularly well suited (and developed) for nonlinear optimization tasks, which are defined as follows (see e.g. [17], Sect. 18.2.1.1):

$$f(\mathbf{x}) = \min! \text{ for } \mathbf{x} \in \mathbb{R}^n \text{ where} \tag{2.1}$$

$$g_i(\mathbf{x}) \leq 0, i \in I = \{1, \ldots, m\}, h_j(\mathbf{x}) = 0, j \in J = \{1, \ldots, r\}, \tag{2.2}$$

[1] This statement, however, is not meant to support the myth mentioned explicitly by Rudolph [58]: "Since early theoretical publications mainly analyzed simple ES without recombination, somehow the myth arose that ES put more emphasis on mutation than on recombination: This is a fatal misconception! Recombination has been an important ingredient of ES from the early beginning and this is still valid today."

T. Bäck et al., *Contemporary Evolution Strategies*, Natural Computing Series,
DOI 10.1007/978-3-642-40137-4_2, © Springer-Verlag Berlin Heidelberg 2013

and the set

$$M = \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \leq 0, \forall i \in I, h_j(\mathbf{x}) = 0, \forall j \in J\} \tag{2.3}$$

is called the set of feasible points and it defines the search space of the optimization problem. A point $\mathbf{x}^* \in \mathbb{R}^n$ is called a global minimum, if

$$f^* = f(\mathbf{x}^*) \leq f(\mathbf{x}) \text{ for all } \mathbf{x} \in M \tag{2.4}$$

Conversely, it is called a local minimum if the above inequality only holds for $\mathbf{x}$ within an ϵ-environment $U_{\epsilon(x)} \subseteq M$.

Formulating an optimization problem as a minimization task is equivalent to searching for a maximum or for a given target value, since maximization of f can be replaced by minimization of $-f$ and a target value $\bar{f}$ can be attained by minimizing $\rho(\bar{f}, f)$ with an arbitrary distance measure[2] ρ.

In this definition of an optimization task it is completely sufficient if the codomain is completely ordered, so that the inequality in Eq. 2.4 can be applied. Throughout this book, we will always deal with the codomain $\mathbb{R}$ only. Moreover, we will not explicitly deal with the handling of constraints (e.g., as defined by Eq. 2.2), and refer the interested reader to Sect. 2.3 where literature references point to state-of-the-art techniques in constraint handling. A special case of constraints are so-called box constraints, as defined below:

$$\begin{aligned} g_1(\mathbf{x}) &= \mathbf{l} - \mathbf{x} \leq \mathbf{0} \text{ where } \mathbf{l} = (l_1, \ldots, l_n)^T \in \mathbb{R}^n \\ g_2(\mathbf{x}) &= \mathbf{x} - \mathbf{u} \leq \mathbf{0} \text{ where } \mathbf{u} = (u_1, \ldots, u_n)^T \in \mathbb{R}^n \end{aligned} \tag{2.5}$$

Vectors $\mathbf{l}$ and $\mathbf{u}$ are called lower and upper bounds, respectively. Box constraints restrict the search space to the hyperrectangle $[l_1, u_1] \times \ldots \times [l_n, u_n]$ and are taken into account for the implementation of algorithms described in this book.

In the field of evolutionary algorithms, the vector $\mathbf{x}$ is often called the decision vector (and its parameters decision parameters), and its objective function value $f(\mathbf{x})$ is also called the fitness value.

2.1.2 Evolution Strategies as a Specialization of Evolutionary Algorithms

Following [8] and [58], evolution strategies are described here as a specialization of evolutionary algorithms. The general framework of an evolutionary algorithm is presented in Algorithm 2.1. During initialization, the first generation, consisting of

[2] See Sect. 12.2.1 in [17] for the definition of a distance measure.

Algorithm 2.1 General outline of an evolutionary algorithm

```
Initialization
repeat
    Recombination
    Mutation
    Evaluation
    Selection
until Termination criterion fulfilled
```

one or more individuals, is created, and the fitness of its individuals is evaluated. After initialization, the so-called evolution loop is entered, which consists of the operators recombination, mutation, evaluation and selection. Recombination creates new individuals, also called offspring, from the parent population. Two major types of recombination, dominant and intermediate recombination, are typically distinguished: In dominant recombination, a property of a parent individual is inherited by the offspring, i.e., this property dominates the corresponding property of the other individuals. For intermediate recombination, the properties of all individuals are taken into account, such that, e.g., in the simplest case, their mean value is used.

The mutation operator provides the main source of variation of offspring in an evolution strategy. Based on sampling random variables, properties of individuals are modified. The newly created individuals are then evaluated, i.e., their fitness values are calculated. Based on these fitness values, selection identifies a subset of individuals which form the new population which is used in the next iteration of the evolution loop. The loop is terminated based on a termination criterion set by the user, such as reaching a maximum number of evaluations, reaching a target fitness value, or stagnation of the search process.

According to [58], evolution strategies as a specific instantiation of evolutionary algorithms are characterized by the following four properties:

- Selection of individuals for recombination is unbiased.
- Selection is a deterministic process.
- Mutation operators are parameterized and therefore they can change their properties during optimization.
- Individuals consist of decision parameters as well as strategy parameters.[3]

The generic framework of an evolutionary algorithm then specializes into a $(\mu/\rho, \kappa, \lambda)$-ES,[4] as described in detail in Algorithm 2.2. Recombination and mutation are summarized here under the term variation. In addition to the description

[3] In the case of the $(1+1)$-ES the strategy parameters may be assigned to the algorithm itself instead of the individual, because only one set of strategy parameters is needed. This also holds for any strategy parameters which are not needed on the individual level (for example the covariance matrix of the CMA-ES).

[4] Algorithm 3 in [58].

Algorithm 2.2 $(\mu/\rho, \kappa, \lambda)$-ES

```
Initialization of $P^{(0)}$ with $\mu$ individuals
$\forall p \in P^{(0)} : p.\Psi.Age \leftarrow 1, p.f \leftarrow f(p.\mathbf{x})$
$t \leftarrow 0$
repeat
    $Q^{(t)} \leftarrow \emptyset$
    for $i = 1 \to \lambda$ do
        Sample $\rho$ parents $p_1, \ldots, p_\rho \in P^{(t)}$ uniformly at random
        $q \leftarrow$ Variation$(p_1, \ldots, p_\rho, \Psi_V)$
        $q.\Psi.Age \leftarrow 0, q.f \leftarrow f(q.\mathbf{x})$
        $Q^{(t)} \leftarrow Q^{(t)} \cup \{q\}$
    end for
    $P^{(t+1)} \leftarrow$ Selection of the $\mu$ best individuals from $Q^{(t)} \cup \{p \in P^{(t)} : p.\Psi.Age < \kappa\}$
    Update $\Psi_V$
    $\forall p \in P^{(t+1)} : p.\Psi.Age \leftarrow p.\Psi.Age + 1$
    $t \leftarrow t + 1$
until Termination criterion fulfilled
```

given in [58] (Algorithm 3), the variation operator of a $(\mu/\rho, \kappa, \lambda)$-ES is defined here by means of a parameter set Ψ_V, and the evaluation operator is explicitly mentioned. A population at generation $t \geq 0$ is denoted $P^{(t)}$ and is a set of individuals. An individual $p \in P^{(t)}$ is a tuple $(\mathbf{x}, \Psi)$ for $\mathbf{x} \in M \subseteq \mathbb{R}^n$, with M as in Eq. 2.3. The sets Ψ and Ψ_V are arbitrary finite sets representing the strategy parameters. Since these parameters are modified internally during execution of the algorithm, they are called endogenous strategy parameters. The number of parent individuals is denoted as μ, the number of offspring individuals as λ, and ρ denotes the number of parents taken into account for generating a single offspring by means of recombination. For these parameters, $\mu, \rho, \lambda \in \mathbb{N}$ and $\rho \leq \mu$ holds.

$\kappa \in \mathbb{N} \cup \{\infty\}$ represents the largest age which can be reached by any individual in the population. In contrast to endogenous parameters, μ, ρ, λ und κ are to be set by the user of the algorithm, such that they are called exogenous strategy parameters.

The setting of κ has a direct impact on the selection operator. Usually, either $\kappa = 1$ (one generation maximum lifetime) or $\kappa = \infty$ (infinite maximum lifetime) is used. The former case is also called comma-selection, the latter plus-selection. Using the standard notation of evolution strategies, this is expressed as $(\mu/\rho, \lambda)$-ES and $(\mu/\rho + \lambda)$-ES, so that κ is not explicitly stated any more. Using $\kappa < \infty$ requires the condition $\lambda \geq \mu$ to hold.

2.1.3 Mutation in $\mathbb{R}^n$

2.1.3.1 The Multivariate Normal Distribution

In [58], three guiding principles for the design of mutation operators are introduced, namely:

- Any point of the search space needs to be attainable with probability strictly larger than zero by means of a finite number of applications of mutation.
- Mutation should be *unbiased*, which can be achieved by using a *maximum entropy distribution*.[5]
- The operator is parameterized, such that the extent of variation can be controlled.

In $\mathbb{R}^n$, these requirements are fulfilled by a multivariate normal distribution. An n-dimensional random vector $\mathbf{X}$ is multivariate normally distributed with expectation $\bar{\mathbf{x}} \in \mathbb{R}^n$ and positive definite[6] covariance matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ if its probability density function is defined according to:

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} (\det \mathbf{C})^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T \mathbf{C}^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right) \tag{2.6}$$

(see p. 86 in [28]). In short notation, this is typically written as $\mathbf{X} \sim N(\bar{\mathbf{x}}, \mathbf{C})$, where $N(\bar{\mathbf{x}}, \mathbf{C})$ denotes the multivariate normal distribution in its general form. In mathematical equations, $N(\bar{\mathbf{x}}, \mathbf{C})$ is sometimes used like a vector, meaning a vector which is actually sampled according to the distribution given. In other words, instead of writing $\mathbf{x}' = \mathbf{x} + \mathbf{X}$ where $\mathbf{X} \sim N(\mathbf{0}, \mathbf{C})$, it is also possible to simply write $\mathbf{x}' = \mathbf{x} + N(\mathbf{0}, \mathbf{C})$.

Due to the positive definiteness of the covariance matrix $\mathbf{C}$, the following eigendecomposition exists (Theorem 1a in [58]):

$$\mathbf{C} = \mathbf{B}\mathbf{D}^2\mathbf{B}^T \tag{2.7}$$

Here, $\mathbf{B}$ denotes an orthogonal matrix,[7] the columns of which are the eigenvectors of $\mathbf{C}$. In [29], $N(\bar{\mathbf{x}}, \mathbf{C})$ is reduced to the distribution $N(\mathbf{0}, \mathbf{I})$ by means of the eigendecomposition given in Eq. 2.7, according to:

$$N(\bar{\mathbf{x}}, \mathbf{C}) \sim \bar{\mathbf{x}} + \mathbf{B}\mathbf{D}N(\mathbf{0}, \mathbf{I}) \tag{2.8}$$

In the field of evolution strategies, the three special cases $N(\mathbf{0}, \mathbf{I})$, $N(\mathbf{0}, \mathrm{diag}(\boldsymbol{\delta}^2))$ and $N(\mathbf{0}, \mathbf{C})$ are used for the definition of the most common algorithms. Figure 2.1 provides a sketch of the corresponding mutation ellipsoids, i.e., isolines of the probability density functions, embedded in a hypothetical two-dimensional fitness function.

The simplest case of generating the mutation $\mathbf{x}'$ from $\mathbf{x}$ is based on using $\mathbf{B} = \mathbf{I}$ and $\mathbf{D} = \sigma\mathbf{I}$ with a global step size $\delta \in \mathbb{R}^+$ for matrices $\mathbf{B}$ and $\mathbf{D}$ as used in Eq. 2.8.

[5] The normal distribution achieves maximum entropy among the distributions on the real domain. (See [64] for more details.)

[6] A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive definite iff $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ for all $\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ [17].

[7] For an orthogonal matrix $\mathbf{A}$, $\mathbf{A}\mathbf{A}^T = \mathbf{A}^T\mathbf{A} = \mathbf{I}$ holds.

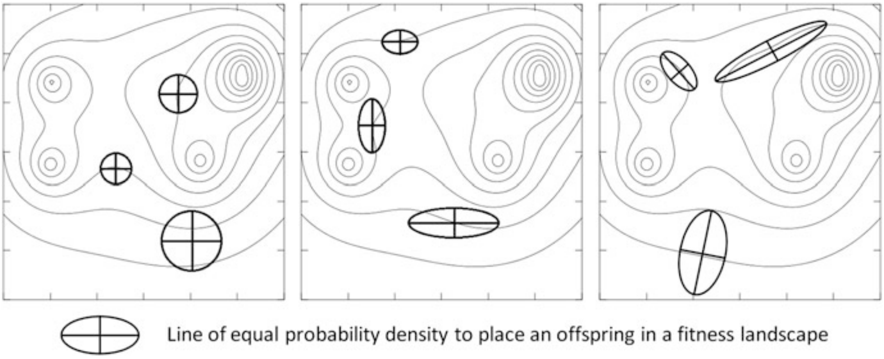
Line of equal probability density to place an offspring in a fitness landscape

Fig. 2.1 Mutation ellipsoids representing $N(\mathbf{0}, \mathbf{I})$, $N(\mathbf{0}, \mathrm{diag}(\boldsymbol{\delta}^2))$ and $N(\mathbf{0}, \mathbf{C})$ (from *left* to *right*)

$$\mathbf{x}' = \mathbf{x} + \delta \cdot N(\mathbf{0}, \mathbf{I}) \tag{2.9}$$

This corresponds with spheres with individual radii defined by δ, as indicated in the left part of Fig. 2.1. This case of an offspring distribution is called isotropic.

