The handle http://hdl.handle.net/1887/19028 holds various files of this Leiden University dissertation.

**Author:** Heinle, Ingo Matthias  
**Title:** Application of evolutionary strategies to industrial forming simulations for the identification and validation of constitutive laws  
**Date:** 2012-05-29
Chapter 2
Optimization

2.1 Introduction

2.1.1 Object variables

A frequently arising problem in the field of engineering is the optimization of the performance of a system. In this thesis the location of the optimum is assumed to be invariant with respect to time (static optimization). The parameters, which are modified in order to obtain the desired performance, are referred to as object variables \( x_i (x_i \in \mathbb{R}) \). It is convenient to assemble these variables into a vector \( x = [x_1, x_2, \ldots, x_n]^T \). Depending on the optimization task, the domain of the object variables may be restricted. Expression (2.1) defines the term feasible region \( M \) [13]:

\[
M := \{ \mathbf{x} \in \mathbb{R}^n | g_i (\mathbf{x}) \leq 0, \forall i \in \{1, \ldots, k\} \}.
\]  

(2.1)

\( k \) is the number of the functions \( g_i \), which are referred to as constraints. Expression (2.2) summarizes the possible states of these functions:

\[
\begin{align*}
g_i (\mathbf{x}) \leq 0 & \Rightarrow \text{satisfied,} \\
g_i (\mathbf{x}) = 0 & \Rightarrow \text{active,} \\
g_i (\mathbf{x}) < 0 & \Rightarrow \text{inactive,} \\
g_i (\mathbf{x}) > 0 & \Rightarrow \text{violated.}
\end{align*}
\]  

(2.2)

If \( M \) is identical to \( \mathbb{R}^n \), the optimization problem is called unconstrained.
CHAPTER 2. OPTIMIZATION

2.1.2 Objective function

For comparing the performance of a system under different choices of the objective variables, a measure is needed, which is termed objective function \( f \). Depending on the problem, an appropriate objective function has to be formulated, which gives a minimum or a maximum value for the best possible performance of the system. Such an objective function enables transforming the optimization task into a search for a minimum or maximum respectively

\[
\min f(x), \quad \max f(x), \quad x \in M, \quad f : M \subseteq \mathbb{R}^n \rightarrow \mathbb{R}. \tag{2.3}
\]

2.1.3 Global, local minimum

The term global minimum of an objective function is defined by [13]

\[
\forall x \in M : f(x^*) \leq f(x), \quad M \neq \emptyset, x^* \in M, \quad f(x^*) > -\infty. \tag{2.4}
\]

\( M \), as introduced above, is the feasible region and \( x^* \) the global minimum. The problem of finding such a global minimum is termed global optimization problem. Any maximization problem can be transformed into a minimization problem, since the identity

\[
\max \{ f(x) | x \in M \} = -\min \{ -f(x) | x \in M \} \tag{2.5}
\]

holds [13]. Hence, without loss of generality only minimization problems are considered in this chapter. A local minimum \( \hat{f}(\hat{x}) \) is defined as

\[
\exists \varepsilon \in \mathbb{R}, \varepsilon > 0 : \forall x \in M : \| x - \hat{x} \| < \varepsilon \Rightarrow \hat{f} \leq f(x). \tag{2.6}
\]

2.1.4 Multicriteria optimization

Optimization problems can comprise multiple objectives. Equation (2.7) shows a minimization problem consisting of \( q \) objectives:

\[
\min f(x) = \left[ \begin{array}{c}
\min f_1(x) \\
\min f_2(x) \\
\vdots \\
\min f_q(x)
\end{array} \right], \quad x \in M, \quad f : M \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^q. \tag{2.7}
\]

The vector \( f \) combines the values of the objectives \( f_i \) with respect to the solution \( x \). If \( x^* \) is the minimum of each \( f_i \), the solution is referred to as ideal.
2.1. INTRODUCTION

Provided that conflicts between the objectives $f_i$ exist (2.8), a unique solution cannot be determined:

$$\min f_i(x) \rightarrow x_i^*, \min f_j(x) \rightarrow x_j^* \land x_j^* \neq x_i^*. \quad (2.8)$$

In this section, an extract of the methods will be introduced for dealing with multi-objective optimization problems.

2.1.5 Pareto optimization

An approach for treating multi-objective problems is the Pareto optimization. A solution $x'$ dominates a solution $x''$ if expression (2.9) applies [14]:

$$x' \prec x'' \iff \forall i \in \{1, \ldots, q\} : f_i(x') \leq f_i(x'') \land \exists j \in \{1, \ldots, q\} : f_j(x') < f_j(x''). \quad (2.9)$$

This definition, which is referred to as the principle of Pareto-Dominance, compares two solutions $x'$ and $x''$ in consideration of each objective of the vector $f$. Consequently, a solution $x'$ is referred to as non-dominated, if no $x''$ exists, which dominates $x'$. The set of non-dominated solutions are termed either Pareto-Set $PS$ (2.10) or efficient set [14]:

$$PS = \{x' \in M | \forall x'' \in M : x'' \prec x'\}. \quad (2.10)$$

This set contains the best solutions of the multi-objective optimization problem. The term Pareto-Front is defined as the mapping of the Pareto-Set into the objective function space. In this thesis, the treatment of constraints in association with the Pareto optimization is omitted. Figure 2.1 illustrates five solutions in the objective function space of a minimization problem consisting of two objectives. Both components of the vectors $f(x_1)$ and $f(x_2)$ are superior to the associated values of $f(x_3)$, $f(x_4)$ and $f(x_5)$. In other words, the solutions $x_1$ and $x_2$ are non-dominated by $x_3$, $x_4$ and $x_5$. The solutions $x_1$ and $x_2$ are incomparable; as neither $x_1$ dominates $x_2$ nor $x_2$ dominates $x_1$.

For the determination of the Pareto-Front diverse solutions have to be identified, which differ sufficiently from each other. As each of these solutions is a compromise, finally one has to make a choice on the basis of additional, a posteriori defined, experience based preferences, which are difficult to formulate as an objective function. A procedure for finding a Pareto-Front is the weighted Tchebycheff method. First of all, for each objective $f_i$, a single-objective optimization is performed

$$f_i^* = \min_{x \in M} f_i(x). \quad (2.11)$$
As a next step, several vectors \( \mathbf{w} \) are chosen and for each vector the optimization, defined by expression (2.12), is performed:

\[
\sum \omega_i = 1, \quad \min_{\mathbf{x} \in M} \max_{i=1, \ldots, q} \omega_i |f_i(\mathbf{x}) - f_i^*| .
\]

(2.12)

The Tchebycheff method minimizes the maximum deviation between each objective \( f_i \) and the related result of the single-objective optimization \( f_i^* \). Expression (2.12) aggregates the results of the multiple objectives to a scalar value. Each of these optimizations gives a point of the Pareto-Front [15]. The Tchebycheff method allows to compute all points on the Pareto-Front by changing the weights \( \omega_i \) [16]. A downside of this method is the computational cost for determining the Pareto-Front. First of all, \( q \) single-objective optimizations are necessary in order to identify all \( f_i^* \). Moreover for each point on the Pareto-Front an additional single-objective optimization has to be performed.

Evolutionary algorithms, which are introduced in chapter 2.3, are especially suitable for the determination of a Pareto-Set. This type of optimization algorithm deals simultaneously with different solutions. Hence, in combination with some extensions, Evolutionary Algorithms are able to identify a Pareto-Set within a single run. As an example the fundamental idea of the Strength Pareto Evolutionary Algorithm 2 (SPEA2), which was published by Zitzler in 2001, is given in this section [17]. Further algorithms can be found in [18] and [19]. The applied Evolutionary Algorithm is complemented by an archive \( P \), which includes a prescribed number of \( N \) solutions, containing the so far detected promising solutions. At the beginning of the optimization, the archive is an empty set and \( t \) is equal to 0. Subsequently, the optimization step \( t + 1 \) is discussed. A raw fitness value, as defined by

\[
R(i) = \sum_{j \in P_t \cup P_0, j \prec i} S(j), \quad S(i) = \left| \{j | j \in P_t \cup P_0 \land i \prec j \} \right|,
\]

(2.13)

is assigned for each solution of the current population \( P_t \) and the members
of the archive $P_t$. In this section, $|A|$ gives the number of the elements of a set $A$. A high value of the raw fitness $R(i)$ indicates that the associated solution is dominated by many other ones. If the raw fitness $R(i)$ is equal to 0.0, the solution is non-dominated. For maximizing the diversity of the generated solutions, a density is introduced, which is assigned to each solution. Mathematically, the density is defined as given by

$$D(i) = \frac{1}{\sigma^k_i + 2}. \quad (2.14)$$

The formulation of (2.14) implies that $D(i) < 1$. For the determination of $\sigma^k_i$, the distance in the objective function space of the solution $i$ to all the solutions of the population and the archive is computed and sorted in an increasing order. $\sigma^k_i$ denotes the distance of the solution $i$ to its $k$-th nearest neighbor. Zitzler recommends choosing $k$ as given by

$$k = \sqrt{N + \overline{N}}. \quad (2.15)$$

($N$: Population size; $\overline{N}$: Archive size). Consequently, an accumulation of solutions in a region of the Pareto-Front is identified by high values of the associated density. In SPEA2, both, the raw fitness and the density are additively combined to a scalar fitness value

$$F(i) = R(i) + D(i). \quad (2.16)$$

Equation (2.17) illustrates the update of the archive:

$$P_{t+1} = \{ i | i \in P_t \cup \overline{P}_t \land F(i) < 1 \}. \quad (2.17)$$

According to (2.17) all non-dominated solutions are transferred to the archive. If the number of known non-dominated solutions is smaller than the prescribed archive size, dominated solutions are copied to the archive. In this case, these dominated solutions are taken, implying the best fitness values (2.16). If the number of non-dominated solutions exceeds the archive size, solutions are iteratively sorted out. Thereby, a solution $i$ is discarded, if $i \leq_d j$ for all $j$ of $P_{t+1}$:

$$i \leq_d j \Leftrightarrow \forall \quad 0 < k < |P_{t+1}| : \sigma^k_i = \sigma^k_j \lor \exists \quad 0 < k < |P_{t+1}| : \left[ \left( \forall \ 0 < l < k : \sigma^l_i = \sigma^l_j \right) \land \sigma^k_i < \sigma^k_j \right]. \quad (2.18)$$

Alternatively stated, the solution, which is located closest to another solution, is discarded from the archive. Additionally, equation (2.18) considers the distance to the second closest solution and so forth. The selection within the EA is performed on the basis of the fitness function (2.16). It has to be mentioned that additional modifications with respect to the EA-Algorithm are necessary in order to obtain the desired search performance. For example, the strategy parameters
of an ES-Algorithm (evolutionary strategy), which will be introduced in the next chapter, should be individually adapted for each member of the population [20]. The individual adaptation of the strategy parameters is necessary, as diverse solutions are desired and the associated optimal strategy parameters may be different. If the stopping criterion for the optimization is satisfied, the set \( P_t \) contains the non-dominated solutions.

2.1.6 Scalarization

Another technique for treating multicriteria optimization problems are the scalarization methods [16]. The idea of scalarization is to transform multiple objectives to a single one by applying an aggregation of the objectives. Two of these methods are shown below.

Additive weighted aggregation

Firstly, the well known additive aggregation of the objectives is discussed. For each objective a weight \( w_i \) is introduced, which has to be defined a priori. As shown by (2.19) the weighted objectives are added in order to generate a single-objective function \( F(x) \) [21]:

\[
F(x) = \sum_{i=1}^{q} \omega_i f_i(x).
\] (2.19)

A drawback of this method is that the result of the optimization depends on the choice of the weights. Additionally, a poor performance induced by the choice of the object variables with respect to a single objective in combination with a small weight may not be reflected by the result of the objective function \( F(x) \). Hence, for real-live problems the selection of the weights might be a difficult task.

From expression (2.20) follows, that a point, which is non-dominated with respect to (2.9) is also non-dominated with respect to (2.19) [16] \((x' \prec x'' \iff f'(x') \prec f''(x''))\):

\[
\forall f', f'' \in \mathbb{R}^q : f' \prec f'' \rightarrow \sum_{i=1}^{q} f'_i < \sum_{i=1}^{q} f''_i. \tag{2.20}
\]

Consequently, also the question arises, whether this type of aggregation enables to determine all points on the Pareto-Front. For introducing theorem 1, which answers this question, the definitions 1, 2 and 3 are necessary.

**Definition 1** A subset \( C \subseteq \mathbb{R}^q \) is called a cone, if \( \alpha d \in C \) for all \( d \in C \) and for all \( \alpha \in \mathbb{R}, \alpha > 0 \) [21].

**Definition 2** A cone \( C \) in \( \mathbb{R}^q \) is called convex, if \( \alpha d_1 + (1 - \alpha) d_2 \in C \) for all \( d_1, d_2 \in C \) and for all \( 0 < \alpha < 1 \) [21].
Definition 3 Given a Pareto optimization problem (Expression 2.7), then a solution \( \mathbf{x} \) is called efficient in the Geoffrion sense or properly efficient, if (a) it is efficient, and (b) there exists a number \( N > 0 \) such that \( \forall i = 1, \ldots, q \) and \( \forall \mathbf{x} \in M \) satisfying \( f_i(\mathbf{x}) < f_i(\mathbf{x}') \), there exists an index \( j \) such that \( f_j(\mathbf{x}) < f_j(\mathbf{x}') \) and \( f_i(\mathbf{x}') - f_i(\mathbf{x}) \leq N \) [16].

Theorem 1 Let us assume a Pareto optimization problem (Expression 2.7) with a Pareto-Front that is cone convex with respect to the positive orthant \( \mathbb{R}_+^q \). Then for each properly efficient point \( \mathbf{x} \in M \) there exist weights \( \omega_1 > 0, \ldots, \omega_q > 0 \) such that \( \mathbf{x} \) is one of the solutions of \( \sum_{i=1}^{q} \omega_i f_i(\mathbf{x}) \rightarrow \min \) [16]. Therefore, by variation of the weights \( \omega_i \), all points on the Pareto-Front can be obtained, if the Pareto-Front is cone convex.

Desirability functions

Another method to transform a multi-objective optimization problem into a single-objective one is to aggregate the objectives multiplicatively in combination with Harrington desirability functions. Expression (2.21) illustrates the one-sided and (2.22) the two-sided Harrington desirability function [22]:

\[
d_1(f(\mathbf{x}), b_0, b_1) = e^{-e^{(-b_0 + b_1 f(\mathbf{x}))}},
\]

\[
d_2(f(\mathbf{x}), U, L, n) = e^{-\left(2f(\mathbf{x})-(U+L)\right)^{n}}.
\]

The expressions (2.21) and (2.22) map the objective function \( f \) to the interval \([-1, 1]\) (figure 2.2). If the objective function is equal to the desired value, the corresponding value of the desirability function is 1.0. For a non-satisfying result of the objective function \( f \), the desirability function gives values close to 0.0. The mapping according to (2.21) and (2.22) implies a gray-zone, which is determined by the parameters \( n \), \( U \), \( L \), \( b_0 \) and \( b_1 \). The gray-zone enables the user to weight deviations of the objective function from the desired value.

This approach is especially suitable, if the corresponding values \( f \) of the optimum and the non-acceptable values of \( f \) are a priori known. Figure 2.3 shows three different choices of the parameter \( n \). According to the example, deviations from the optimum are stronger penalized, if \( n \) is chosen to be small.