To turn the spheres into anisotropic ellipsoids with main axes parallel to the coordinate axes, as shown in the middle of Fig. 2.1, matrix $\mathbf{D}$ in Eq. 2.8 must be turned into a diagonal matrix $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_n)^T \in \mathbb{R}^n$ with different entries on the main diagonal. As in the previous case, $\mathbf{B}$ is a diagonal matrix:

$$\begin{aligned}\mathbf{x}' &= \mathbf{x} + \mathbf{I}\mathrm{diag}(\boldsymbol{\delta})N(\mathbf{0}, \mathbf{I}) \\ &= \mathbf{x} + N(\mathbf{0}, \mathrm{diag}(\boldsymbol{\delta}^2))\end{aligned} \tag{2.10}$$

The length ratios of the main axes of the mutation ellipsoids depend on the ratios between corresponding components of the vector δ. A rotation of mutation hyperellipsoids with respect to the coordinate axes, as shown in the rightmost part of Fig. 2.1, is achieved by using a covariance matrix $\mathbf{C}$ with off-diagonal entries different from zero. This case is denoted by the term correlated mutation. In contrast with the two previous cases, the matrix $\mathbf{B}$ is not just an identity matrix:

$$\begin{aligned}\mathbf{x}' &= \mathbf{x} + \mathbf{B}\mathrm{diag}(\delta)N(\mathbf{0}, \mathbf{I}) \\ &= \mathbf{x} + \mathbf{B}N(\mathbf{0}, \mathrm{diag}(\boldsymbol{\delta}^2)) \\ &= \mathbf{x} + N(\mathbf{0}, \mathbf{C})\end{aligned} \tag{2.11}$$

The choice of one of the three cases explained above has a direct impact on the complexity of the endogenous parameters controlling the multivariate normal distribution. In general, if n denotes the dimensionality of the search space, the number of endogenous strategy parameters in case of Eq. 2.9 is $O(1)$, i.e., constant. In case of 2.10 a vector of size $O(n)$ of endogenous parameters is required, and adaptation of an arbitrary covariance matrix, i.e., a symmetric $n \times n$-matrix, according to Eq. 2.11, requires $O(n^2)$ endogenous parameters.

For defining algorithm DR3 in Sect. 2.2.1 and for all algorithms based on the CMA-ES, the so-called *line distribution* [31] is of special interest: For $\mathbf{u} \in \mathbb{R}^n$, the distribution $N(\mathbf{0}, \mathbf{u}\mathbf{u}^T)$ is a multivariate normal distribution with the variance $\|\mathbf{u}\|^2$ in the direction of the vector $\mathbf{u}$. It is the normal distribution with highest probability of generating $\mathbf{u}$.

2.1.3.2 Relationship Between Covariance Matrix and Hessian

In the previous section, using a multivariate normal distribution was motivated by certain requirements which should hold for the mutation operator. In this section, we will clarify why it is useful to use an arbitrary covariance matrix, as in Eq. 2.11, for adaptation.

Any differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ can be approximated by a Taylor series expansion in the vicinity of a position[8] $\tilde{\mathbf{x}} \in \mathbb{R}^n$. Cutting off the Taylor series after the quadratic term, the following approximation is obtained:

$$f(\mathbf{x}) \approx f(\tilde{\mathbf{x}}) + (\mathbf{x} - \tilde{\mathbf{x}})^T \nabla f(\tilde{\mathbf{x}}) + \frac{1}{2}(\mathbf{x} - \tilde{\mathbf{x}})^T \nabla^2 f(\tilde{\mathbf{x}})(\mathbf{x} - \tilde{\mathbf{x}}) \tag{2.12}$$

Here, $\nabla f(\tilde{\mathbf{x}})$ denotes the gradient, and $\nabla^2 f(\tilde{\mathbf{x}})$ is the symmetric, positive definite *Hessian*, denoted by $\mathbf{H}$ in the following. For a quadratic function f, the Taylor series expansion is exact, and $\mathbf{H}$ contains information about the shape of the isolines of f. In general, these are ellipsoids, as shown in the rightmost part of Fig. 2.1. Hansen describes the relationship between the Hessian $\mathbf{H}$ and the covariance matrix $\mathbf{C}$ of a distribution $N(\mathbf{0}, \mathbf{C})$ informally [29]. It is argued that using $\mathbf{C} = \mathbf{H}^{-1}$ for optimizing a quadratic function is equivalent to using $\mathbf{C} = \mathbf{I}$ for optimizing an isotropic function, such as the sphere function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}\mathbf{x}^T$.

In other words: Adapting an arbitrary covariance matrix simplifies the optimization by transforming the objective function into an isotropic function. A more formal description of this topic can be found in Rudolph's work, e.g., in the section *Advanced Adaptation Techniques in* $\mathbb{R}^n$ in [58], and also in [55].

2.2 Algorithms

This section contains descriptions of the key variants of evolution strategies in chronological order of their publication. On a high level, we differentiate between the two main Sects. 2.2.1 and 2.2.2, with the first one corresponding with the time frame 1964 until 1996.

[8] See Sect. 6.2.2.3 in [17].

This first Sect. 2.2.1 describes five main algorithms, namely, the (1+1)-ES as the historically first version of an evolution strategy and the (μ, λ)-MSC-ES (in [58] also called CORR-ES) as the first evolution strategy which adapts an arbitrary covariance matrix (see Sect. 2.1.3 for an explanation). The first derandomized algorithm variants, DR1, DR2, and DR3, complete this selection of older variants of evolution strategies. Their choice is motivated by the fact that they are derandomization steps towards the CMA-ES (see also [63]).

The second main Sect. 2.2.2 describes modern evolution strategies, a term which is used in this book to denote the CMA-ES and algorithms based on it. This distinction might seem somewhat arbitrary, but in fact the development of the CMA-ES defined a turning point in the history of evolution strategies, for two main reasons: First, the CMA-ES is the first algorithm which adapts a covariance matrix in a completely derandomized way. Second, the CMA-ES is seen by many authors as the state of the art in evolution strategies (e.g., [6,13,15,26,35,58,63], and [66]).

2.2.1 *From the (1+1)-ES to the CMA-ES*

2.2.1.1 (1+1)-ES

The foundation of the first evolution strategy was laid in the 1960s at the Technical University of Berlin by three students, namely Hans-Paul Schwefel, Ingo Rechenberg, and Peter Bienert. As described in [8] or [58], standard methods for solving black-box optimization problems, such as gradient-based methods (see [44]), were not able to deliver satisfactory solution quality for certain optimization problems in engineering applications. Inspired by lectures about biological evolution, they aimed at developing a solution method based on principles of variation and selection. In its first version, a very simple evolution loop without any endogenous parameters was used [59]. This algorithm generates a single offspring $\mathbf{x}' = \mathbf{x} + (N_1(0,\sigma), \dots, N_n(0,\sigma))^T = \mathbf{x} + \sigma \cdot N(\mathbf{0}, \mathbf{I})$ from a single parent individual $\mathbf{x} \in \mathbb{R}^n$. If the offspring performs better than its parent (in terms of fitness), it becomes the new parent. Otherwise, the parent remains. The standard deviation σ of the normal distribution was a fixed scalar value.

According to [53], by pure luck the value of σ was chosen in a way that made this first approach towards a (1+1)-ES successful. Only later on, the necessary step size adaptation was added to the algorithm [52]. Based on two fitness functions, the so-called corridor model[9] and the so-called sphere model,[10] a theoretical result

[9] The rectangular corridor model according to [8]: $f_1(\mathbf{x}) = c_0 + c_1 \cdot x_1$ if the constraints $g_j(\mathbf{x}) : x_j \le b$ with $b \in \mathbb{R}^+$ for $j \in \{2, \dots, n\}$ are fulfilled, $f_1(\mathbf{x}) = \infty$ otherwise.

[10] The sphere model according to [8]: $f_2(\mathbf{x}) = c_0 + c_1 \cdot \sum_{n}^{i=1} (x_i - x_i^*)^2$.

Algorithm 2.3 (1+1)-ES

```
$P_0 \leftarrow \{\mathbf{x}\}$
$\phi \leftarrow f(\mathbf{x})$
$p_S \leftarrow 0$
initialize archive $A$ for storing successful mutations
$t \leftarrow 0$
repeat
    $t \leftarrow t + 1$
    $\mathbf{x}' \leftarrow \mathbf{x} + \sigma \cdot \mathbf{N}(\mathbf{0}, \mathbf{I})$
    $\phi' \leftarrow f(\mathbf{x}')$
    if $\phi' < \phi$ then
        $\mathbf{x} \leftarrow \mathbf{x}'$
        $\phi \leftarrow \phi'$
        store success in $A$
    else
        store failure in $A$
    end if
    $P_t \leftarrow \{\mathbf{x}\}$
    if $t \mod n = 0$ then
        get #$successes$ and #$failures$ from at most $10n$ entries in $A$
        $p_S = \frac{\#successes}{\#successes + \#failures}$
        $\sigma' \leftarrow \begin{cases} \sigma \cdot c \text{ if } p_S < 1/5 \\ \sigma / c \text{ if } p_S > 1/5 \\ \sigma \text{ if } p_S = 1/5 \end{cases}$
    end if
    $\sigma \leftarrow \sigma'$
until termination criterion fulfilled
```

was derived for introducing step size adaptation: Maximum convergence velocity (i.e., speed of progress of the optimization) is achieved when about 1/5 of all mutations are successful, i.e., improvements over their parent.[11] This insight led to the development of the so-called 1/5-success rule for step size adaptation. If about 1/5 of all mutations are successful, the step size is optimal and no adaptation is required. If the success rate falls below 1/5, the step size needs to be reduced. If it grows above 1/5, the step size needs to be increased. To obtain the new step size $\sigma' = \sigma \cdot c^{\{-1,1\}}$, the previous σ is decreased or increased, respectively, by multiplication or division by $0.817 \leq c \leq 1$. The recommended value of $c = 0.817$ was derived by Schwefel according to theoretical arguments about step size adaptation speed [61]. The step size adaptation according to the above rule is applied each n iterations of the algorithm, and the success rate p_S is measured over a sliding window of the last $10 \cdot n$ mutations [8]. The pseudocode of the (1+1)-ES according to [8] is shown in Algorithm 2.3.

[11] The exact values are 0.184 and 0.2025 for the corridor and sphere models, respectively [8].

2.2.1.2 (μ, λ)-MSC-ES

The (μ, λ)-MSC-ES[12] was the very first evolution strategy capable of adapting an arbitrary covariance matrix. The algorithm was developed by Schwefel [62] and is also called (μ, λ)-CORR-ES [58]. In this strategy, the covariance matrix is obtained as a product of $n(n-1)/2$ rotation matrices, where a single rotation matrix R_{ij} for a rotation angle α between axis i and axis j, with $i, j \in \{1, \ldots, n\}$ and $i \neq j$, is given by an identity matrix, extended by the entries $R(i,i) = R(j,j) = \cos \alpha_{ij}$ and $R(i,j) = -R(j,i) = -\sin \alpha_{ij}$.

Indeed, this method is able to generate arbitrary correlated mutations, as proven by Rudolph [55]. In the framework of the (μ, λ)-MSC-ES, endogenous strategy parameters are modified by means of the so-called self-adaptation principle. For self-adaptation, an individual consists not only of the decision parameters $\mathbf{x}$, but also contains an additional vector $\sigma \in \mathbb{R}^n_+$ of step sizes and a vector $\alpha \in (-\pi, \pi]^{n(n-1)/2}$ of rotation angles. The underlying idea of mutative step size adaptation is based on the assumption of individuals with good settings of strategy parameters to generate good offspring, such that the good strategy parameters survive selection. Recombination of decision parameters and endogenous strategy parameters is performed through global intermediary recombination, i.e., by averaging all of the μ parents. Concerning the exogenous strategy parameters, the local and global learning rates τ and τ' need to be set. Following [8], after Schwefel [61], the settings $\tau = \frac{1}{\sqrt{2\sqrt{n}}}$ and $\tau' = \frac{1}{2\sqrt{n}}$ are recommended, depending only on the problem dimensionality n. Pseudocode of the (μ, λ)-MSC-ES is provided in Algorithm 2.4. Concerning the population sizes, we are using $\mu = 15$ and $\lambda = 7 \cdot \mu = 105$ throughout this book, close to the recommendations in [63].

2.2.1.3 DR1

The (μ, λ)-MSC-ES as described in the previous section is based on mutative self-adaptation for step sizes $\delta \in \mathbb{R}^n_+$. However, as Ostermeier et al. [47] claim, self-adaptation of individual step sizes is not possible in the case of small population sizes, and they identify two key reasons: First, a successful mutation of the decision parameters is not necessarily caused by a good step size, but can also be due to an advantageous instantiation of the normally distributed random vector (i.e., a lucky sample). Second, there is a conflict between the goals of maintaining a large variance of step sizes within one generation and avoiding too large fluctuations of step sizes between successive generations. The first derandomized evolution strategy, abbreviated DR1,[13] solves the first problem by using the length of the most successful mutation step within one generation (i.e., the one that yielded the best

[12] MSC is an abbreviation of *mutative self-adaptation of covariances*.

[13] In the original publication it is called $(1, \lambda)$-ES with *derandomized mutative step size*.

Algorithm 2.4 (μ, λ)-MSC-ES

```
initialize population
$P^{(0)} \leftarrow \{(\mathbf{x}_1, \sigma_1, \alpha_1), \ldots, (\mathbf{x}_\mu, \sigma_\mu, \alpha_\mu)\}$
$t \leftarrow 0$
repeat
    $t \leftarrow t + 1$
    // recombination
    $\bar{x} \leftarrow \frac{1}{\mu} \sum_{i=1}^{\mu} \mathbf{x}_i$
    $\bar{\sigma} \leftarrow \frac{1}{\mu} \sum_{i=1}^{\mu} \sigma_i$
    $\bar{\alpha} \leftarrow \frac{1}{\mu} \sum_{i=1}^{\mu} \alpha_i$
    for $i = 1 \rightarrow \lambda$ do
        // mutation
        $\eta \leftarrow \tau' \cdot N(0, 1)$
        $\sigma_i \leftarrow \bar{\sigma} \cdot \exp(\eta + \tau \cdot N(\mathbf{0}, \mathbf{I}))$
        $\alpha_i \leftarrow \bar{\alpha} + \beta \cdot N(\mathbf{0}, \mathbf{I})$
        $\mathbf{C} \leftarrow \prod_{i=1}^{n-1} \prod_{j=i+1}^{n} R_{ij}$
        $\mathbf{x}_i \leftarrow \bar{\mathbf{x}} + \mathbf{C} \cdot \sigma_i \cdot N(\mathbf{0}, \mathbf{I})$
        // evaluation
        $\phi_i \leftarrow f(\mathbf{x}_i)$
    end for
    // selection
    $P^{(t)}$ are the $\mu$ best $(\mathbf{x}_i, \sigma_i, \alpha_i)$ from $1 \leq i \leq \lambda$
until termination criterion fulfilled
```

offspring) for controlling step size adaptation [47]. The second problem is solved by using a factor $\xi \in \{\frac{5}{7}, \frac{7}{5}\}$ to provide sufficient variance of step sizes within one generation, and to dampen[14] this factor by applying an exponent β with $0 < \beta < 1$ for step size adaptation, to reduce undesired fluctuations [47]. An offspring $\mathbf{x}'$ of a parent $\mathbf{x}$ is then generated as follows:

$$\mathbf{x}' = \mathbf{x} + \xi \cdot \delta \otimes \mathbf{z} \text{ where } \mathbf{z} = N(\mathbf{0}, \mathbf{I})$$

Adaptation of step sizes δ is based on the most successful $\mathbf{z}$ (i.e., the normally distributed vector sample which generated the best offspring during this generation), which is first transformed as follows:

$$\xi_{\mathbf{z}} = \left(\exp\left(|z_1| - \sqrt{2/\pi}\right), \ldots, \exp\left(|z_n| - \sqrt{2/\pi}\right)\right)^T$$

Combined with the exponents β and $\beta_{scal} \in \mathbb{R}$ for damping the adaptation, as well as ξ and $\xi_{\mathbf{z}}$ of the best mutation, the new step sizes δ' are obtained as follows:

$$\delta' = (\xi)^{\beta} \cdot (\xi_{\mathbf{z}})^{\beta_{scal}} \otimes \delta$$

[14] This way, adapting the step size by a factor ξ requires at least $1/\beta > 1$ generations.