Nevertheless, the choice of the exponent is arbitrary and has to be performed based on experience. Additionally, the desirability function can be applied to model constraints. A violation of the constraint corresponds to a value close to 0.0. If the constraint is fulfilled, the desirability function is equal to 1.0. Finally, the objectives and the constraints can be multiplicatively aggregated.
Figure 2.2: Left: Visualization of a one-sided Harrington desirability function; Right: Visualization of a two-sided Harrington desirability function.

Figure 2.3: Effect of $n$ on the shape of the two-sided Harrington desirability function.
2.2. ONE DIMENSIONAL STRATEGIES

\[ d_A(x) = \prod_{i=1}^{r} d_1(f_i(x), b_{0i}, b_{1i}) \prod_{j=1}^{s} d_2(f_j(x), U_j, L_j, n_j) \]
\[ \prod_{k=1}^{t} d_1(g_k(x), b_{0k}, b_{1k}) \prod_{l=1}^{u} d_2(g_l(x), U_l, L_l, n_l). \]

(2.23)

This type of aggregation gives pessimistic results, as shown by

\[ d_A(x) \leq \min_{i=1,...,r;j=1,...,s;k=1,...,t;l=1,...,u} (d_1(f_i(x), b_{0i}, b_{1i}) ; d_2(f_j(x), U_j, L_j, n_j) ; d_1(g_k(x), b_{0k}, b_{1k}) ; d_2(g_l(x), U_l, L_l, n_l)). \]

(2.24)

Expression (2.25) shows the transformation of (2.23) into a minimization problem:

\[ F(x) = 1 - d_A(x). \]

(2.25)

A similar type of desirability functions can be found in [23].

2.2 One dimensional strategies

The discussion of one dimensional strategies is limited to functions \( f(x) : \mathbb{R} \rightarrow \mathbb{R} \), which are unimodal in \( x \in [a, b] \subseteq \mathbb{R} \), and the desired minimum is located within this interval.

2.2.1 Simultaneous optimization

A possible solution is to create a set of points \( P = \{ x_i \in [a; b] \}, i \in \{1, ..., n\} \) and to compute the corresponding objective function values \( f(x_i) \). The point \( \pi = \min_{x_i \in P} (f(x_i)) \) is assumed to be the minimum of \( f(x) \). It can be theoretically shown that an equidistant distribution of the search points is the best choice [24]. The interval of uncertainty regarding the true minimum \( x^* \) is defined by \( x^* \in [c; d] \subseteq [a; b] \). For a given number \( n \) of uniformly distributed points, the size of the interval of uncertainty is given by expression

\[ u_n = \frac{2(b - a)}{n + 1} < \epsilon \]

(2.26)
Thereby, \( n \) refers to the amount of points inside of the interval \([a; b] \). Expression (2.27) shows the determination of the equidistant distributed points \( x_i \), needed for seeking the optimum:

\[
x_i = a + \frac{(b - a)}{n + 1} i, \quad i = 1, \ldots, n.
\] (2.27)

The interval of uncertainty is given with respect to smallest computed objective function value by \([\tau - \frac{u_n}{2}; \tau + \frac{u_n}{2}]\). An estimation of an appropriate choice of \( n \) for a given accuracy \( \epsilon \) is shown by expression

\[
\frac{2(b - a)}{\epsilon} - 1 < n.
\] (2.28)

An advantage of this method is the possibility of computing the objective function values simultaneously. However, depending on the needed accuracy \( \epsilon \) the method can be computationally expensive. In order to reduce the needed amount of objective function evaluations, it is advantageous to adapt the search according to the information, available during the optimization, regarding the location of the optimum. Therefore, sequential methods are introduced subsequently, which possess this property.

### 2.2.2 Sequential methods

The above shown equidistant minimum search method can be applied multiple times. Each time the search is repeated, the original interval is reduced by a factor \( \alpha \). Consequently, after \( k \) repetitions, the interval shrinks by the factor \( \alpha^k \). The factor \( \alpha \) depends on the amount of objective function evaluations per repetition \( \alpha = \frac{2}{n + 1} \). It is more advantageous to reduce \( n \) and to perform more repetitions than vice versa [24], as the shrinking of the investigated interval depends exponentially on \( k \). The minimum number of objective function evaluations per step (repetition) is \( n = 2 \). As the reduced interval is defined by the lowest objective function value \( \tau^{(k)} \) with respect to the actual step \( k \) and both neighbor points, the value of \( n \) cannot be smaller than \( n = 2 \). If \( n \) is chosen to be equal to three, only in the first step three computations of the objective function are necessary. For the following steps, \( f(x) \) needs to be evaluated only two times, as one of the points \( x_i^{(k)} \) is coincident with a point \( x_i^{(k-1)} \) of the previous step.

By giving up the constraint of using equidistant points, it is possible to enforce on the basis of two points (\( n = 2 \)) that each step comprises a point of the previous one. Consequently, only one evaluation of the objective function is needed for each step, which leads to a better performance compared with the equidistant interval division methods.

Figure 2.4, taken from [25], shows the interval division for the steps \( k - 1 \) and \( k \). \( \tau \) defines the relative position of the points \( x_1^{(k)}, x_2^{(k)} \) with respect to the interval \([a^{(k)}; b^{(k)}]\). The equation (2.29), which leads to coincident points between the steps, can be directly derived from this figure:
2.2. ONE DIMENSIONAL STRATEGIES

\[ a^{(k-1)} \quad x_1^{(k-1)} \quad x_2^{(k-1)} \quad b^{(k-1)} \]

\[ (b^{(k-1)}-a^{(k-1)})_1 \tau \]

\[ (b^{(k-1)}-a^{(k-1)})_2 \]

\[ \tau = \frac{\sqrt{5} - 1}{2} \approx 0.618. \]

Consequently, using this value of \( \tau \) leads to an interval division, which comprises only one evaluation of \( f(x) \) per step. This procedure is referred to as golden section interval division.

2.2.3 Interpolation methods

Newton method

A root of the function \( f(x) \) can be computed by applying the Newton method, provided the function is differentiable. The method estimates the root starting from a linearization of \( f(x) \) at \( x^{(k)} \). The root of the linear function \( x^{(k+1)} \) is taken as an improved value for a further iteration. Expression (2.30) summarizes both steps and gives \( x^{(k+1)} \):

\[ x^{(k+1)} = x^{(k)} - f(x^{(k)}) \left[ \frac{df(x)}{dx} \right]_{x=x^{(k)}}^{-1}. \]  

(2.30)

If the function \( f(x) \) is linear, the Newton method gives the exact solution within one iteration.
Regula Falsi iteration

Expression (2.31) illustrates an approximation of the derivative at the point \( x^{(k)} \) [26]:

\[
\frac{df(x)}{dx}\bigg|_{x=x^{(k)}} \approx \frac{f(x^{(k)}) - f(x^{(m)})}{x^{(k)} - x^{(m)}}, \quad m < k. \tag{2.31}
\]

An exchange of the derivative of (2.30) by expression (2.31) leads to the Regula Falsi iteration

\[
x^{(k+1)} = x^{(k)} - f(x^{(k)}) \left( \frac{f(x^{(k)}) - f(x^{(m)})}{x^{(k)} - x^{(m)}} \right)^{-1}. \tag{2.32}
\]

A root of a continuous function \( f(x) \) within the interval \([a; b]\) exists, if \( f(a) \) and \( f(b) \) are of opposite sign. Therefore, the sign of the corresponding function values of the initial values \( x^{(0)} \) and \( x^{(1)} \) has to be different.

Provided \( f(x) \) is linear, expression (2.31) gives the exact derivative and therefore also (2.32) leads to the exact solution by one iteration. For a stable convergence of the Regula Falsi iteration, \( f(x^{(k)}) \) and \( f(x^{(m)}) \) have to be of opposite sign [26], which is achieved by selecting an appropriate value of \( m \).

Lagrangian interpolation

Finally, the Lagrangian interpolation is introduced. For searching a minimum, the function \( f(x) \) is interpolated by a pth order polynomial, which is fitted on the basis of \( p + 1 \) evaluations of \( f(x) \). The optimum is derived from the stationary point of the interpolated polynomial. Expression (2.33) shows the computation of the stationary point of a second order polynomial [24]:

\[
A^{(k)} = \left[ (b^{(k)})^2 - (c^{(k)})^2 \right] f(a^{(k)})
\]

\[
B^{(k)} = \left[ (c^{(k)})^2 - (a^{(k)})^2 \right] f(b^{(k)})
\]

\[
C^{(k)} = \left[ (a^{(k)})^2 - (b^{(k)})^2 \right] f(c^{(k)})
\]

\[
d^{(k)} = \frac{1}{2} \frac{A^{(k)} + B^{(k)} + C^{(k)}}{b^{(k)} - c^{(k)}} \left[ f(a^{(k)}) + \frac{b^{(k)} - a^{(k)}}{f(b^{(k)})} ] f(b^{(k)}) + [a^{(k)} - b^{(k)}] f(c^{(k)}) \right]. \tag{2.33}
\]

The stationary value is a minimum, if the denominator is positive. In order to improve the computed minimum, the procedure can be repeated. It is recommended to choose the initial value \( b^{(0)} \) in the middle of the interval \([a^{(0)}; c^{(0)}]\) [24]. Each further iteration needs only one additional objective function evaluation, as
shown by table 2.1. If the function $f(x)$ is a quadratic function, the Lagrange interpolation gives the exact minimum within one iteration.

Table 2.1: Recursion algorithm for the Lagrange interpolation.

$$
\begin{align*}
&\text{if } a^{(k)} < d^{(k)} < b^{(k)} \land f(d^{(k)}) < f(b^{(k)}) \quad \begin{cases} a^{(k+1)} = a^{(k)} \\ b^{(k+1)} = d^{(k)} \\ c^{(k+1)} = b^{(k)} \end{cases} \\
&\text{if } a^{(k)} < d^{(k)} < b^{(k)} \land f(d^{(k)}) > f(b^{(k)}) \quad \begin{cases} a^{(k+1)} = d^{(k)} \\ b^{(k+1)} = b^{(k)} \\ c^{(k+1)} = c^{(k)} \end{cases} \\
&\text{if } b^{(k)} < d^{(k)} < c^{(k)} \land f(d^{(k)}) < f(b^{(k)}) \quad \begin{cases} a^{(k+1)} = b^{(k)} \\ b^{(k+1)} = d^{(k)} \\ c^{(k+1)} = c^{(k)} \end{cases} \\
&\text{if } b^{(k)} < d^{(k)} < c^{(k)} \land f(d^{(k)}) > f(b^{(k)}) \quad \begin{cases} a^{(k+1)} = a^{(k)} \\ b^{(k+1)} = b^{(k)} \\ c^{(k+1)} = d^{(k)} \end{cases}
\end{align*}
$$

2.3 **Evolutionary algorithms**

Evolutionary algorithms are methods, inspired by the natural evolution, for solving optimization problems. According to the Darwinian theory of evolution, the adaptive change of species is explained by the principle of natural selection. Species are favored, which are best adapted to their environmental conditions. Furthermore, Darwin recognized that offspring are subject to mutation, in which the occurrence of small, apparently random and undirected variations of the genetic material occurs. The copying process of the genetic information between the parents and the offspring works highly accurately, however not perfect [13]. These slight imperfections are equivalent to the mutation. The fitness in the sense of evolution of an individual is measured indirectly by its ability to survive and to reproduce in its environment. Mutation enables genetic changes, which optimize in combination with selection the fitness. In nature, selection is based on the fitness of the individual. The fitter the individual, the higher is the chance to bring its genetic information to the next generation (“survival of the fittest”).

The objective of natural evolution is to find continuously an appropriate adaption of the species with respect to the environmental conditions. In literature, recombination, which describes the combination of the parental genetic material, is not
considered as an essential contribution to the evolution, because this process does not induce any new information.

As mentioned above, evolutionary algorithms (EA), which are applied for optimizing fitness functions, imitate natural evolution. In the field of evolutionary algorithms, the term fitness function is usually applied, which is equivalent to the above introduced objective function. For the solution of the numerical optimization problem, the objective function is interpreted as a so-called adaptive landscape in the context of evolutionary algorithms [13]. During the optimization, the development of the initial population subject to genetic operators (selection, mutation and recombination) is simulated. In contrast to natural evolution, the fitness of individuals can be directly quantified. The objective is to find a choice of the object variables, which minimizes the fitness function. The term evolutionary algorithm summarizes three main streams of algorithms. The origins of two of them are in the United States of America. Holland (1965; 1975) has developed the genetic algorithms (GA) and Fogel (1962; 1966) published another variant of EA called evolutionary programming (EP). Finally, evolutionary strategies (ES) were developed by students at the Technical University of Berlin (Rechenberg, 1965; 1971; Schwefel, 1965, 1975) [27]. Subsequently, only evolutionary strategies are discussed, because these are the most powerful variants of evolutionary algorithms for real-valued and mixed-integer variables.

2.4 Evolutionary strategies

Typically, evolutionary strategies (ES) are applied for optimizations of non-linear fitness functions. During the optimization procedure, trial choices of the object variables, called individuals, are created. The set of individuals is referred to as a population. The term generation is introduced for distinguishing between different populations. Usually, the initial population is generated randomly. As a next step, the fitness of the individuals of the first population is computed. In order to obtain a progress, only the best individuals with respect to the fitness are selected for creating the subsequent population. The optimization enters at this state into a loop. The termination condition of this loop is either a resource (maximum number of generations, maximum CPU time) or a convergence (in the space of fitness values, object variables, strategy parameters) criterion. The offspring of the selected individuals is generated by recombination and mutation operators. The first mentioned operator combines the parental genetic information. The mutation operator generates, on the basis of the result of recombination, stochastic variations, which have to comply with the boundary conditions of the optimization problem. The occurrence of small mutations should be more likely. This operator introduces alterations regarding the object variables into the optimization process, which is essential for finding an optimum. Again, the fitness of the members of the new generation is evaluated and only those individuals are selected, which show promising properties. The loop is repeated until the termination condition is fulfilled. Figure 2.5 summarizes the general form of an evolutionary strategy.
2.4. EVOLUTIONARY STRATEGIES

ES do not use any information regarding the structure of the optimization problem. Only the associated fitness value of a given choice of the objective variables is needed. Hence, from the perspective of the optimizer, the fitness function acts as a black box. Because of this property, the ES belong to the zero-order optimization algorithms. A first-order optimization algorithm also takes the first derivative of objective function into account and a second-order one the second derivative respectively.

The evaluation of the fitness values of each individual, belonging to a population, can be performed independently. If the determination of the fitness of individuals is time consuming, this property of evolutionary strategies offers the potential to perform the evaluation of the fitness function simultaneously, i.e., to exploit parallelism.

The standard notation \((\mu/\rho+\lambda)\)-ES describes the selection strategy (plus, comma strategy), the number of parents involved in the creation of one offspring \(\rho\), the amount of the created offspring \(\lambda\) and the number of parents \(\mu\). For the special case \(\rho = 1\), the offspring is generated without any recombination (cloning). For the remaining cases \(\rho > 1\), the offspring is procreated on the basis of more than one parent. \(\lambda, \mu\) and \(\rho\) are referred to as exogenous parameters, as these quantities are defined in advance and are kept constant during the optimization.