Algorithm 2.5 DR1

```
initialize $\mathbf{x}$, $\boldsymbol{\delta} \leftarrow (1, \ldots, 1)^T$
$t \leftarrow 0$
repeat
    $t \leftarrow t + 1$
    for $i = 1 \to \lambda$ do
        $\mathbf{z}_i \leftarrow N(\mathbf{0}, \mathbf{I})$
        $\mathbf{x}_i \leftarrow \mathbf{x} + \xi_i \cdot \boldsymbol{\delta} \otimes \mathbf{z}_i$ where $P(\xi_i = \frac{5}{7}) = P(\xi_i = \frac{7}{5}) = \frac{1}{2}$
        $\phi_i \leftarrow f(\mathbf{x}_i)$
    end for
    $sel \leftarrow i$ with best value of $\phi_i$
    $\mathbf{x} \leftarrow \mathbf{x}_{sel}$
    $\xi_{\mathbf{z}_{sel}} = \left(\exp\left(|z_{sel_1}| - \sqrt{2/\pi}\right), \ldots, \exp\left(|z_{sel_n}| - \sqrt{2/\pi}\right)\right)^T$
    $\delta \leftarrow (\xi_{sel})^{\beta} \left(\xi_{\mathbf{z}_{sel}}\right)^{\beta_{scal}} \otimes \delta$
until termination criterion fulfilled
```

Pseudocode of the DR1 evolution strategy is given in Algorithm 2.5. Concerning the offspring population size λ, a constant setting of $\lambda = 10$, independently of dimensionality n, was used in [47]. The DR1 algorithm is based on a single parent individual ($\mu = 1$), and sometimes also denoted as $(1, 10)$-DR1-ES. Ostermeier et al.[47] recommends for the exponents β and β_{scal} the following values:

$$\beta = \sqrt{1/n}$$
$$\beta_{scal} = 1/n$$

2.2.1.4 DR2

The DR2 evolution strategy[15] represents the next step of derandomization for evolution strategies [48]. The creation of an offspring by mutation is parameterized by a global step size δ and local step sizes $\boldsymbol{\delta}_{scal} \in \mathbb{R}^n$:

$$\mathbf{x}' = \mathbf{x} + \delta \cdot \boldsymbol{\delta}_{scal} \otimes \mathbf{z} \text{ where } \mathbf{z} = N(\mathbf{0}, \mathbf{I})$$

As in DR1, adaptation of step sizes is based on the most successful $\mathbf{z}$. However, in addition to information about the most successful mutation of the current generation, the most successful mutation steps of previous generations are also taken into account, thereby accumulating information over generations. The accumulation takes place in a vector $\boldsymbol{\zeta} \in \mathbb{R}^n$, using a factor $c \in (0, 1]$ to control the weight of previous generations in contrast to the current one:

$$\boldsymbol{\zeta}' = (1 - c) \cdot \boldsymbol{\zeta} + c \cdot \mathbf{z}_{sel} \qquad (2.13)$$

[15] In the original paper, the algorithm is called $(1, \lambda)$-ES with *derandomized mutative step size control using accumulated information.*

Algorithm 2.6 DR2

```
initialize $\mathbf{x}$, $\boldsymbol{\zeta} \leftarrow \mathbf{0}$, $\delta \leftarrow 1$, $\boldsymbol{\delta}_{scal} \leftarrow (1, \ldots, 1)^T$
$t \leftarrow 0$
repeat
    $t \leftarrow t + 1$
    for $i = 1 \rightarrow \lambda$ do
        $\mathbf{z}_i \leftarrow N(\mathbf{0}, \mathbf{I})$
        $\mathbf{x}_i \leftarrow \mathbf{x} + \delta \cdot \boldsymbol{\delta}_{scal} \otimes \mathbf{z}_i$
        $\phi_i \leftarrow f(\mathbf{x}_i)$
    end for
    $sel \leftarrow i$ with best value of $\phi_i$
    $\boldsymbol{\zeta}' \leftarrow (1 - c) \cdot \boldsymbol{\zeta} + c \cdot \mathbf{z}_{sel}$
    $\delta' \leftarrow \delta \cdot \left(\exp\left(\frac{\|\boldsymbol{\zeta}'\|}{\sqrt{n} \cdot \sqrt{\frac{c}{2-c}}} - 1 + \frac{1}{5n}\right)\right)^{\beta}$
    $\boldsymbol{\delta}'_{scal} \leftarrow \boldsymbol{\delta}_{scal} \otimes \left(\frac{|\zeta'_i|}{\sqrt{\frac{c}{2-c}}} + \frac{7}{20}\right)^{\beta_{scal}}$
    $\mathbf{x} \leftarrow \mathbf{x}_{sel}$
    $\boldsymbol{\zeta} \leftarrow \boldsymbol{\zeta}'$
    $\delta \leftarrow \delta'$
    $\boldsymbol{\delta}_{scal} \leftarrow \boldsymbol{\delta}'_{scal}$
until termination criterion fulfilled
```

Adaptation of step sizes δ and $\boldsymbol{\delta}_{scal}$ is then based on the updated mutation path $\boldsymbol{\zeta}'$:

$$\delta' = \delta \cdot \left(\exp\left(\frac{\|\boldsymbol{\zeta}'\|}{\sqrt{n}\sqrt{\frac{c}{2-c}}} - 1 + \frac{1}{5n}\right)\right)^{\beta}$$

$$\boldsymbol{\delta}'_{scal_i} = \boldsymbol{\delta}_{scal_i} \cdot \left(\frac{|\boldsymbol{\zeta}'_i|}{\sqrt{\frac{c}{2-c}}} + \frac{7}{20}\right)^{\beta_{scal}} \quad \forall i \in \{1, \ldots, n\}$$

Standard settings for the exponents β and β_{scal} as well as the parameter c are as follows:

$$\begin{aligned} \beta &= \sqrt{1/n} \\ \beta_{scal} &= 1/n \\ c &= \sqrt{1/n} \end{aligned}$$

The pseudocode of the DR2 evolution strategy is given in Algorithm 2.6.

2.2.1.5 DR3

The DR3 evolution strategy [33], also called $(1, \lambda)$-GSA-ES (*generating set adaptation*), is able to generate mutations according to an arbitrary multivariate normal distribution, corresponding to the adaptation of an arbitrary covariance matrix according to Eq. 2.11. This process is not based on implicitly using a covariance matrix, but on transforming an isotropic random vector $\mathbf{z} = N(\mathbf{0}, \mathbf{I})$ into a correlated random vector $\mathbf{y}$ by multiplication with a matrix[16] $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_m) \in \mathbb{R}^{n \times m}$.

As described in Sect. 2.1.3, this can be interpreted as superposition of multiple line distributions. For the number m of column vectors, $n^2 \leq m \leq 2n^2$ holds, with a smaller value of m providing a faster adaptation and a larger value of m a more accurate adaptation. Like in DR1, for variation of the global step size $\delta \in \mathbb{R}$ a factor $\xi \in \{\frac{2}{3}, \frac{3}{2}\}$ with $P(\xi_i = 2/3) = P(\xi_i = 3/2) = 1/2$ is used. To guarantee an approximately constant length of the column vectors in $\mathbf{B}$, $\mathbf{y}$ is adapted by using a factor c_m. Based on its parents $\mathbf{x}$, an offspring is then created as follows:

$$\mathbf{x}' = \mathbf{x} + \delta \cdot \xi \cdot \mathbf{y} \text{ where } \mathbf{y} = c_m \cdot \mathbf{B} N(\mathbf{0}, \mathbf{I})$$

The adaptation of endogenous strategy parameters is based on the selected $\mathbf{y}_{sel}$ and ξ_{sel}. The column vectors of matrix $\mathbf{B}$ are updated according to:

$$\begin{aligned} \mathbf{b}'_1 &= (1 - c) \cdot \mathbf{b}_1 + c \cdot (c_u \xi_{sel} \mathbf{y}_{sel}) \\ \mathbf{b}'_{i+1} &= \mathbf{b}_i \ \forall i \in \{1, \ldots, m-1\} \end{aligned}$$

Like with the previous versions of derandomized evolution strategies, the global step size δ is adapted based on the selected ξ_{sel}, by using a damping exponent β:

$$\delta' = \delta \cdot (\xi_{sel})^{\beta}$$

For the exogenous parameters, the standard settings are given in [33] as follows:

$$\begin{aligned} c &= \sqrt{1/n} \\ \beta &= \sqrt{1/n} \\ m &= \frac{3}{2} n^2 \\ c_m &= (1/\sqrt{m})(1 + 1/m) \\ c_u &= \sqrt{(2 - c)/c} \\ \lambda &= 10 \end{aligned}$$

The corresponding pseudocode of the DR3 evolution strategy is provided in Algorithm 2.7.

[16] The column vectors of the matrix $\mathbf{B}$ form a so-called *generating set*, which motivates the terminology *generating set adaptation*.

Algorithm 2.7 DR3

```
initialize $\mathbf{x}, \delta, \mathbf{B} \leftarrow (\mathbf{0}, N(\mathbf{0}, (1/n)\mathbf{I})) \in \mathbb{R}^{n\times m}$
$t \leftarrow 0$
repeat
  $t \leftarrow t + 1$
  for $i = 1 \rightarrow \lambda$ do
    $\mathbf{z}_i \leftarrow N(\mathbf{0}, \mathbf{I})$ where $\mathbf{z}_i \in \mathbb{R}^m$
    $\mathbf{y}_i \leftarrow c_m \cdot \mathbf{B}\mathbf{z}_i$
    $\mathbf{x}_i \leftarrow \mathbf{x} + \delta \cdot \xi_i \cdot \mathbf{y}_i$ where $P(\xi_i = 2/3) = P(\xi_i = 3/2) = 1/2$
    $\phi_i \leftarrow f(\mathbf{x}_i)$
  end for
  $sel \leftarrow i$ with best value of $\phi_i$
  $\mathbf{b} \leftarrow (1-c) \cdot \mathbf{b}_1 + c \cdot (c_u \xi_{sel} \mathbf{y}_{sel})$
  $\delta' \leftarrow \delta \cdot (\xi_{sel})^{\beta}$
  $\mathbf{B}' \leftarrow (\mathbf{b}, \mathbf{b}_1, \ldots, \mathbf{b}_{m-1})$
  $\mathbf{x} \leftarrow \mathbf{x}_{sel}$, $\delta \leftarrow \delta'$ and $\mathbf{B} \leftarrow \mathbf{B}'$
until termination criterion fulfilled
```

2.2.2 Modern Evolution Strategies

2.2.2.1 (μ_W, λ)-CMA-ES

Algorithms DR1, DR2 and DR3, as described in Sect. 2.2.1, are derandomized evolution strategies in the sense of adapting endogenous strategy parameters depending on the selected mutation vector. This has also been called the first level of derandomization [63]. In addition, the second level of derandomization aims at the following goals [63]:

- Increase the probability of generating the same mutation step again.
- Provide a direct control mechanism for the rate of change of strategy parameters.
- Keep the strategy parameters unchanged in case of random selection.

The so-called CMA-ES, as introduced in [31], meets these goals by means of two techniques, namely the *covariance matrix adaptation, CMA* and the *cumulative step size adaptation*, CSA, for adapting a global step size. The description of a CMA-ES as provided in [31] is focused on explaining these two techniques, and recombination in case of $\mu > 1$ is not discussed at all. Therefore, we will discuss the CMA-ES in this section as a (μ_W, λ)-CMA-ES with weighted intermediary recombination, as described in [29] and [32].[17] Using the notation for evolution strategies as introduced in Sect. 2.1.2, the algorithm ought to be denoted more precisely as $(\mu/\mu_W, \lambda)$-CMA-ES, with index W denoting the weighted recombination. However, the simplified notation is motivated by arguing that the notation μ/μ_W suggests two different numbers (μ and μ_W), although it is μ in

[17] According to [32], the suggestion to use weighted recombination within the CMA-ES is due to Ingo Rechenberg, based on personal communication in 1998.

both cases. Here, we adopt the simplified notation, and denote the CMA-ES with weighted recombination as (μ_W, λ)-CMA-ES.

Based on a parent $\mathbf{x}$, an offspring $\mathbf{x}'$ is then generated as follows:

$$\mathbf{x}' = \mathbf{x} + \sigma \mathbf{B}\mathbf{D}\mathbf{z} \text{ where } \mathbf{z} = N(\mathbf{0}, \mathbf{I})$$

Matrices $\mathbf{B}$ and $\mathbf{D}$ result from an eigendecomposition of the covariance matrix $\mathbf{C}$ according to Eq. 2.7, and $\sigma \in \mathbb{R}$ denotes the global step size. After generating and evaluating an offspring population of size λ according to this mutation operator, the μ best individuals of the offspring population are selected and undergo weighted intermediary recombination.

Weighted intermediary recombination is a generalization of classical global intermediary recombination. Weighted intermediary recombination is based on using μ weights $w_1 \geq w_2 \geq \ldots \geq w_\mu$ with $\sum_{i=1}^{\mu} w_i = 1$ for generating the new parent $\langle \mathbf{x} \rangle$ and the best mutation step $\langle \mathbf{y} \rangle$ as weighted averages:

$$\langle \mathbf{x} \rangle = \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}$$

$$\langle \mathbf{y} \rangle = \sum_{i=1}^{\mu} w_i \mathbf{B}\mathbf{D}\mathbf{z}_{i:\lambda}$$

For adapting the strategy parameters, the so-called *variance effective selection mass* μ_{eff} is required:

$$\mu_{\mathit{eff}} = \left(\sum_{i=1}^{\mu} w_i^2 \right)^{-1}$$

According to [29], $1 \leq \mu_{\mathit{eff}} \leq \mu$ holds, and for identical weights $w_i = \frac{1}{\mu}$ $(\forall i \in \{1, \ldots, \mu\})$: $\mu_{\mathit{eff}} = \mu$. In analogy with Eq. 2.13 for DR2, the strategy parameter adaptation techniques, CMA and CSA, use so-called *evolution paths* for accumulating strategy parameter information across several generations. The (μ_W, λ)-CMA-ES uses two evolution paths, $\mathbf{p}_c$ for the adaptation of the covariance matrix and $\mathbf{p}_\sigma$ for global step size adaptation. The evolution paths are updated as follows:

$$\mathbf{p}_c' = (1 - c_c) \cdot \mathbf{p}_c + h_\sigma \sqrt{c_c(2 - c_c)\mu_{\mathit{eff}}} \langle \mathbf{y} \rangle$$

$$\mathbf{p}_\sigma' = (1 - c_\sigma) \cdot \mathbf{p}_\sigma + \sqrt{c_\sigma(2 - c_\sigma)\mu_{\mathit{eff}}} \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^T \langle \mathbf{y} \rangle$$

For updating $\mathbf{p}_c$, the function h_σ is used, which is defined according to:

$$h_\sigma = \begin{cases} 1 & \text{if } \frac{\|\mathbf{p}_\sigma\|}{\sqrt{1-(1-c_\sigma)^{2(t+1)}}} < \left(\frac{7}{5} + \frac{2}{n+1}\right) E(\|N(\mathbf{0}, \mathbf{I})\|) \\ 0 & \text{otherwise} \end{cases}$$

The purpose of h_σ is to avoid an update of $\mathbf{p}_c$ to take information of the current generation t into account, when $\|\mathbf{p}_c\|$ becomes too large. The expectation $E(\|N(\mathbf{0},\mathbf{I})\|)$ of the length of a multivariate, normally distributed vector of dimensionality n, can be approximated (based on the gamma function[18]) as follows:

$$E(\|N(\mathbf{0},\mathbf{I})\|) = \sqrt{2}\Gamma(\frac{n+1}{2})/\Gamma(\frac{n}{2}) \approx \sqrt{n}\left(1 - \frac{1}{4n} + \frac{1}{21n^2}\right)$$

The covariance matrix adaptation is performed according to the equation below:

$$\mathbf{C}' = (1 - c_l - c_\mu)\mathbf{C} + c_l(\mathbf{p}_c\mathbf{p}_c^T + \delta(h_\sigma)\mathbf{C}) + c_\mu \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}\mathbf{y}_{i:\lambda}^T \tag{2.14}$$

The first term in the summation represents the contribution of the previous covariance matrix. The second term is called the *rank-one-update* and takes the information accumulated in the evolution path $\mathbf{p}_c$ into account. The third term, the so-called *rank-μ-update*, was introduced with the extension of the CMA-ES for population sizes with $\mu > 1$ [46]. The global step size σ is updated according to:

$$\sigma' = \sigma \cdot \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_\sigma\|}{E(\|N(\mathbf{0},\mathbf{I})\|)} - 1\right)\right)$$

For the exogenous strategy parameters of the (μ_W, λ)-CMA-ES, the following standard settings are defined in [29]:

$$\begin{aligned}
\lambda &= 4 + \lfloor 3\ln n \rfloor \\
\mu &= \lfloor \frac{\lambda}{2} \rfloor \\
w_i &= \frac{\ln(\frac{\lambda+1}{2}) - \ln i}{\sum_{j=1}^{\mu} \ln(\frac{\lambda+1}{2}) - \ln j} \text{ for } i \in \{1, \ldots, \mu\} \\
c_\sigma &= \frac{\mu_{\mathit{eff}} + 2}{n + \mu_{\mathit{eff}} + 5} \\
d_\sigma &= 1 + 2\max\left(0, \sqrt{\frac{\mu_{\mathit{eff}} - 1}{n+1}}\right) + c_\sigma \\
c_c &= \frac{4 + \mu_{\mathit{eff}}/n}{n + 4 + 2\mu_{\mathit{eff}}/n}
\end{aligned}$$

[18] See [17]: $\Gamma(n) = \int_0^\infty x^{n-1}\exp(-x)\,\mathrm{d}x$.

Algorithm 2.8 (μ_W, λ)-CMA-ES

```
initialize $\langle \mathbf{x} \rangle$
$\mathbf{p}_c \leftarrow \mathbf{0}$
$\mathbf{p}_\sigma \leftarrow \mathbf{0}$
$\mathbf{C} \leftarrow \mathbf{I}$
$t \leftarrow 0$
repeat
  $t \leftarrow t + 1$
  $\mathbf{B}$ and $\mathbf{D} \leftarrow$ eigendecomposition of $\mathbf{C}$
  for $i = 1 \to \lambda$ do
    $\mathbf{z}_i \leftarrow N(\mathbf{0}, \mathbf{I})$
    $\mathbf{y}_i \leftarrow \mathbf{BDz}_i$
    $\mathbf{x}_i \leftarrow \langle \mathbf{x} \rangle + \sigma \mathbf{y}_k$
    $f_i \leftarrow f(\mathbf{x}_i)$
  end for
  $\langle \mathbf{y} \rangle \leftarrow \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}$
  $\langle \mathbf{x} \rangle \leftarrow \langle \mathbf{x} \rangle + \sigma \langle \mathbf{y} \rangle = \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}$
  $\mathbf{p}_\sigma \leftarrow (1 - c_\sigma)\mathbf{p}_\sigma + \sqrt{c_\sigma(2 - c_\sigma)\mu_{eff}}\mathbf{BD}^{-1}\mathbf{B}^T \langle \mathbf{y} \rangle$
  $\sigma \leftarrow \sigma \cdot \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_\sigma\|}{E\|N(\mathbf{0},\mathbf{I})\|} - 1\right)\right)$
  $\mathbf{p}_c \leftarrow (1 - c_c)\mathbf{p}_c + h_\sigma \sqrt{c_c(2 - c_c)\mu_{eff}}\langle \mathbf{y} \rangle$
  $\mathbf{C} \leftarrow (1 - c_1 - c_\mu)\mathbf{C} + c_1(\mathbf{p}_c\mathbf{p}_c^T + \delta(h_\sigma)\mathbf{C}) + c_\mu \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}\mathbf{y}_{i:\lambda}^T$
until termination criterion fulfilled
```

$$c_1 = \frac{2}{\left(n + \frac{13}{10}\right)^2 + \mu_{eff}}$$

$$c_\mu = \min\left(1 - c_1, \alpha_\mu \frac{\mu_{eff} - 2 + 1/\mu_{eff}}{(n + 2)^2 + \alpha_\mu \mu_{eff}/2}\right) \text{ with } \alpha_\mu = 2$$

Putting it all together, the pseudocode of the (μ_W, λ)-CMA-ES is given in Algorithm 2.8.

2.2.2.2 LS-CMA-ES

The LS-CMA-ES [6] is a $(1, \lambda)$-ES implementing the idea to adapt the covariance matrix $\mathbf{C}$ based on the inverse Hessian $\mathbf{H}^{-1}$. The Hessian itself is estimated by solving the appropriate *least squares estimation* problem. Based on Theorem 5 in [55], it is known that this requires at least $m \geq \frac{1}{2}\left(n^2 + 3n + 4\right)$ tuples $(\mathbf{x}, f(\mathbf{x}))$. To achieve this, the algorithm saves all tuples $(\mathbf{x}, f(\mathbf{x}))$ in an archive A. Based on the Taylor series expansion (Eq. 2.12), the *least squares estimation* problem is defined through the following minimization task:

$$\min_{\mathbf{g} \in \mathbb{R}^n, \mathbf{H} \in \mathbb{R}^{n \times n}} \sum_{k=1}^{m} \left(f(\mathbf{x}_k) - f(\mathbf{x}_0) - (\mathbf{x}_k - \mathbf{x}_0)^T \mathbf{g} - \frac{1}{2}(\mathbf{x}_k - \mathbf{x}_0)^T \mathbf{H}(\mathbf{x}_k - \mathbf{x}_0) \right)^2 \tag{2.15}$$

The result of minimizing 2.15 provides estimators $\hat{\mathbf{g}}$ for the gradient and $\hat{\mathbf{H}}$ for the Hessian.

Since the Taylor series expansion up to the quadratic term provides only an approximation of the true fitness landscape at $\mathbf{x}_0$, we are also interested in obtaining an error measure $Q(\hat{g}, \hat{\mathbf{H}})$ of the estimate for deciding whether $\hat{\mathbf{H}}^{-1}$ can be used for covariance matrix adaptation. The following error measure is used for this purpose:

$$Q(\hat{g}, \hat{\mathbf{H}}) = \frac{1}{m} \sum_{k=1}^{m} \left(\frac{f(\mathbf{x}_k) - f(\mathbf{x}_0) - (\mathbf{x}_k - \mathbf{x}_0)^T \hat{\mathbf{g}} - \frac{1}{2}(\mathbf{x}_k - \mathbf{x}_0)^T \hat{\mathbf{H}} (\mathbf{x}_k - \mathbf{x}_0)}{f(\mathbf{x}_k) - f(\mathbf{x}_0) - (\mathbf{x}_k - \mathbf{x}_0)^T \hat{\mathbf{g}}} \right)^2 \tag{2.16}$$

Unfortunately, solving Eq. 2.15 and inverting $\hat{\mathbf{H}}$ by means of numerical methods requires algorithms with time complexity $O(n^6)$, so that, especially for large n, an execution of these steps in each generation is not affordable. To solve this problem, the LS-CMA-ES provides two different working modes, denoted LS and CMA, for adapting the covariance matrix.

In mode LS, an approximation of $\mathbf{H}$ is performed only each n_{upd} generations.[19] If the error Q falls below a required threshold Q_t, the covariance matrix $\mathbf{C} = \frac{1}{2}\hat{\mathbf{H}}^{-1}$ is used by the algorithm and remains unchanged until a new update after another n_{upd} generations is performed.

If Q is bigger than the threshold value Q_t, the LS-CMA-ES switches into mode CMA. Before explaining this mode, the creation of an offspring $\mathbf{x}'$ from the parent $\langle \mathbf{x} \rangle$ is defined below:

$$\mathbf{x}' = \langle \mathbf{x} \rangle + \sigma d N(\mathbf{0}, \mathbf{C}) \text{ where } d = \exp(\tau N(0, 1))$$

In addition to the covariance matrix $\mathbf{C}$, a global step size σ is used, which is updated by mutative step size adaptation. If b denotes the index of the best offspring, the global step size is changed according to $\sigma' = \sigma \cdot d_b$. Adapting the covariance matrix $\mathbf{C}$ is based on a *rank-one update* (i.e., the second term in Eq. 2.14) by using an evolution path $\mathbf{p}_c$:

$$\mathbf{p}'_c = (1 - c_c) \cdot \mathbf{p}_c + \frac{\sqrt{(c_c(2 - c_c))}}{\sigma} (\mathbf{x}_b - \langle \mathbf{x} \rangle)$$
$$\mathbf{C}' = (1 - c_{cov}) \cdot \mathbf{C} + c_{cov} \mathbf{p}_c (\mathbf{p}_c)^T$$

The evolution path $\mathbf{p}_c$ is also updated when operating in mode LS, to make sure $\mathbf{C}$ is updated based on up-to-date information when the algorithm switches into mode CMA.

The pseudocode of the LS-CMA-ES is given in Algorithm 2.9, and the exogenous strategy parameters are set as follows:

[19] With the additional condition for A to consist of at least $m = n^2$ tuples.

$$\begin{aligned}\lambda &= 10\\ \tau &= \frac{1}{\sqrt{n}}\\ n_{upd} &= 100\\ Q_t &= 10^{-3}\\ c_c &= \frac{4}{n+4}\\ c_{cov} &= \frac{2}{(n+\sqrt{2})^2}\end{aligned}$$

2.2.2.3 LR-CMA-ES

The LR-CMA-ES (*local restart*) extends the (μ_W, λ)-CMA-ES by introducing restarts [4]. The strategy introduces five criteria for identifying stagnation of the optimization process and, in case of stagnation, starts a new run of the (μ_W, λ)-CMA-ES. Each run of the (μ_W, λ)-CMA-ES initializes the starting point of the search and the strategy parameters anew, so that the runs are independent of each other. For defining the termination criteria, the tolerance values $T_x = \sigma 10^{-12}$ and $T_f = 10^{-12}$ are used. Any other exogenous parameters are the same as in the (μ_W, λ)-CMA-ES.

The first termination criterion, called *equalfunvalhist*, is satisfied if either the best fitness values $f(\mathbf{x}_{1:\lambda})$ of the last $\lceil 10 + 30n/\lambda \rceil$ generations are identical or the difference between their maximum and minimum values is smaller than T_x.

The second criterion, *TolX*, is satisfied if the components of the vector $\mathbf{v} = \sigma \mathbf{p}_c$ are all smaller than T_x, i.e., $v_i < T_x \ \forall i \in \{1, \ldots, n\}$.

The third criterion, *noeffectaxis*, takes changes with respect to the main coordinate axes induced by $\mathbf{C}$ into account. These are given by the eigenvectors $\mathbf{u}_i$ and eigenvalues γ_i, $i \in \{1, \ldots, n\}$, of $\mathbf{C}$, and they are found (normalized) in the columns of matrix $\mathbf{B}$ and the main diagonal elements of $\mathbf{D}$. The termination criterion does not check all main axes at once, but in generation t it takes the axis $i = t \bmod n$ into account. It is satisfied when $\frac{\sigma}{10}\sqrt{\gamma_i}\mathbf{u}_i \approx 0$.

The fourth criterion, *noeffectcoord*, analyzes changes with respect to the coordinate axes. It is satisfied if $\frac{\sigma}{5} C_{i,i} \approx 0 \ \forall i \in \{1, \ldots, n\}$.

Finally, the criterion *conditioncov* checks whether the condition number of the matrix $\mathbf{C}$, $\mathrm{cond}(\mathbf{C}) = \frac{\max(\{\gamma_1,\ldots,\gamma_n\})}{\min(\{\gamma_1,\ldots,\gamma_n\})}$ exceeds 10^{14}.

The pseudocode of the LR-CMA-ES, as shown in Algorithm 2.10, consists of a simple outer loop managing the restarts of the (μ_W, λ)-CMA-ES. The local termination criteria are exactly the five criteria introduced above for discovering stagnation. In contrast, the global termination criterion is the same as used in previous sections, see Sect. 2.1.2.