2.4.1 Selection

After creating \(\lambda\) descendants and computing the related fitness values, the comma strategy chooses the best \(\mu\) individuals of the offspring for the new generation. The plus strategy selects the best \(\mu\) individuals out of the set, consisting of parents and offspring. Hence, the latter mentioned strategy enables individuals to survive several generations. Consequently, in both cases the population size remains constant. Furthermore, the presented selection schemes are strictly deterministic [29]. The comma strategy even discards good solutions, if all offspring is worse than the parents. Because of this behavior the scheme is able to overcome local minima. This property is especially for multimodal landscapes advantageous [13]. Additionally, in contrast to the plus strategy, the comma strategy is suitable for

---

```
1  t = 0    P : Population
2  P_t ← Init ()
3  Evaluate (P_t)
4  while (termination criterion is not fulfilled) do
5      G_t ← Generate (P_t)
6          Evaluate (G_t)
7      P_{t+1} ← Select (G_t ∪ P_t)
8      t ← t + 1
9  end while
```

Figure 2.5: A general scheme of evolutionary strategies [28].
CHAPTER 2. OPTIMIZATION

the application to changing environments. The plus strategy preserves solutions and is therefore not able to follow a moving optimum. A drawback of the comma scheme is its convergence behavior. In the worst case, the comma strategy can even diverge, whereas the plus strategy shows only a premature stagnation [27]. Nevertheless, the comma selection mechanism is usually applied in ES [29], due to the fact that it is advantageous for the self-adaptation of the strategy parameters.

2.4.2 Recombination

Inspired by nature, the recombination combines parental genetic information for the creation of individuals. The term multirecombination is applied for the case, if more than two parents are involved in the creation of offspring $\rho > 2$ [27]. The recombination operator is only used for populations, which consist of at least two or more individuals $\mu > 1$. Generally, this operator causes in ES the creation of a single offspring. The recombination is performed on the basis of $\rho$ parental vectors $a$. The recombination is termed discrete, if each component of the resultant vector $r$ is obtained by a random selection from the corresponding components of the parental vectors $a$. Expression (2.34) illustrates the introduced procedure [27] ($n$ dimension of the parameter space):

$$ (r)_k := (a_{m_k})_k, \text{ with } m_k = \text{random} \{1, \ldots, \rho\}, \ k \in [1;n]. \quad (2.34) $$

Another approach is the intermediate recombination, which takes all $\rho$ parents into account. The computation of the resultant vector $r$ is performed in analogy with the center of mass

$$ (r)_k := \frac{1}{\rho} \sum_{i=1}^{\rho} (a_i)_k. \quad (2.35) $$

2.4.3 Mutation

The selection guides the optimization into promising search space regions, whereas mutation is needed for exploring the search space. Generally, the variation of the parental state should not depend on any fitness information. The performance of the mutation operator depends on the optimization problem. A general method for designing such an operator has not been established yet. Subsequently, the recommendations of Beyer [27] are given, which have been derived from theoretical considerations and analyses of ES-implementations.

The first principle requires the reachability of an arbitrary point in the search space, independent of the position of the parental point, within a finite number of mutations. Another principle is referred to as scalability, which formulates the need for an adaptation of the mutation operator with respect to the properties of the fitness landscape. Finally, the variation of the parental state should not introduce any bias (principle of unbiasedness). A violation of the presented design
2.4. EVOLUTIONARY STRATEGIES

principles does not necessarily indicate that the considered operator will generally fail.

A search space can consist of combinations of real-valued, integer-valued or binary object variables. Subsequently, a pure real-valued search space is assumed. A discussion of the remaining types of object variables is omitted in this work. A comprehensive introduction regarding the treatment of these types of variables can be found in [30]. Commonly, the standard normal distribution plays a crucial role in formulating mutation operators

\[ N(\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(-\frac{1}{2\sigma^2} (x - \mu)^2\right). \]  

(2.36)

The offspring \( x^{(g+1)} \) is created by adding a mutation to a parent \( x^{(g)} \)

\[ x^{(g+1)} = x^{(g)} + z. \]  

(2.37)

Equation (2.38) shows a possible mutation operator, consisting of standard normal distributions (2.36) for each dimension \( z \sim \sigma (N_1 (0, 1), N_2 (0, 1), \ldots, N_n (0, 1))^T \).

\[ z \sim \sigma (N_1 (0, 1), N_2 (0, 1), \ldots, N_n (0, 1))^T. \]  

(2.38)

The symbol \( \sim \) denotes the equality in distribution. According to the scalability principle, the parameter \( \sigma \) enables the adaptation of (2.38) with respect to the fitness landscape. Strategy parameters, like \( \sigma \), which are continuously changed during the optimization, are referred to as endogenous. Procedures for adapting the endogenous parameters are shown in the next section.

Each component of the random vector \( z \) (2.38) is drawn independently from the normal distribution with a zero mean and a variance of one. In this case, the surfaces of equal probability density are concentric spheres in the search space (figure 2.6 (left)). The midpoint of these spheres coincides with the parental state \( x^{(g)} \). Expression (2.38) is also termed isotropic mutation operator. For a \( n \)-dimensional vector \( x \), the multivariate Gaussian distribution takes the form

\[ N(\mathbf{m}, \mathbf{C}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\mathbf{C}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (x - \mathbf{m})^T \mathbf{C}^{-1} (x - \mathbf{m})\right). \]  

(2.39)

The eigenvalues of the covariance matrix \( \mathbf{C} \) must be strictly positive, otherwise the distribution cannot be normalized properly [31]. A covariance matrix, implying this property, is said to be positive definite. Additionally, the covariance matrix has to be symmetric [32]. Therefore, surfaces of constant densities are hyper ellipsoids, which are aligned along the principal axes of the covariance matrix. The center of these hyper ellipsoids is coincident with \( \mathbf{m} \). As the eigenvectors, obtained from solving the standard eigenvalue problem of a symmetric matrix, are orthogonal [26], the principal axes system of the covariance matrix \( \mathbf{C} \) is also orthogonal. Generally, the covariance matrix \( \mathbf{C} \) can be decomposed into a diagonal matrix \( \mathbf{D}^2 \), containing the eigenvalues, and an orthogonal matrix \( \mathbf{B} \):
\[
\begin{align*}
C &= BD^2B^T, \quad BB^T = I, \\
D^2 &= \text{diag}(d_1, d_2, \ldots, d_n)^2 = \text{diag}(d_1^2, d_2^2, \ldots, d_n^2).
\end{align*}
\]

The expressions (2.40) are only valid for a positive definite and symmetric matrix. The matrix \( D \), containing \( n \) independent components, scales the spherical distribution with respect to the principal axes, whereas the matrix \( B \) defines the rotation between the coordinate system of the search space and the principal axes. The latter mentioned matrix implies \( n(n-1)/2 \) degrees of freedom [33]. For generating a Gaussian normal distributed random vector \( N (m, C) \), the following expression can be applied

\[
N (m, C) \sim m + BDN (0, I).
\]

(2.41)

According to (2.41), the desired random vector can be obtained by realizing \((0, 1)\)-normally distributed numbers, which can be easily realized on a computer. Expression (2.42) shows the relation between the distribution of the vector \( z \) and the distribution of its components \( z_i \)

\[
p (z) = \prod_{i=1}^{n} p_i (z_i).
\]

(2.42)

On the basis of (2.38) and (2.39) an alternative formulation of (2.38) can be derived:

\[
z \sim N (0, C_1 (\sigma)), \quad C_1 (\sigma) = \sigma^2 I,
\]

(2.43)

\[
D^2_1 = C_1, \quad B = I.
\]

(2.44)

Expression (2.45) illustrates an extended version of (2.38), which introduces for each dimension an endogenous parameter \( \sigma_i \):

\[
z \sim (\sigma_1 N_1 (0, 1), \sigma_2 N_2 (0, 1), \ldots, \sigma_n N_n (0, 1))^T.
\]

(2.45)

As a consequence, the surfaces of equal probability density are in this case ellipsoidal. This approach enables a better adaptation of the mutation operator with respect to the fitness landscape. Provided, the endogenous strategy parameters \( \sigma_i \) are well determined, the probability density reflects promising directions. Hence, the likelihood of generating successful mutations is higher, than it would be if an isotropic mutation operator were applied. Expression (2.46) shows an alternative form of (2.45):
2.4. EVOLUTIONARY STRATEGIES

\[ z \sim N(0, C_2(\sigma_i)), \quad C_2(\sigma_i) = \begin{pmatrix} \sigma_1^2 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_n^2 \end{pmatrix} \]  
\hspace{1cm} (2.46)

As \( C_2 \) is a diagonal matrix, the eigenvectors of \( C_2 \) are orthogonal and identical to the coordinate system of the search space. Therefore, the ellipsoidal surfaces are orientated along the axes of the search space (figure 2.6 (middle)). Expression (2.47) shows the eigendecomposition of \( C_2 \):

\[ D_2^2 = C_2, \quad B = I. \]  
\hspace{1cm} (2.47)

Figure 2.6: Equal density of different normal distributions.