Algorithm 2.9 LS-CMA-ES

```
initialize $\langle \mathbf{x} \rangle, \sigma$
$\mathbf{C} \leftarrow \mathbf{I}$
Archive $A \leftarrow \emptyset$
$\mathbf{p}_c \leftarrow \mathbf{0}$
mode $\leftarrow$ LS
$t \leftarrow 0$
repeat
  $t \leftarrow t + 1$
  $\mathbf{B}$ and $\mathbf{D} \leftarrow$ eigendecomposition of $\mathbf{C}$
  for $i = 1 \rightarrow \lambda$ do
    $d_i \leftarrow \exp(\tau N(0, 1))$
    $\mathbf{x}_i \leftarrow \langle \mathbf{x} \rangle + \sigma \cdot d_i \mathbf{BD} N(\mathbf{0}, \mathbf{I})$
    $f_i \leftarrow f(\mathbf{x}_i)$
    $A \leftarrow A \cup \{(\mathbf{x}_i, f_i)\}$
  end for
  $b \leftarrow$ index of best offspring
  $\sigma \leftarrow \sigma \cdot d_b$
  $\mathbf{p}_c \leftarrow (1 - c_c)\mathbf{p}_c + \frac{\sqrt{c_c(2-c_c)}}{\sigma}(\langle \mathbf{x} \rangle - \mathbf{x}_b)$
  if mode = LS then
    $\mathbf{C}$ unchanged
  else if mode = CMA then
    $\mathbf{C} \leftarrow (1 - c_{cov})\mathbf{C} + c_{cov}\mathbf{p}_c\mathbf{p}_c^T$
  end if
  if $t$ modulo $n_{upd} = 0$ then
    Obtain $\hat{\mathbf{g}}$ and $\hat{\mathbf{H}}$ based on the last $n^2$ tuples of $A$ by solving Equation 2.15 where $\mathbf{x}_0 = \langle \mathbf{x} \rangle$.
    Obtain $Q(\hat{\mathbf{g}}, \hat{\mathbf{H}})$ from Equation 2.16
    if $Q(\hat{\mathbf{g}}, \hat{\mathbf{H}}) < Q_t$ then
      mode $\leftarrow$ LS
      $\mathbf{C} \leftarrow \left(\frac{1}{2}\hat{\mathbf{H}}\right)^{-1}$
    else
      mode $\leftarrow$ CMA
    end if
  end if
  $\langle \mathbf{x} \rangle \leftarrow \mathbf{x}_b$
until termination criterion fulfilled
```

Algorithm 2.10 LR-CMA-ES

```
repeat
  execute $(\mu_W, \lambda)$-CMA-ES (Algorithm 2.8) using the local termination criteria
until global termination criterion satisfied
```

2.2.2.4 IPOP-CMA-ES

The IPOP-CMA-ES [5] is an extension of the LR-CMA-ES as described in the previous section. Whenever a run of the (μ_W, λ)-CMA-ES is terminated due to a local termination criterion (as introduced for LR-CMA-ES), the population size is increased by a factor η for the next run of the (μ_W, λ)-CMA-ES. This strategy is

Algorithm 2.11 IPOP-CMA-ES

```
repeat
    execute $(\mu_W, \lambda)$-CMA-ES (Algorithm 2.8) using the local termination criteria
    $\mu \leftarrow \eta \cdot \mu$
    $\lambda \leftarrow \eta \cdot \lambda$
until global termination criterion satisfied
```

motivated by empirical investigations of the behavior of the (μ_W, λ)-CMA-ES with different population sizes for multimodal test functions [30]. As these investigations clarified, the global convergence properties of the algorithm improve with increasing population size. The corresponding pseudocode is given in Algorithm 2.11. When using non-integer values for η, the new number of parents μ and offspring λ are obtained by rounding. For η, the interval $\left[\frac{3}{2}, 5\right]$ is identified as a reasonable range, and the default value $\eta = 2$ is recommended.

2.2.2.5 (1+1)-Cholesky-CMA-ES

The (1+1)-Cholesky-CMA-ES [38] introduces a method for adapting the covariance matrix $\mathbf{C}$ implicitly, without using an eigendecomposition of $\mathbf{C}$. Consequently, the approach reduces the computational complexity within each generation from $O(n^3)$ to $O(n^2)$.

The algorithm is based on the so-called Cholesky decomposition[20] of the covariance matrix, $\mathbf{C} = \mathbf{A}\mathbf{A}^T$. As proven in [38], an update of the Cholesky factors $\mathbf{A}$ is possible without explicit knowledge of the covariance matrix $\mathbf{C}$. The corresponding lemma and theorem are stated here without proof. The lemma states that, for any vector $\mathbf{v} \in \mathbb{R}^n$ and $\varsigma = \frac{1}{\|\mathbf{v}\|^2}\left(\sqrt{1 + \|\mathbf{v}\|^2} - 1\right)$, the following equation holds:

$$\mathbf{I} + \mathbf{v}\mathbf{v}^T = \left(\mathbf{I} + \varsigma\mathbf{v}\mathbf{v}^T\right)\left(\mathbf{I} + \varsigma\mathbf{v}\mathbf{v}^T\right)$$

This lemma is required for the proof of the following theorem:

Theorem 2.2.1. *Let* $\mathbf{C} \in \mathbb{R}^n$ *be a symmetric, positive definite matrix with Cholesky decomposition* $\mathbf{C} = \mathbf{A}\mathbf{A}^T$. *Let* $\mathbf{C}' = \alpha\mathbf{C} + \beta\mathbf{v}\mathbf{v}^T$ *be an update of* $\mathbf{C}$ *with* $\mathbf{v}, \mathbf{z} \in \mathbb{R}^n$, $\mathbf{v} = \mathbf{A}\mathbf{z}$ *and* $\alpha, \beta \in \mathbb{R}^+$. *The updated Cholesky factor* $\mathbf{A}'$ *of* $\mathbf{C}'$ *is then given by* $\mathbf{A}' = \sqrt{\alpha}\mathbf{A} + \frac{\sqrt{\alpha}}{\|\mathbf{z}\|^2}\left(\sqrt{1 + \frac{\beta}{\alpha}\|\mathbf{z}\|^2} - 1\right)(\mathbf{A}\mathbf{z})\,\mathbf{z}^T$.

Based on a parent individual $\mathbf{x}$, an offspring $\mathbf{x}'$ is then created according to:

$$\mathbf{x}' = \mathbf{x} + \sigma\mathbf{A}\mathbf{z} \text{ with } \mathbf{z} = N(\mathbf{0}, \mathbf{I})$$

[20] Compare Sect. 19.2.1.2 in [17].

Using Theorem 2.2.1, the Cholesky factor $\mathbf{A}$ is adapted as follows:

$$\mathbf{A}' = c_a\mathbf{A} + \frac{c_a}{\|\mathbf{z}\|^2}\left(\sqrt{1 + \frac{(1-c_a^2)\|\mathbf{z}\|^2}{c_a^2}} - 1\right)\mathbf{A}\mathbf{z}\mathbf{z}^T,$$

with a constant exogenous strategy parameter c_a. The adaptation above is applied if the value of a measure $\bar{p}_s$ (explained in the following) is smaller than a threshold value p_t.

The adaptation of the global step size δ is in some ways similar to the 1/5-success rule of the (1+1)-ES (see Sect. 2.2.1). If the offspring is better than the parent, $\lambda_s = 1$ in the equation below, otherwise, $\lambda_s = 0$. These success indicators are accumulated across generations by using a learning rate c_p, resulting in an accumulated success rate $\bar{p}_s$:

$$\bar{p}_s = (1 - c_p)\bar{p}_s + c_p\lambda_s$$

Using this measure and its target value p_s^t for the success rate, the global step size σ is updated as follows:

$$\sigma' = \sigma \cdot \exp\left(\frac{1}{d}\left(\bar{p}_s - \frac{p_s^t}{1 - p_s^t}(1 - \bar{p}_s)\right)\right)$$

The pseudocode is given in Algorithm 2.12, and the default settings of the exogenous strategy parameters are:

$$p_s^t = \frac{2}{11}$$

$$p_t = \frac{11}{25}$$

$$c_a = \sqrt{1 - \frac{2}{n^2 + 6}}$$

$$c_p = \frac{1}{12}$$

$$d = 1 + \frac{1}{n}$$

2.2.2.6 Active-CMA-ES

The (μ_W, λ)-CMA-ES uses weighted recombination of the μ best offspring to generate a new point in the search space. As shown by Rudolph [57], the convergence velocity of an evolution strategy can be further increased by also taking

Algorithm 2.12 (1+1)-Cholesky-CMA-ES

```
initialize $\mathbf{x}$, $\sigma$
$\mathbf{A} \leftarrow \mathbf{I}$
$\bar{p}_s \leftarrow p_s^t$
repeat
    $\mathbf{z} \leftarrow N(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}' \leftarrow \mathbf{x} + \sigma \mathbf{A}\mathbf{z}$
    if $f(\mathbf{x}') \leq f(\mathbf{x})$ then
        $\lambda_s \leftarrow 1$
    else
        $\lambda_s \leftarrow 0$
    end if
    $\bar{p}_s \leftarrow (1 - c_p)\bar{p}_s + c_p \lambda_s$
    $\sigma \leftarrow \sigma \cdot \exp\left(\frac{1}{d}\left(\bar{p}_s - \frac{p_s^t}{1-p_s^t}(1 - \bar{p}_s)\right)\right)$
    if $f(\mathbf{x}') \leq f(\mathbf{x})$ then
        $\mathbf{x} \leftarrow \mathbf{x}'$
        if $\bar{p}_s \leq p_t$ then
            $\mathbf{A} \leftarrow c_a \mathbf{A} + \frac{c_a}{\|\mathbf{z}\|^2}\left(\sqrt{1 + \frac{(1-c_a^2)\|\mathbf{z}\|^2}{c_a^2}} - 1\right)\mathbf{A}\mathbf{z}\mathbf{z}^T$
        end if
    end if
until termination criterion satisfied
```

the worst offspring into account for recombination, however, with negative weights. The Active-CMA-ES [40] is based on this idea,[21] however, it is not used during the process of recombination,[22] but exclusively for adapting the covariance matrix. Therefore, the corresponding extension of the (μ_W, λ)-CMA-ES mainly consists of changing the covariance matrix adaptation method, modifying Eq. 2.14 of the (μ_W, λ)-CMA-ES within the Active-CMA-ES into:

$$\mathbf{C}' = \mathbf{C} \leftarrow (1 - c_c)\mathbf{C} + c_c \mathbf{p}_c \mathbf{p}_c^T + \beta \mathbf{Z} \text{ where}$$

$$\mathbf{Z} = \mathbf{BD}\left(\frac{1}{\mu}\sum_{k=1}^{\mu} \mathbf{z}_{k:\lambda}\mathbf{z}_{k:\lambda}^T - \frac{1}{\mu}\sum_{k=\lambda-\mu+1}^{\lambda} \mathbf{z}_{k:\lambda}\mathbf{z}_{k:\lambda}^T\right)(\mathbf{BD})^T$$

In addition, the exogenous parameter c_c is now modified to $c_c = \frac{2}{(n+\sqrt{2})^2}$. The parameter β has been tuned by means of an empirical investigation, which is described in detail in [39]. Its setting of $\beta = \frac{4\mu - 2}{(n+12)^2 + 4\mu}$ reflects a compromise between the conflicting goals of achieving a large convergence velocity on the one

[21] The term *active* is motivated by the fact that specifically the bad offspring individuals play an active role, although they would normally not be taken into account after selection has been applied.

[22] This is explicitly avoided due to the occurrence of numerical instabilities for certain objective functions; see [40].

Algorithm 2.13 Active-CMA-ES

```
initialize $\langle \mathbf{x} \rangle$
$\mathbf{p}_c \leftarrow \mathbf{0}$
$\mathbf{p}_\sigma \leftarrow \mathbf{0}$
$\mathbf{C} \leftarrow \mathbf{I}$
$t \leftarrow 0$
repeat
    $t \leftarrow t + 1$
    $\mathbf{B}$ and $\mathbf{D} \leftarrow$ from eigendecomposition of $\mathbf{C}$
    for $i = 1 \to \lambda$ do
        $\mathbf{z}_i \leftarrow N(\mathbf{0}, \mathbf{I})$
        $\mathbf{y}_i \leftarrow \mathbf{BDz}_i$
        $\mathbf{x}_i \leftarrow \langle \mathbf{x} \rangle + \sigma \mathbf{y}_k$
        $f_i \leftarrow f(\mathbf{x}_i)$
    end for
    $\langle \mathbf{y} \rangle \leftarrow \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}$
    $\langle \mathbf{x} \rangle \leftarrow \langle \mathbf{x} \rangle + \sigma \langle \mathbf{y} \rangle = \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}$
    $\mathbf{p}_\sigma \leftarrow (1 - c_\sigma)\mathbf{p}_\sigma + \sqrt{c_\sigma(2 - c_\sigma)\mu_{\mathit{eff}}}\mathbf{BD}^{-1}\mathbf{B}^T \langle \mathbf{y} \rangle$
    $\sigma \leftarrow \sigma \cdot \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_\sigma\|}{E\|N(\mathbf{0},\mathbf{I})\|} - 1\right)\right)$
    $\mathbf{p}_c \leftarrow (1 - c_c)\mathbf{p}_c + h_\sigma \sqrt{c_c(2 - c_c)\mu_{\mathit{eff}}}\langle \mathbf{y} \rangle$
    $\mathbf{Z} \leftarrow \mathbf{BD}\left(\frac{1}{\mu}\sum_{k=1}^{\mu} \mathbf{z}_{k:\lambda}\mathbf{z}_{k:\lambda}^T - \frac{1}{\mu}\sum_{k=\lambda-\mu+1}^{\lambda} \mathbf{z}_{k:\lambda}\mathbf{z}_{k:\lambda}^T\right)(\mathbf{BD})^T$
    $\mathbf{C} \leftarrow (1 - c_c)\mathbf{C} + c_c\mathbf{p}_c\mathbf{p}_c^T + \beta\mathbf{Z}$
until termination criterion satisfied
```

hand and ensuring that $\mathbf{C}$ remains positive definite, to drive the evolution strategy into a robust working regime. The pseudocode is provided in Algorithm 2.13, and the default settings of the exogenous strategy parameters are, except for c_c and β, identical to those used in the (μ_W, λ)-CMA-ES.

2.2.2.7 (μ,λ)-CMSA-ES

The (μ,λ)-CMSA-ES [13], more precisely denoted the $(\mu/\mu_I, \lambda)$-CMA-σ-SA-ES, reintroduces self-adaptation of the global step size σ, just like in the (μ, λ)-MSC-ES, into the algorithm. This approach is motivated by the fact that reintroducing self-adaptation decreases the number of exegenous strategy parameters to two,[23] consequently providing a simplification of the (μ_W, λ)-CMA-ES, which requires five exogenous strategy parameters. Offspring individuals $\mathbf{x}_i$ and their step sizes σ_i, $i \in \{1, \ldots, \lambda\}$, are created based on the parent $\mathbf{x}$, the global step size σ, and the matrices $\mathbf{B}$ and $\mathbf{D}$ (from an eigendecomposition of the covariance matrix $\mathbf{C}$), as follows:

[23] Population sizes μ and λ are not counted.

$$
\begin{aligned}
\sigma_i &= \sigma \cdot \exp\left(\tau N(0,1)\right) \\
\mathbf{s}_i &= \mathbf{BD}N(\mathbf{0},\mathbf{I}) \\
\mathbf{z}_i &= \sigma_i \cdot \mathbf{s}_i \\
\mathbf{x}_i &= \mathbf{x} + \mathbf{z}_i
\end{aligned}
$$

Recombination is based on identical weights $1/\mu$, resulting in averaging the μ best offspring. It is applied to the vectors $\mathbf{z}_{i:\lambda}$, $\mathbf{s}_{i:\lambda}$, and step sizes $\sigma_{i:\lambda}$, for $i \in \{1,\ldots,\mu\}$, and results in the vectors $\langle\mathbf{z}\rangle$, $\langle\mathbf{s}\rangle$ and the new global step size σ. The new parent $\mathbf{x}'$ is then obtained as $\mathbf{x}' = \mathbf{x} + \langle\mathbf{z}\rangle$. Vector $\langle\mathbf{s}\rangle$ is required for adapting the covariance matrix $\mathbf{C}$, and its update uses the learning rate τ_C by proceeding as follows:

$$
\mathbf{C}' = \left(1 - \frac{1}{\tau_C}\right)\mathbf{C} + \frac{1}{\tau_C}\langle\mathbf{s}\rangle\langle\mathbf{s}\rangle^T \tag{2.17}
$$

The default settings of the exogenous strategy parameters are:

$$
\begin{aligned}
\mu &= \max\left(\left\lfloor \frac{n}{10} \right\rfloor, 2\right) \\
\lambda &= 4\mu \\
\tau &= \frac{1}{\sqrt{2n}} \\
\tau_C &= 1 + \frac{n(n+1)}{2\mu}
\end{aligned}
$$

The pseudocode of the corresponding (μ,λ)-CMSA-ES is given in Algorithm 2.14.

2.2.2.8 sep-CMA-ES

The sep-CMA-ES [54] is a variation of the (μ_W, λ)-CMA-ES which reduces space and time complexity to reach $O(n)$, i.e., linear in n. This is achieved by using, instead of an arbitrary covariance matrix, just a diagonal matrix $\mathbf{D}$ as in Eq. 2.10. Consequently, this kind of evolution strategy is not able anymore to generate correlated mutations, in return for the advantage of saving the computationally intensive eigendecomposition of the covariance matrix $\mathbf{C}$. $\mathbf{D}$ can then be obtained from $\mathbf{C}$ by taking the square roots of the main diagonal elements of $\mathbf{C}$. The covariance matrix is adapted according to the following update rule:

$$
\mathbf{C}' = (1 - c_{cov})\mathbf{C} + \frac{1}{\mu_{eff}} c_{cov}\mathbf{p}_c(\mathbf{p}_c)^T + c_{cov}\left(1 - \frac{1}{\mu_{eff}}\right)\sum_{i=1}^{\mu} w_i \mathbf{D}\mathbf{z}_{i:\lambda}(\mathbf{D}\mathbf{z}_{i:\lambda})^T
$$

Algorithm 2.14 (μ,λ)-CMSA-ES

```
initialize $\mathbf{x}$, $\sigma$
$\mathbf{C} \leftarrow \mathbf{I}$
$\langle\sigma\rangle \leftarrow \sigma$
repeat
    $\mathbf{B}$ and $\mathbf{D} \leftarrow$ from eigendecomposition of $\mathbf{C}$
    for $i = 1 \rightarrow \lambda$ do
        $\sigma_i \leftarrow \langle\sigma\rangle \exp \tau N(0,1)$
        $\mathbf{s}_i \leftarrow \mathbf{BD}N(\mathbf{0},\mathbf{I})$
        $\mathbf{z}_i \leftarrow \sigma_i \cdot \mathbf{s}_i$
        $\mathbf{y}_i \leftarrow \mathbf{x} + \mathbf{z}_i$
        $f_i \leftarrow f(\mathbf{y}_i)$
    end for
    $\langle\mathbf{z}\rangle \leftarrow$ average of the best $\mu$ $\mathbf{z}_i, i \in \{1,\ldots,\lambda\}$
    $\langle\mathbf{s}\rangle \leftarrow$ average of the best $\mu$ $\mathbf{s}_i, i \in \{1,\ldots,\lambda\}$
    $\langle\sigma\rangle \leftarrow$ average of the best $\mu$ $\sigma_i, i \in \{1,\ldots,\lambda\}$
    $\mathbf{x} \leftarrow \mathbf{x} + \langle\mathbf{z}\rangle$
    $\mathbf{C} \leftarrow \left(1 - \frac{1}{\tau_C}\right)\mathbf{C} + \frac{1}{\tau_C}\langle\mathbf{ss}^T\rangle$
until termination criterion satisfied
```

Due to the reduced complexity of the covariance matrix, the learning rate c_{cov} can be increased to accelerate the adaptation process. The learning rate c_{cov} is then set as follows:

$$c_{cov} = \frac{n+2}{3}\left(\frac{1}{\mu_{eff}}\frac{2}{(n+\sqrt{2})^2} + (1 - \frac{1}{\mu_{eff}})\min\left(1, \frac{2\mu_{eff}-1}{(n+2)^2+\mu_{eff}}\right)\right)$$

All other settings of the sep-CMA-ES are identical to those used within the (μ_W, λ)-CMA-ES. The resulting pseudocode of the sep-CMA-ES is shown in Algorithm 2.15.

2.2.2.9 $(1 \overset{+}{,} \lambda_m^s)$-ES

The $(1 \overset{+}{,} \lambda_m^s)$-ES [16] introduces the two new concepts of *mirrored sampling* and *sequential selection*. These two mutually independent concepts change the algorithmic processes of offspring creation and their selection, and thus they do not establish a complete evolution strategy. The concept of *mirrored sampling* can be used within a $(1+\lambda)$-ES as well as a $(1,\lambda)$-ES. The application of *sequential selection* is only possible in the case of a plus-strategy, explaining also the use of the notation $\overset{+}{,}$. Furthermore, the indices s and m of λ represent the algorithmic concepts of *sequential selection* (s) and *mirrored sampling* (m), respectively.

The idea of *mirrored sampling* is to generate part of the offspring in a derandomized way by generating for a mutation vector $\mathbf{z}$ not only the offspring $\mathbf{x} + \mathbf{z}$, but also

Algorithm 2.15 sep-CMA-ES

```
initialize $\langle \mathbf{x} \rangle$
$\mathbf{C} \leftarrow \mathbf{I}$
$\mathbf{D} \leftarrow \mathbf{I}$
$\mathbf{p}_\sigma \leftarrow \mathbf{0}$
$\mathbf{p}_c \leftarrow \mathbf{0}$
$t \leftarrow 0$
repeat
    $t \leftarrow t + 1$
    for $i = 1 \rightarrow \lambda$ do
        $\mathbf{z}_i \leftarrow N(\mathbf{0}, \mathbf{I})$
        $\mathbf{x}_i \leftarrow \langle \mathbf{x} \rangle + \sigma \mathbf{D} \mathbf{z}_i$
    end for
    $\langle \mathbf{x} \rangle \leftarrow \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}$
    $\langle \mathbf{z} \rangle \leftarrow \sum_{i=1}^{\mu} w_i \mathbf{z}_{i:\lambda}$
    $\mathbf{p}_\sigma \leftarrow (1 - c_\sigma)\mathbf{p}_\sigma + \sqrt{c_\sigma(2 - c_\sigma)}\sqrt{\mu_{\mathit{eff}}}\langle \mathbf{z} \rangle$
    if $\frac{\|\mathbf{p}_\sigma\|}{\sqrt{1-(1-c_\sigma)^{2t}}} < \left(\frac{7}{5} + \frac{2}{n+1}\right) E(\|N(\mathbf{0}, \mathbf{I})\|)$ then
        $H_\sigma \leftarrow 1$
    else
        $H_\sigma \leftarrow 0$
    end if
    $\mathbf{p}_c \leftarrow (1 - c_c)\mathbf{p}_c + H_\sigma \sqrt{c_c(2 - c_c)}\sqrt{\mu_{\mathit{eff}}}\mathbf{D}\langle \mathbf{z} \rangle$
    $\mathbf{C} \leftarrow (1 - c_{cov})\mathbf{C} + \frac{c_{cov}}{\mu_{\mathit{eff}}}\mathbf{p}_c\mathbf{p}_c^T + c_c\left(1 - \frac{1}{\mu_{\mathit{eff}}}\right)\sum_{i=1}^{\mu} w_i \mathbf{D}\mathbf{z}_{i:\lambda}\,(\mathbf{D}\mathbf{z}_{i:\lambda})^T$
    $\sigma \leftarrow \sigma \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_\sigma\|}{E(\|N(\mathbf{0},\mathbf{I})\|}-1\right)\right)$
    $\mathbf{D} = \operatorname{diag}\left(\sqrt{C_{1,1}}, \ldots, \sqrt{C_{n,n}}\right)$
until termination criterion satisfied
```

the additional offspring $\mathbf{x} - \mathbf{z}$. These two offspring are obviously symmetrical[24] with respect to $\mathbf{x}$. As a potential application, mentioned in [3], *mirrored sampling* can increase the robustness of the *Evolutionary Gradient Search* algorithm and increase convergence velocity in the sphere model. Theoretical convergence rates for variants of the $(1 \overset{+}{,} \lambda_m^s)$-ES have been derived; see [16] for the corresponding results.

Sequential selection can be used to reduce the number of function evaluations. It is applied within a $(1 + \lambda)$-ES by sequentially executing the steps mutation and evaluation for single offspring individuals, rather than generating all λ offspring first and then evaluating their fitness. In *sequential selection*, as soon as an offspring has a better fitness than the parent, the offspring can replace the parent, and no more offspring need to be generated and evaluated. In this way, up to $\lambda - 1$ function evaluations can potentially be saved at each generation.

The two concepts can be used independently of each other, or in combination. As explained before, the $(1 \overset{+}{,} \lambda_m^s)$-ES does not constitute a complete evolution strategy, but rather a method for generating the parent $\langle \mathbf{x} \rangle'$ for the next generation based on the previous parent $\langle \mathbf{x} \rangle$ and a method *mutationOffset*, which generates a

[24] Instead of the term symmetrical, this is called *mirrored* in the context of this strategy.

Algorithm 2.16 $(1 \overset{+}{,} \lambda^s_m)$-ES

```
Input:   search point $\langle \mathbf{x} \rangle$ and a method mutationOffset
Output:   new search point $\langle \mathbf{x} \rangle'$
  $i \leftarrow 0$
  $j \leftarrow 0$
  while $i < \lambda$ do
    $i \leftarrow i + 1$
    $j \leftarrow j + 1$
    if (mirrored sampling) $\wedge$ ($j$ modulo $2 = 0$) then
      $\mathbf{x}_i \leftarrow \langle \mathbf{x} \rangle - \mathbf{z}_i$
    else
      $\mathbf{z}_i \leftarrow$ mutationOffset()
      $\mathbf{x}_i \leftarrow \langle \mathbf{x} \rangle + \mathbf{z}_i$
    end if
    if (sequential selection) $\wedge$ ($f(\mathbf{x}_i) < f(\langle \mathbf{x} \rangle)$) then
      $j \leftarrow 0$
      break
    end if
  end while
  $\langle \mathbf{x} \rangle' \leftarrow \mathrm{argmin}\,(\{f(\mathbf{x}_1), \ldots, f(\mathbf{x}_i)\})$
```

mutation step and is determined by the underlying evolution strategy. The approach is summarized in pseudocode in Algorithm 2.16.

2.2.2.10 xNES

The xNES algorithm (*exponential natural evolution strategies*) [26] is a $(1, \lambda)$-ES which adapts its endogenous strategy parameters by using the so-called *natural gradient* (see [1]). The idea was implemented for the first time in the context of NES (*natural evolution strategies*) [71] and was then developed further by introducing the eNES (*efficient natural evolution strategies*)[25] [66].

In the following, the underlying ideas of the xNES are briefly summarized, without giving detailed descriptions of the underlying concepts, such as, e.g., the *Fisher information matrix*. These fundamentals can be found in the original work of Glasmachers et al. and the corresponding references, see [26].

This family of evolution strategy algorithms also relies on the multivariate normal distribution $N(\langle \mathbf{x} \rangle, \mathbf{C})$ for generating correlated mutations of the current search point $\langle \mathbf{x} \rangle$. Similar to the $(1 + 1)$-Cholesky-CMA-ES (see Sect. 2.2.2.5), rather than working with the covariance matrix $\mathbf{C}$ explicitly, a Cholesky factor $\mathbf{A}$ with $\mathbf{C} = \mathbf{A}\mathbf{A}^T$ is used. The current search point and the covariance matrix are combined to form the tuple $\theta = (\langle \mathbf{x} \rangle, \mathbf{C})$, representing the quantities subject to adaptation within an xNES. Rewriting the probability density function of a normal distribution

[25] In [26] the eNES are called *exact natural evolution strategies*.

as a function of the current search point $\langle \mathbf{x} \rangle$ and the Cholesky factor $\mathbf{A}$, its probability density $N(\langle \mathbf{x} \rangle, \mathbf{C})$ turns into:

$$p(\mathbf{x}|\theta) = \frac{1}{\left(\sqrt{2\pi}\right)^n \det \mathbf{A}} \cdot \exp\left(-\frac{1}{2} \left\|\mathbf{A}^{-1} \cdot (\mathbf{x} - \langle \mathbf{x} \rangle)\right\|^2\right)$$

Given the distribution described by θ, the expectation $J(\theta)$ of the fitness becomes:

$$J(\theta) = E(f(\mathbf{x})|\theta) = \int f(\mathbf{x}) p(\mathbf{x}|\theta) d\mathbf{x}$$

The gradient of the expectation $J(\theta)$, $\nabla_\theta J(\theta)$, can be calculated by using the so-called *log-likelihood trick* according to

$$\nabla_\theta J(\theta) = \int \left(f(\mathbf{x}) \nabla \log\left(p(\mathbf{x}|\theta)\right)\right) p(\mathbf{x}|\theta) d\mathbf{x},$$

which can be approximated by Monte Carlo estimation based on the offspring individuals $\mathbf{x}_i$, $i \in \{1, \ldots, \lambda\}$:

$$\nabla_\theta J(\theta) \approx \frac{1}{\lambda} \sum_{i=1}^{\lambda} f(\mathbf{x}_i) \nabla \log\left(p(\mathbf{x}|\theta)\right).$$

For calculating the term $\nabla \log\left(p(\mathbf{x}|\theta)\right)$, we refer to [67]. Combining this with the *Fisher information matrix* (FIM) $\mathbf{F} \in \mathbb{R}^{N \times N}$, where $N = n + n(n+1)/2$, the natural gradient G is obtained as:

$$G = \mathbf{F}^{-1} \nabla_\theta J(\theta)$$

Use of G is motivated by the fact that it is invariant with respect to linear transformations, so that the gradient converges in a correlated search space pretty much like in an isotropic one.