An improved adaptation of the mutation operator with respect to the fitness landscape can be obtained by introducing orthogonal rotations of the eigenvectors (figure 2.6 (right)). This procedure assures, that the covariance matrix remains positive definite. Expression (2.48) and (2.49) define the rotation matrix \( B \) [13]:

\[ B(\alpha) = \begin{pmatrix} \prod_{i=1}^{n-1} \prod_{j=i+1}^{n} R(\alpha_{ij}) \end{pmatrix}, \]  
\hspace{1cm} (2.48)

\[ R(\alpha_{ij}) = (r_{kl}), \quad r_{ii} = r_{jj} = \cos \alpha_{ij}, \quad r_{ij} = -r_{ji} = -\sin \alpha_{ij}. \]  
\hspace{1cm} (2.49)

The endogenous strategy parameters \( \alpha_{ij} \) define the angles of the rotation with respect to the axes \( i \) and \( j \). In the most general case, \( n(n-1)/2 \) rotation angles are applied. Consequently, all degrees of freedom of the orthogonal matrix \( B \) are defined by a rotation angle. The expressions (2.50) illustrate the mutation operator, which comprises a scaling of each diagonal element and a rotation:

\[ z \sim N(0, C_3(\sigma_i, \alpha_{ij})), \quad C_3 = B C_2 B^T, \quad C_3 = B D_2^2 B^T. \]  
\hspace{1cm} (2.50)
2.4.4 Adaptation of endogenous strategy parameters

The adaptation of the endogenous strategy parameter plays a crucial role in evolutionary strategies. Generally, without changing the strategy parameters a poor performance is expected [34]. On the basis of the strategy parameters, the size of mutation steps (step-size) in the search space is varied. At the beginning of this section, the original 1/5th-rule is introduced, which is applied for the adaption of an isotropic Gaussian mutation operator (2.38) in (1 + 1)-ES. The performance of such an ES strongly depends on the choice of the mutation strength \( \sigma \) [27]. According to (2.38), the expected length of the mutation with respect to the search space is proportional to \( \sigma \). The smaller the mutation strength \( \sigma \) is, the smaller the expected length of the mutation step in the search space is. The term success probability is introduced for describing the likelihood that the offspring replaces the parent. The success probability becomes approximately 50%, if the mutation strength is assumed to be very small (i.e., asymptotically approaching zero) and the fitness landscape is smooth. On the one hand, such a mutation strength leads to a remarkable evolvability. On the other hand, the expected length of the mutation steps is small and therefore the number of necessary optimization steps for finding the optimum is unacceptable high. In case the mutation strength is very large, the success probability reduces and, consequently, a progress is not obtained anymore. Thus, the success probability is very low. Again, such a selection of the mutation strength would lead to an unsatisfying search performance. The term evolutionary window, introduced by Rechenberg (1973), describes a range of mutation strengths between the discussed extreme cases, which maximize the performance of the ES. Rechenberg has performed investigations on the basis of a sphere and a corridor test problem for \( n \gg 1 \). For both problems the success probability, leading to an optimum regarding the ES performance (\( P_{\text{sphere}} \approx 0.27; P_{\text{corridor}} \approx 0.184 \), was determined. Rechenberg regarded these test problems as representative for real-world fitness functions. On the basis of these results he recommended to execute the ES under a success probability of 0.2 (1/5th rule), which is a compromise between the results of the investigated test problems. The desired success probability can be obtained by adapting the mutation strength during the optimization. Figure (2.7) shows the implementation of the 1/5th rule. For the computation of the success probability, \( G \) generations are created by a constant mutation strength and the successful mutations \( G_s \) are counted. The obtained success probability is compared with the desired value of 0.2 and, if necessary, the step size is multiplicatively adapted. This procedure assures a positive sign of the mutation strength, as the modification is performed multiplicatively. The best possible value of the exogenous parameter depends on the fitness function, the choice of \( G \) and the dimension of the search space. Schwefel recommended using \( 0.85 \leq a < 1 \), provided the search space consists of more than 30 dimensions and \( G \) is selected to be equal to \( n \) [27]. Finally, one has to bear in mind, that the 1/5th-rule is only suitable for the adaptation of a single strategy parameter. Additionally, this adaptation procedure is usually only applied for a (1 + 1)-ES.
2.4. EVOLUTIONARY STRATEGIES

1. perform the \((1 + 1)\) ES for a number \(G\) of generations:
   - keep \(\sigma\) constant during this period
   - count the number \(G_s\) of successful mutations during this period

2. determine an estimate of \(P_s\) by
   \[ P_s = \frac{G_s}{G} \]

3. change \(\sigma\) according to
   \[ \sigma = \begin{cases} 
   \frac{\sigma}{a} & \text{if } P_s > 1/5 \\
   \sigma a & \text{if } P_s < 1/5 \\
   \sigma & \text{if } P_s = 1/5 
   \end{cases} \]

4. goto 1

Figure 2.7: Implementation of the \(1/5\)th rule [27].

The self-adaptation, which is not restricted to a single strategy parameter, is another procedure for changing the endogenous parameters. For applying this method, each individual is extended by a set of strategy parameters. Both the position of the individual in the search space and the strategy parameters undergo variation. Nevertheless, the decision whether an individual becomes a parent of the next generation is only performed on the basis of the fitness value, which is determined by the object variables. It is assumed that successful individuals are an indication for a good quality of the parental strategy parameters. In other words, well adapted strategy parameters with respect to the fitness landscape should produce on average fitter individuals and therefore have a higher chance to survive. The above mentioned variation of the strategy parameters consists of recombination and mutation. An intermediate recombination is recommended for the adaptation of the strategy parameters [13]. The self-adaptation approach is suitable for all of the introduced mutation operators. First of all, the procedure for mutating the strategy parameter of an isotropic Gaussian mutation operator is shown (2.38). The space of the individuals \(I\) consists of the search space in conjunction with the space of the strategy parameter \(\sigma\)

\[ I = \mathbb{R}^n \times \mathbb{R}. \]  \hspace{1cm} (2.51)

The parameter \(\sigma\) of (2.38) has to remain positive. Thus, the mutation of \(\sigma\) is performed multiplicatively. In order to avoid any deterministic drift without selection, the distribution, applied for the mutation of the strategy parameter, should give 1.0 as mean value. Furthermore, the distribution should lead to a higher probability of small changes than of large ones. Finally, the probability of the occurrence of a value and its associated reciprocal value should be equal. All the mentioned requirements are fulfilled by the log-normal distribution [24]. Ex-
pression (2.52) shows the density function of the log-normal distribution \( \text{Ln}(\mu, \sigma^2) \) [32]:

\[
f(x) = \begin{cases} 
\frac{1}{\sigma \sqrt{2\pi}} \frac{1}{x} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right] & \text{if } x > 0 \\
0 & \text{if } x \leq 0
\end{cases}, \\
\mu \in \mathbb{R}, \ \sigma \in \mathbb{R}^+.
\] (2.52)

In order to draw a \( \text{Ln}(\mu, \sigma^2) \) distributed random number, relation

\[
x = \exp(\mu + \sigma y) \sim \text{Ln}(\mu, \sigma), \ y \sim \text{N}(0, 1)
\] (2.53)

can be applied [32]. Expression (2.53) transforms a \( \text{N}(0, 1) \) distributed random number \( y \) to a \( \text{Ln}(\mu, \sigma^2) \) distributed one. The update of the mutation strength for an isotropic Gaussian mutation operator is performed on the basis of a product, consisting of a \( \text{Ln}(0, 1) \) random number and the mutation strength of the generation \( g \)

\[
\sigma^{(g+1)} \sim \sigma^{(g)} \exp(\tau \text{N}(0, 1)).
\] (2.54)

The exogenous strategy parameter \( \tau \), which is referred to as learning parameter [27], determines the rate of the self-adaptation. Schwefel (1975) and Beyer (1996) suggested choosing \( \tau \) to be inversely proportional to the square root of the problem dimension

\[
\tau \propto \frac{1}{\sqrt{n}}.
\] (2.55)

For avoiding a mutation strength \( \sigma \) close to zero, which would lead to negligible mutations on average, a threshold is introduced

\[
\sigma < \epsilon_0 \rightarrow \sigma = \epsilon_0.
\] (2.56)

If the mutation strength \( \sigma \) reaches a value below the threshold \( \epsilon_0 \), \( \sigma \) is set equal to the threshold.

Now, the update procedure of the endogenous parameters for the mutation operator implying \( n \) strategy parameters is introduced. This scheme implies an extension of the space of the individuals as given by

\[
I = \mathbb{R}^n \times \mathbb{R}^n.
\] (2.57)

Schwefel (1977) suggested the expression (2.58) for computing the vector \( \sigma^{(g+1)} = (\sigma_1, \ldots, \sigma_n)^T \):

\[
\sigma_i^{(g+1)} \sim \sigma_i^{(g)} \exp(\tau_0 \text{N}(0, 1) + \tau N_i(0, 1)).
\] (2.58)
Expression (2.59) shows recommended values for the learning parameters (Schwe- 
fel, 1977):

\[ \tau_0 \propto \frac{1}{\sqrt{2n}}, \quad \tau \propto \frac{1}{\sqrt{2\sqrt{n}}} \]  

(2.59)

Again, as for the isotropic Gaussian mutation operator, a threshold is intro-
duced for avoiding \( \sigma_i \) values close to zero. The threshold \( \epsilon_0 \) is applied for each component of the vector \( \sigma \)

\[ \sigma_i < \epsilon_0 \rightarrow \sigma_i = \epsilon_0. \]  

(2.60)

Finally, the strategy parameter adaptation of the most general mutation op-
erator (2.50) is discussed. This approach consists, as mentioned above, of \( n \) \( \sigma_i \) values and \( n(n-1)/2 \) rotation angles. Expression (2.61) shows the space of the individuals:

\[ I = \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n(n-1)/2}. \]  

(2.61)

The adaption of the rotation angles differs from the \( \sigma_i \) values, as the desired expected change of these angles is zero. Additionally, expression (2.62) should be complied:

\[ p(\alpha_i) = p(-\alpha_i). \]  

(2.62)

These requirements are fulfilled by a Gaussian normal distribution and an additive mutation scheme. Expression (2.63) summarizes the adaptation of the \( \sigma_i \) values and the rotation angles:

\[ \begin{align*}
\sigma_i^{(g+1)} &\sim \sigma_i^{(g)} \exp (\tau_0 N(0, 1) + \tau N_i (0, 1)), \\
\alpha_j^{(g+1)} &\sim \alpha_j^{(g)} + \beta N_j (0, 1).
\end{align*} \]  

(2.63)

Schwefel recommends to choose the factor \( \beta \) equal to 0.0873. The rotation angles \( \alpha_j \) are defined within the interval \([-\pi, \pi]\). Whenever an angle \( \alpha_j \) is outside the permissible interval, a mapping is performed as shown by (2.64) [13]:

\[ \left| \alpha_j^{(g+1)} \right| > \pi \rightarrow \alpha_j^{(g+1)} = \alpha_j^{(g+1)} - 2\pi \text{sign} \left( \alpha_j^{(g+1)} \right). \]  

(2.64)

Experimental studies showed that the presented self-adaptation mechanism of the strategy parameters is a noisy process, which can show remarkable fluctuations during the optimization [27]. The above mentioned intermediate recombination is able to reduce these fluctuations. Nevertheless, the introduced self-adaptation can fail. In this case the evolution gets stuck in a local optimum because the mutation operator is unable to create superior offspring with respect to their parents. One has to bear in mind that there is no general solution for this problem. In the
subsequent sections advanced techniques are introduced, which take the history of the search process into account.

2.4.5 Derandomized self adaptation

Ostermeier [35] analyzed the self-adaptation mechanism on the basis of an \((1, \lambda)\)-ES. The investigated algorithm comprises for each offspring \(k\) \((1 \leq k \leq \lambda)\) a global mutation strength variation parameter \(\xi^k\), which takes two possible states with equal probability \((\alpha, 1/\alpha; \alpha \in [1.1; 1.5])\) and a vector \(\bar{\xi}^k\) for the independent variation of the object variables (Individual strategy parameter \(\xi^k_i\)). The components of the vector \(\bar{\xi}^k\) are drawn from an appropriate distribution. Expression (2.65) shows the mutation operator (O: Offspring, P: Parent):

\[
\mathbf{x}_{O_k}^{(g)} = \mathbf{x}_{P}^{(g)} + \xi^k \bar{\xi}^k \mathbf{z}^k. \tag{2.65}
\]

Each component of the random vector \(\mathbf{z}\) is drawn from a normal distribution with zero mean and a variance of one. The random vector \(\mathbf{z}\) is modified multiplicatively by the strategy parameters of the previous generation \(\delta^{(g)}\), the global mutation strength variation \(\bar{\xi}^k\) and the individual strategy parameter variation \(\xi^k\). The fittest individual is selected as a parent for the following generation \((\text{sel} \in 1, \ldots, \lambda, \text{index of the selected individual})\)

\[
\mathbf{x}_{P}^{(s+1)} = \mathbf{x}_{O_{\text{sel}}}^{(g)}. \tag{2.66}
\]

The individual strategy parameters for the next generation are computed on the basis of the parameter variation of the selected offspring

\[
\delta^{(g+1)} = \xi^\text{sel} \xi^\text{sel} \delta^{(g)}. \tag{2.67}
\]

The multiplications of (2.65) and (2.67) are performed component wise.

Ostermeier identified two basic shortcomings of the presented self-adaptation mechanism. First of all, the scheme does not relate the size of the mutation of the objective variables of the selected individual with the variation of the individual strategy parameters for the following generation. As the components of \(\mathbf{z}\) are drawn from a normal distribution, a large mutation is possible, even when a small value of the related individual strategy parameter variation occurs. However, in such a case, the individual strategy parameter variation should reflect the large mutation. Furthermore, the variation of the strategy parameter within a generation is the same as the one between different generations. On the one hand, a distinct variation is necessary for assuring a successful selection procedure. On the other hand, for reducing random fluctuations of the strategy parameters in the generation sequence the inter-generational variation should be much smaller.