The NES suffer from the disadvantage of their impracticable computational complexity of $O(n^6)$, caused by the explicit calculation of the FIM and its inversion. In contrast, the xNES do not require an explicit calculation of the FIM anymore. Based on using a so-called *exponential parameterization* (see Sect. 4.1 in [26]) a transformation of θ into *natural coordinates* (see Sect. 4.2 in [26]) is applied. Using step size δ and Cholesky factor $\mathbf{B}$, an offspring $\mathbf{x}$ is then generated from the parent $\langle \mathbf{x} \rangle$ according to:

$$\mathbf{x} = \langle \mathbf{x} \rangle + \delta \mathbf{B} \mathbf{z} \text{ where } \mathbf{z} = N(\mathbf{0}, \mathbf{I}) \tag{2.18}$$

Similar to weighted recombination, the xNES uses so-called *utility values* u_i. This approach is also called *fitness shaping* in the context of an xNES. Using the rank i given by the fitness values, utility values are calculated as follows:

$$u_i = \frac{\max\left(0, \log\left(\frac{\mu}{2}+1\right) - \log(i)\right)}{\sum_{j=1}^{\mu} \max\left(0, \log\left(\frac{\mu}{2}+1\right) - \log(i)\right)} - \frac{1}{\lambda}$$

Using the mutation vectors $\mathbf{z}_i$ from Eq. 2.18, the gradients $\mathbf{G}_M$ for the covariance matrix and $\mathbf{G}_\delta$ for the current search point are defined by:

$$\mathbf{G}_M = \frac{1}{2}\sum_{i=1}^{\lambda} u_i \left(\mathbf{z}_i\mathbf{z}_i^T - \mathbf{I}\right)$$

$$\mathbf{G}_\delta = \sum_{i=1}^{\lambda} u_i \mathbf{z}_i$$

For calculating the gradients, all λ offspring individuals are taken into account, i.e., a selection in the classical sense is not applied. Using those gradients and the learning rates η_x, η_σ and η_B, the new search point $\langle\mathbf{x}\rangle'$, the new step sizes σ', and the new Cholesky factor $\mathbf{B}'$ are calculated:

$$\begin{aligned}
\langle\mathbf{x}\rangle' &= \langle\mathbf{x}\rangle + \eta_x \cdot \mathbf{G}_\delta \\
\sigma' &= \sigma \cdot \exp\left(\frac{\eta_\sigma}{2n} \cdot \mathrm{tr}\left(\sum_{i=1}^{\lambda} u_i \cdot \left(\mathbf{z}_i\mathbf{z}_i^T - \mathbf{I}\right)\right)\right) \\
\mathbf{B}' &= \mathbf{B} \cdot \exp\left(\frac{\eta_B}{2} \cdot \mathbf{G}_M\right)
\end{aligned}$$

Here, the exponential function of a matrix $\mathbf{A}$ is defined by $\exp(\mathbf{A}) = \sum_{n=0}^{\infty} \frac{\mathbf{A}^n}{n!}$, see [26].

The resulting pseudocode of the xNES is given in Algorithm 2.17. The default parameters of the exogenous strategy parameters are as follows:

$$\begin{aligned}
\lambda &= 4 + \lfloor 3\log(n) \rfloor \\
\eta_x &= 1 \\
\eta_\sigma &= \frac{3}{5} \cdot \frac{3 + \log(n)}{n\sqrt{n}} \\
\eta_B &= \eta_\sigma
\end{aligned}$$

Algorithm 2.17 xNES

```
initialize $\langle \mathbf{x} \rangle$
$\mathbf{B} \leftarrow \mathbf{I}$
$\sigma \leftarrow \sqrt[d]{|\det \mathbf{B}|}$
for $i = 1 \rightarrow \lambda$ do
    $u_i \leftarrow \frac{\max\left(0, \log\left(\frac{\lambda}{2}+1\right)-\log(i)\right)}{\sum_{j=1}^{\lambda} \max\left(0, \log\left(\frac{\lambda}{2}+1\right)-\log(i)\right)} - \frac{1}{\lambda}$
end for
repeat
    for $i = 1 \rightarrow \lambda$ do
        $\mathbf{z}_i \leftarrow N(\mathbf{0}, \mathbf{I})$
        $\mathbf{x}_i \leftarrow \langle \mathbf{x} \rangle + \sigma \mathbf{B} \mathbf{z}_i$
    end for
    sort $\{(\mathbf{z}_i, \mathbf{x}_i)\}$ by $f(\mathbf{x}_i)$
    $\mathbf{G}_\delta \leftarrow \sum_{i=1}^{\lambda} u_i \cdot \mathbf{z}_i$
    $\mathbf{G}_M \leftarrow \sum_{i=1}^{\lambda} u_i \cdot \left(\mathbf{z}_i \mathbf{z}_i^T - \mathbf{I}\right)$
    $G_\sigma \leftarrow \mathrm{tr}(\mathbf{G}_M)/n$
    $\mathbf{G}_B \leftarrow \mathbf{G}_M - G_\sigma \cdot \mathbf{I}$
    $\langle \mathbf{x} \rangle \leftarrow \langle \mathbf{x} \rangle + \eta_x \cdot \sigma \mathbf{B} \cdot \mathbf{G}_\delta$
    $\sigma \leftarrow \sigma \cdot \exp\left(G_\sigma \cdot \frac{\eta_\sigma}{2}\right)$
    $\mathbf{B} \leftarrow \mathbf{B} \cdot \exp\left(\mathbf{G}_B \cdot \frac{\eta_B}{2}\right)$
until termination criterion satisfied
```

2.2.2.11 (1+1)-Active-CMA-ES

Extending the (1+1)-Cholesky-CMA-ES with the idea of the Active-CMA-ES to take information of unsuccessful offspring into account for covariance matrix adaptation consequently leads to the development of a hybrid, the (1+1)-Active-CMA-ES [2]. Instead of using an explicit covariance matrix $\mathbf{C} = \mathbf{A}\mathbf{A}^T$, the (1+1)-Active-CMA-ES works directly with the Cholesky factor $\mathbf{A}$ and its inverse $\mathbf{A}^{-1}$. The update of $\mathbf{A}$ has been defined previously, based on Theorem 2.2.1. In order to use $\mathbf{A}^{-1}$, an extended version of this theorem is required, which we state below (without proof, see [2]):

Theorem 2.2.2. *Let* $\mathbf{C} \in \mathbb{R}^{n \times n}$ *be a symmetric, positive definite matrix with Cholesky decomposition* $\mathbf{C} = \mathbf{A}\mathbf{A}^T$*, and let* $\mathbf{C}' = \alpha \mathbf{C} + \beta \mathbf{v}\mathbf{v}^T$ *be an update transformation of* $\mathbf{C}$ *where* $\mathbf{v} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$*,* $\alpha \in \mathbb{R}^+$ *and* $\beta \in \mathbb{R}$*. Let* $\mathbf{w} = \mathbf{A}^{-1}\mathbf{v}$ *with* $\alpha + \beta\|\mathbf{w}\|^2 > 0$ *and let* $\mathbf{C}' = \mathbf{A}'\mathbf{A}'^T$ *be the Cholesky decomposition of the updated matrix* $\mathbf{C}'$*. Then, the Cholesky factor* $\mathbf{A}'$ *and its inverse* $\mathbf{A}'^{-1}$ *are obtained as follows:* $\mathbf{A}' = \sqrt{\alpha}\mathbf{A} + \frac{\sqrt{\alpha}}{\|\mathbf{w}\|^2}\left(\sqrt{1 + \frac{\beta}{\alpha}\|\mathbf{w}\|^2} - 1\right)\mathbf{A}\mathbf{w}\mathbf{w}^T$ *and* $\mathbf{A}'^{-1} = \frac{1}{\sqrt{\alpha}}\mathbf{A}^{-1} - \frac{1}{\sqrt{\alpha}\|\mathbf{w}\|^2}\left(1 - \frac{1}{\sqrt{1+\beta\|\mathbf{w}\|^2/\alpha}}\right)\mathbf{w}\mathbf{w}^T\mathbf{A}^{-1}$.

The offspring $\mathbf{x}'$ is generated from its parent $\mathbf{x}$ according to:

$$\mathbf{x}' = \mathbf{x} + \sigma \mathbf{A}\mathbf{z} \text{ where } \mathbf{z} = N(\mathbf{0}, \mathbf{I})$$

As for the (1+1)-Cholesky-CMA-ES, the success rate p_s, i.e., the fraction of successful mutations, is updated by taking the learning rate c_p into account:

$$p_s' = \begin{cases} (1 - c_p)p_s + c_p & \text{if } f(\mathbf{x}') \leq f(\mathbf{x}) \\ (1 - c_p)p_s & \text{if } f(\mathbf{x}') > f(\mathbf{x}) \end{cases}$$

Based on the success rate p_s, a damping parameter $d \in \mathbb{R}^+$ and the target success rate p_t, the global step size σ is updated as follows:

$$\sigma' = \sigma \cdot \exp\left(\frac{1}{d}\frac{p_s - p_t}{1 - p_t}\right)$$

The algorithm uses $p_t = \frac{2}{11}$ which makes the update similar to the 1/5-success rule update mechanism of the (1+1)-ES.

If the offspring performs better than its parent, a positive Cholesky update is applied. In contrast to the (1+1)-Cholesky-CMA-ES, which uses the mutation step **z** for this update, the (1+1)-Active-CMA-ES relies on a search path **s**, accumulating successful mutation steps with a learning rate c and updating **s** as follows:

$$\mathbf{s}' = (1 - c)\mathbf{s} + \sqrt{c(2 - c)}\mathbf{A}\mathbf{z}$$

With a constant $c_c^+ > 0$ and the vector $\mathbf{w} = \mathbf{A}^{-1}\mathbf{s}$, the positive update of matrices $\mathbf{A}$ and $\mathbf{A}^{-1}$ can now be defined according to Theorem 2.2.2:

$$\mathbf{A}' = a\mathbf{A} + b(\mathbf{A}\mathbf{w})\mathbf{w}^T \text{ and} \tag{2.19}$$

$$\mathbf{A}^{-1'} = \frac{1}{a}\mathbf{A}^{-1'} - \frac{b}{a^2 + ab\|\mathbf{w}\|^2}\mathbf{w}(\mathbf{w}^T\mathbf{A}^{-1}) \text{ where} \tag{2.20}$$

$$a = \sqrt{1 - c_c^+} \text{ and}$$

$$b = \frac{\sqrt{1 - c_c^+}}{\|\mathbf{w}\|^2}\left(\sqrt{1 + \frac{c_c^+}{1 - c_c^+}\|\mathbf{w}\|^2} - 1\right)$$

In the case of an Active-CMA-ES, the $\lambda - \mu$ worst individuals are used for the negative update of the covariance matrix, and these individuals can be called the "especially bad" individuals. In the case of the corresponding (1+1)-strategy, as introduced here, this definition is not applicable. Instead, the (1+1)-Active-CMA-ES stores past function evaluations and defines an individual to be "especially bad", if its fitness value is worse than the fitness of its k-th predecessor. For an "especially bad" offspring, a negative update according to Eqs. 2.19 and 2.20 is performed, using modified values of the coefficients a and b. In contrast to the positive update, rather than the transformed search path $\mathbf{w} = \mathbf{A}^{-1}\mathbf{s}$ the vector $\mathbf{z}$ is used for the negative update:

$$a = \sqrt{1 + c_c^-}$$

$$b = \frac{\sqrt{1 + c_c^-}}{\|\mathbf{z}\|^2}\left(\sqrt{1 - \frac{c_c^-}{1 - c_c^-}\|\mathbf{z}\|^2} - 1\right)$$

To ensure a positive definite covariance matrix, $1 - \frac{c_c^-}{1+c_c^-}\|\mathbf{z}\|^2 > 0$ needs to hold for the constant c_c^-. Moreover, the convergence behavior of the algorithm can become unstable if the value of $1 - \frac{c_c^-}{1+c_c^-}\|\mathbf{z}\|^2$ is very close to zero. As a countermeasure, in case of $1 - \frac{c_c^-}{1+c_c^-}\|\mathbf{z}\|^2 < 1/2$, the value of c_c^- is provided with an upper bound of $1/(2\|\mathbf{z}\|^2)$.

The default settings of the exogenous parameters are:

$$\begin{aligned} d &= 1 + n/2 \\ c &= 2/(n+2) \\ c_p &= 1/12 \\ p_t &= 2/11 \\ c_c^+ &= \frac{2}{n^2 + 6} \\ c_c^- &= \frac{2}{5(n^{8/5} + 1)} \end{aligned}$$

The pseudocode of the (1+1)-Active-CMA-ES is given in Algorithm 2.18.

2.2.2.12 $(\mu/\mu_W, \lambda_{iid} + \lambda_m)$-ES

The $(\mu/\mu_W, \lambda_{iid} + \lambda_m)$-ES [7] is based on extending the idea of *mirrored sampling*, as described in Sect. 2.2.2.9 for a $(1 \overset{+}{,} \lambda_m^s)$-ES, for the case $\mu > 1$. The offspring population size is given by the number of samples λ_{iid} (independent, identically distributed samples from the mutation distribution) and the number of offspring, λ_m ($\lambda_m \leq \lambda_{iid}$), which are also subject to mirroring. Using *mirrored sampling* in combination with weighted recombination and cumulative step size adaptation (see Sect. 2.2.2.1) introduces a *bias* with respect to the step size, i.e., the step size is more than desirably reduced, thus potentially causing a premature stagnation effect for the algorithm. To avoid this issue, the concept of *pairwise selection* is introduced, i.e., it is made sure that recombination will not involve an offspring individual and its mirrored version at the same time, but either one or the other.

The $(\mu/\mu_W, \lambda_{iid} + \lambda_m)$-ES introduces two different versions of mirroring, namely *random mirroring* and *selective mirroring*. In the case of *random mirroring*, denoted by $(\mu/\mu_W, \lambda_{iid} + \lambda_m^{rand})$-ES, the λ_m offspring subject to mirroring are randomly selected out of the total number of offspring, λ_{iid}. In the case of *selective*

Algorithm 2.18 (1+1)-Active-CMA-ES

```
initialize $\mathbf{x}, \sigma, \mathbf{A} \leftarrow \mathbf{I}, \mathbf{A}^{-1} \leftarrow \mathbf{I}, \mathbf{h} \leftarrow \mathbf{0} \in \mathbb{R}^k$
$t \leftarrow 0$
repeat
  $t \leftarrow t + 1$
  $\mathbf{z} \leftarrow N(\mathbf{0}, \mathbf{I})$
  $\mathbf{y} \leftarrow \mathbf{x} + \sigma \mathbf{A} \mathbf{z}$
  if $t > k$ then
    $h_i \leftarrow h_{i+1}\ \forall i \in \{1, \ldots, k-1\}$
    $h_k \leftarrow f(\mathbf{y})$
  else
    $h_t \leftarrow f(\mathbf{y})$
  end if
  if $f(\mathbf{y}) \leq f(\mathbf{x})$ then
    $\mathbf{x} \leftarrow \mathbf{y}$
    $p_s \leftarrow (1 - c_p) p_s + c_p$
    $\mathbf{s} \leftarrow (1 - c)\mathbf{s} + \sqrt{c(2-c)} \mathbf{A} \mathbf{z}$
    $\mathbf{w} \leftarrow \mathbf{A}^{-1} \mathbf{s}$
    $a \leftarrow \sqrt{1 - c_c^+}$
    $b \leftarrow \frac{\sqrt{1-c_c^+}}{\|\mathbf{w}\|^2} \left( \sqrt{1 + \frac{c_c^+}{1-c_c^+} \|\mathbf{w}\|^2} - 1 \right)$
    $\mathbf{A} \leftarrow a\mathbf{A} + b\,(\mathbf{A}\mathbf{w})\,\mathbf{w}^T$
    $\mathbf{A}^{-1} \leftarrow \frac{1}{a}\mathbf{A}^{-1} - \frac{b}{a^2 + ab + \|\mathbf{w}\|^2} \mathbf{w} \left( \mathbf{w}^T \mathbf{A}^{-1} \right)$
  else
    $p_s \leftarrow (1 - c_p) p_s$
    if $h_0 < f(\mathbf{y})$ then
      $a \leftarrow \sqrt{1 + c_c^-}$
      $b \leftarrow \frac{a}{\|\mathbf{z}\|^2} \left( \sqrt{1 - \frac{c_c^-}{1+c_c^-} \|\mathbf{z}\|^2} - 1 \right)$
      $\mathbf{A} \leftarrow a\mathbf{A} + b\,(\mathbf{A}\mathbf{w})\,\mathbf{w}^T$
      $\mathbf{A}^{-1} \leftarrow \frac{1}{a}\mathbf{A}^{-1} - \frac{b}{a^2 + ab + \|\mathbf{w}\|^2} \mathbf{w} \left( \mathbf{w}^T \mathbf{A}^{-1} \right)$
    end if
  end if
  $\sigma \leftarrow \sigma \exp\left( \frac{1}{d} \frac{p_s - p_t}{1 - p_t} \right)$
until termination criterion satisfied
```

mirroring, denoted by $(\mu/\mu_W, \lambda_{iid} + \lambda_m^{sel})$-ES, the λ_{iid} offspring are first sorted by fitness and the λ_m worst individuals undergo mirroring. This approach is motivated by considering that, in a convex objective function topology, mirroring the best offspring cannot yield any further improvement, such that it will be advantageous to mirror the worst individuals. Moreover, since bad offspring in the case of a (μ_W, λ)-ES are often generated by applying too-large mutation steps, *selective mirroring* itself will also favor large mutation steps [7]. To counteract this undesired bias, the *resample length* approach changes the length of the mirrored mutation step by additionally using a second, newly sampled mutation vector $\mathbf{z}'$. The mirrored version $\mathbf{x}_m$ of the offspring $\mathbf{x} = \langle \mathbf{x} \rangle + \sigma \mathbf{z}$ is then created according to $\mathbf{x}_m = \langle \mathbf{x} \rangle - \sigma \frac{\|\mathbf{z}'\|}{\|\mathbf{z}\|} \mathbf{z}$.

Like for the $(1 \overset{+}{,} \lambda_m^s)$-ES, theoretical results for the convergence velocity on the sphere model have been derived, see [7]. In particular, it has been shown that, for

Algorithm 2.19 $(\mu/\mu_W, \lambda_{iid} + \lambda_m)$-ES

```
initialize $\langle \mathbf{x} \rangle, \sigma$
$r \leftarrow 0$
**repeat**
    $i \leftarrow 0$
    **while** $i < \lambda_{iid}$ **do**
        $r \leftarrow r + 1$
        $i \leftarrow i + 1$
        $\mathbf{x}_i \leftarrow \langle \mathbf{x} \rangle + \sigma N(\mathbf{0}, \mathbf{I})$
    **end while**
    **if** *selective mirroring* **then**
        $\mathbf{x}_1, \ldots, \mathbf{x}_{\lambda_{iid}} = \text{argsort}\left(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{\lambda_{iid}})\right)$
    **end if**
    **while** $i < \lambda_{iid} + \lambda_m$ **do**
        $i \leftarrow i + 1$
        **if** *resample length* **then**
            $r \leftarrow r + 1$
            $\mathbf{x}_i \leftarrow \langle \mathbf{x} \rangle - \frac{\sigma \| N(\mathbf{0},\mathbf{I}) \|}{\| \mathbf{x}_{i-\lambda_m} - \langle \mathbf{x} \rangle \|} \left(\mathbf{x}_{i-\lambda_m} - \langle \mathbf{x} \rangle\right)$
        **else**
            $\mathbf{x}_i \leftarrow \langle \mathbf{x} \rangle - \left(\mathbf{x}_{i-\lambda_m} - \langle \mathbf{x} \rangle\right)$
        **end if**
    **end while**
    $\mathbf{x}_1, \ldots, \mathbf{x}_{\lambda_{iid}} = \text{argsort}(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{\lambda_{iid}-\lambda_m}),$
        $\min\{f(\mathbf{x}_{\lambda_{iid}-\lambda_m+1}), f(\mathbf{x}_{\lambda_{iid}+1})\}, \ldots,$
        $\min\{f(\mathbf{x}_{\lambda_{iid}}), f(\mathbf{x}_{\lambda_{iid}+\lambda_m})\})$
    $\sigma \leftarrow \text{updateStepSize}(\sigma, \mathbf{x}_1, \ldots, \mathbf{x}_{\lambda_{iid}}, \langle \mathbf{x} \rangle)$
    $\langle \mathbf{x} \rangle \leftarrow \langle \mathbf{x} \rangle + \sum_{i=1}^{\mu} w_i (\mathbf{x}_i - \langle \mathbf{x} \rangle)$
**until** termination criterion satisfied
```

the sphere model, maximum convergence velocity is achieved for a setting of $r = \lambda_m/\lambda_{iid} \approx 0.1886$, which can serve as a guideline for the fraction of offspring individuals which should be mirrored.

The pseudocode as given in Algorithm 2.19 is based on using a method *updateStepSize*[26] to update the step size σ, and weights $w_i \; \forall i \in \{1, \ldots, \mu\}$, such that $\sum_{i=1}^{\mu} w_i = 1$.

2.2.2.13 SPO-CMA-ES

The SPO-CMA-ES [70] is essentially a restart-version of the (μ_W, λ)-CMA-ES. It is based on using *sequential parameter optimization* (SPO) [11] to optimize the exogenous parameters of an evolution strategy. SPO uses methods of *design of experiments* (DoE) and *design and analysis of computer experiments* (DACE).[27]

[26] The aforementioned techniques self-adaptation (see Sect. 2.2.1.2) or cumulative step size adaptation (see Sect. 2.2.2.1) are suitable.

[27] See [70] for literature references on these topics as well as the Kriging modeling method.

Concerning the exogenous parameters subject to sequential parameter optimization, the number of offspring individuals[28] $\lambda \in \{\lambda_{def}, \ldots, 1{,}000\}$, the initial step size $\sigma_{init} \in [1, 5]$ and the so-called *selection pressure* $\lambda/\mu \in [1.5, 2.5]$ are identified.

The pseudocode of the SPO-CMA-ES is provided in Algorithm 2.20, and the approach is explained in the following by discussing the various methods used in the algorithm. To begin with, using *latin hypercube sampling* (LHS) [68] an initial design of experiments for the exogenous parameters is created. In the next step (runDesign), independent runs of the (μ_W, λ)-CMA-ES are executed, using the parameter sets of the DoE plan. The results, i.e., the best evaluated individual with its fitness value, of each run is collected in the set Y. This initial phase of the algorithm is called the *exploration phase*.

The next phase, called the *exploitation phase*, is repeated until the predefined budget of function evaluations is reached. Using a function aggregateRuns, a performance measure y is calculated for every run configuration in Y. Based on these performance measure values as outputs and the corresponding parameter sets according to the experimental plan, a Kriging model[29] $\mathcal{M}$ is trained in the method fitModel. This Kriging model $\mathcal{M}$ is then used by the method modelOptimization to identify a new design point, e.g., by running an optimization on the Kriging model and using the resulting point. The new design point d is then added to the experimental plan D, and the loop is executed again. Default settings are not given for the size of the initial experimental plan, N_{init}, nor for the split of the number of function evaluations between the two phases of the algorithm [70]. Rather, the user of the algorithm can fix them, depending on the optimization task at hand. In the case of noisy objective functions, the method runDesign can execute more than the one run, in order to use, e.g., the averages as an estimation of the true fitness value.

2.3 Further Aspects of ES

So far, we have described the ES algorithms as single-criterion optimizers with $\mathbb{R}^n$ as search domain and without handling of constraints. The next three sections give summarized overviews and literature references for further aspects of ES, namely constraint handling, binary and integer search spaces, and multiobjective optimization.

[28] For λ_{def} the standard setting of a (μ_W, λ)-CMA-ES with $\lambda_{def} = 4 + \lfloor 3 \log n \rfloor$ is used.

[29] In principal, any modeling technique can be used to establish the relationship between the exogenous parameters and the performance measure.

Algorithm 2.20 SPO-CMA-ES

```
Input:   box constraints $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$ and size $N_{init}$ of the initial design
Output:  final model $\mathcal{M}$ and best design point $d^*$
  $i \leftarrow 0, D \leftarrow \emptyset$
  $d_i \leftarrow \mathrm{LHS}(\mathbf{l}, \mathbf{u}, N_{init})$
  $Y \leftarrow \mathrm{runDesign}(d_i)$
  $D \leftarrow D \cup d_i$
  while function evaluation budget not exhausted do
    $i \leftarrow i + 1$
    $y \leftarrow \mathrm{aggregateRuns}(Y)$
    $\mathcal{M} \leftarrow \mathrm{fitModel}(D, y)$
    $d_i \leftarrow \mathrm{modelOptimization}(\mathcal{M})$
    $Y \leftarrow Y \cup \mathrm{runDesign}(d_i)$
    $D \leftarrow D \cup d_i$
  end while
  $d^* \leftarrow d_k$ with the best $y_k \in \{y_0, \ldots, y_i\}$
```

2.3.1 Constraint Handling

In Sect. 2.1.1 we defined the optimization problem used throughout this book with equality and inequality constraints as in Eq. 2.2. There are many techniques for handling constraints ranging from simple penalty methods to more complex ones like hybrid methods involving Lagrangian multipliers. Coello gives an overview [18] of constraint-handling techniques to be used with Evolutionary Algorithms but some of these methods may be applied to ES as well. A review by Kramer [42] specializes in constraint-handling methods dedicated to ES and presents the four techniques *penalty methods*, a *multiobjective bioinspired approach*, *coordinate alignment techniques*, and *metamodeling of constraints*.

2.3.2 Beyond Real-Valued Search Spaces

There are many optimization problems where the search domain is not constrained to the real domain. Especially decision problems[30] use categorical search spaces, in most cases binary search spaces, i.e., $\mathbf{x} \in \{0, 1\}^n$, as the simplest categorical search space. Another search space of practical use is the integer search space representable as a subset of $\mathbb{Z}$. Originally, Genetic Algorithms (see [27] or [25] for a comprehensive introduction) were designed to handle binary search spaces, but there are approaches to incorporate those search spaces into ES. In Sect. 2.1.3 we named three guidelines to choose a distribution to be used for mutation. Rudolph [56] introduces a mutation operator for integer search spaces using the difference

[30] For example the NP-hard Traveling Salesman Problem.

of two geometrical distributions. Each discrete variable of a categorical search space is assigned a probability whether to mutate or not. The new value of the discrete variable is drawn uniformly from all possible values. The MI-ES (*mixed-integer* evolution strategies) [43] solve optimization problems which are mixed in their search domain, i.e. the search domain is a composition of real, integer and categorical search spaces. They use the aformentioned mutation approaches together with self-adaptation for the endogenous parameters. An overview of other approaches for handling mixed search spaces is given by Li [43].

2.3.3 Multiobjective Optimization

In single-objective optimization fitness values can be ordered to decide whether one solution is better than another. In multiobjective optimization, where fitness values are represented as vectors, such a strict ordering does not exist anymore. Solutions are partially ordered and based on the partial order solutions can be either *dominated* or *non-dominated* by other solutions. Hence there is not a single optimum to be found but a set of solutions which is called the *Pareto set* or *Pareto front*. For a detailed description of these concepts see [20]. Algorithms for multiobjective optimization have to measure how well a Pareto front is approximated. The most common measures for this task are the *crowding distance* and the *hypervolume contribution*. The former is used for example by NSGA-II [21] the latter by SMS-EMOA [12].

http://www.springer.com/978-3-642-40136-7

Contemporary Evolution Strategies
Bäck, Th.; Foussette, C.; Krause, P.
2013, XIII, 90 p. 33 illus., 31 illus. in color., Hardcover
ISBN: 978-3-642-40136-7