Ostermeier suggested a derandomized mutative step-size control in order to overcome the mentioned shortcomings. In this case, the modification of the individual strategy parameters is derived from the sampled random vector \(\mathbf{z}\) of the
selected offspring and not drawn from a distribution. The expected values of $E(|z_i|)$ and $E(x_i)$ are equal, as shown by

$$z_i \sim N(0,1),$$

$$y_i \sim \chi^2 \sim z_i^2,$$

and

$$x_i \sim \chi \sim \sqrt{y_i}.$$ (2.68)

The distribution of the absolute value of each component of $z$ is given by (2.70) [36]. Expression (2.71) [32] shows the computation of the expected value of a $\chi$ distributed random variable $x_i$:

$$E(x_i) = \sqrt{2} \frac{\Gamma ((k + 1)/2)}{\Gamma (k/2)}.$$ (2.71)

As the value $|z_i|$ is a scalar, the degree of freedom $k$ is equal to one (The results of the gamma function $\Gamma$ can be found in [26]):

$$E(x_i) = \sqrt{\frac{2}{\pi}}, \Gamma \left(\frac{1}{2}\right) = \sqrt{\pi}, \Gamma (1) = 1.$$ (2.72)

As the adaptation of the individual strategy parameter is performed multiplicatively, the exponential function is applied for transforming the expected value to one and to obtain an equal probability for the occurrence of variation and its associated reciprocal variation

$$\xi_i = \exp \left(|z_{sel}^i| - \sqrt{\frac{2}{\pi}}\right).$$ (2.73)

The update of the strategy parameters is given by

$$\delta^{(g+1)} = (\xi_{sel}^{i})^{\beta} (\xi_{scal}^{i})^{\beta_{scal}} \delta^{(g)}.$$ (2.74)

For each component of the vector $z$, the absolute value $|z_i|$ is computed. According to expression (2.73), the individual strategy parameter $\delta^g$ is increased or decreased, if $|z_i|$ is bigger or smaller than the expected value $\sqrt{\frac{2}{\pi}}$.

Ostermeier also introduced two exogenous strategy parameters $\beta \ (0 < \beta < 1)$ and $\beta_{scal} \ (0 < \beta_{scal} < 1)$ for reducing the random inter-generational fluctuations without affecting the variation between competing offspring. These parameters allow decreasing the modification of the individual strategy parameters. Finally, the expressions (2.75) summarize the discussed modifications. Again, all the multiplications and powers of vectors refer to components:
\[
\begin{align*}
    x_{G_k}^{(g)} & = x_{P}^{(g)} + \xi_k \delta^{g} z, \\
    x_{P}^{(g+1)} & = x_{G_{sel}}, \\
    \delta^{(g+1)} & = (\xi_{sel})^{\beta} (\xi_{z_{sel}})^{\beta_{seg}} \delta^{(g)}, \\
    \xi_{z_{sel}} & = (\xi_1, \xi_2, \ldots, \xi_n)^T, \text{ with } \xi_i = \exp \left( \frac{|z_{sel}|}{\sqrt{2 \pi}} \right).
\end{align*}
\] 

\text{(2.75)}

2.4.6 Cumulative step-size adaptation

The following section deals with a further generalized derandomized procedure for the adaptation of the global mutation strength in consideration of inter-generational information. In [37] the cumulative step-size adaptation is presented on the basis of a \((\mu/1, \lambda)\)-CSA-ES and a \((\mu/\mu, \lambda)\)-CSA-ES. The latter mentioned algorithm is discussed in this section, which uses an intermediate recombination. The core of the cumulative step-size adaptation (CSA) is the evolution path. Figure 2.8 [33] shows three different evolution paths in an idealized representation, consisting each of five generations. The solid vectors illustrate the connection between the resultant centroids of the intermediate recombination from one generation to the next one and the dashed vector shows the resultant from the initial state to the last generation. As opposed to the vectors connecting the generational sequence of the centroids, the resultant ones differ significantly in length. A long evolution path (length of the resultant vector), shown by 2.8 (right), in comparison with a path obtained by random selection, illustrated by 2.8 (middle), leads under the application of the CSA to an increased mutation strength and vice versa. A mathematical definition of the reference state for assessing the path length is given below. If the evolution path is too short, the search is probably close to an optimum. Hence, the mutation strength is reduced in order to focus the optimization on this region. In the other case, a long evolution path indicates a long distance to the optimum. In such a case it is reasonable to increase the mutation strength.

Figure 2.8: Evolution paths of different length [33].
Expression

\[ x_k^{(g+1)} = \langle x \rangle_{\mu}^{(g)} + \delta^{(g)} z_k \]  

(2.76)

illustrates the isotropic mutation of the offspring \( k \) \((1 \leq k \leq \lambda)\) originating from the centroid \( \langle x \rangle_{\mu}^{(g)} \) of the previous generation

\[ \langle x \rangle_{\mu}^{(g)} = \frac{1}{\mu} \sum_{i \in I_{sel}^{(g)}} x_i^{(g)} . \]  

(2.77)

\( I_{sel}^{(g)} \) contains the \( \mu \) indices, selected within the generation \( g \). \( \delta^{(g)} \in \mathbb{R} \) is the mutation strength of the generation \( g \) \((\delta^{(g)} > 0)\) and \( z_k \) is a normal distributed random vector. The components of \( z_k \) are drawn independently on the basis of a zero mean and a variance of one. Each \( z_k \) is determined independently from the other random vectors of the generation \( g+1 \). A mathematical formulation of the evolution path is given by

\[ s^{(g+1)} = (1-c) s^{(g)} + c u \sqrt{\mu} \delta^{(g)} \left( \langle x \rangle_{\mu}^{(g+1)} - \langle x \rangle_{\mu}^{(g)} \right) , \]  

(2.78)

which is a weighted sum of the consecutive centroids and

\[ \delta^{(g+1)} = \delta^{(g)} \exp \left( \frac{||s^{(g+1)}|| - E(||N(0,1)||)}{DE(||N(0,1)||)} \right) \]  

(2.79)

defines the update of the mutation strength. Expression (2.79) compares the Euclidean norm of the evolution path \( s^{(g+1)} \) with the expected value of the Euclidean norm of a normal distributed random vector with zero mean and a identity matrix as a covariance matrix \((E(||N(0,1)||))\).

The expressions

\[ z \sim N(0, I) , \]  

(2.80)

\[ y \sim \chi^2 \sim z^T z \]  

(2.81)

and

\[ x \sim \chi \sim \sqrt{y} \]  

(2.82)

show the equivalence of the random variables \( ||z|| \) and \( x \). \( E(||N(0,1)||) \) is the expected value of a \( \chi \) distribution

\[ E(||z||) = E(||N(0,1)||) = E(x) = \sqrt{2} \frac{\Gamma((k+1)/2)}{\Gamma(k/2)} . \]  

(2.83)

The degree of freedom \( k \) of this distribution is identical to the dimension of the search space. If the length of the evolution path \( s^{(g+1)} \) is smaller than
42  

CHAPTER 2. OPTIMIZATION

\( E (\|N (0, I)\|) \), the global mutation strength is decreased according to (2.79). In the opposite case, the mutation strength is increased. The expected length of a normal distributed random vector is the above mentioned reference state for assessing the evolution path length, which is obtained from random selection. If the Euclidean norm of \( s^{(g+1)} \) is equal to \( E (\|N (0, I)\|) \), the global mutation strength remains unchanged. The successful steps are expected to be perpendicular to each other [33]

\[
x_{\text{sel}}^{(g)} \cdot x_{\text{sel}}^{(g+1)} \approx 0 ; \quad z_{\text{sel}}^{(g+1)} = \frac{\sqrt{\mu}}{\delta^{(g)}} \left( \langle x \rangle_{\mu}^{(g+1)} - \langle x \rangle_{\mu}^{(g)} \right).
\]  

(2.84)

Large values of the exogenous damping parameter \( D \) lead to a slow adaptation of the mutation strength. The exogenous parameter \( c \) weights the influence of the history with respect to the evolution path. In the limit case \( c = 1 \) the history of the evolution path is ignored. Hansen 1998 recommended for the mentioned parameters the values

\[
c = \frac{1}{\sqrt{\mu}}; \quad D = \sqrt{\mu}.
\]  

(2.85)

Originating from

\[ z = ax, \quad x \sim N(0, \sigma_x^2), \quad z \sim N(0, a^2 \sigma_x^2) \]  

(2.86)

and

\[ x \sim N(0, \sigma_x^2), \quad y \sim N(0, \sigma_y^2), \quad x + y \sim N(0, \sigma_x^2 + \sigma_y^2), \]  

(2.87)

which gives the rule for the addition of variances of normal distributions,

\[
\frac{1}{\mu} \sum_{i=1}^{\mu} z_i \sim \frac{1}{\sqrt{\mu}} N (0, I), \quad z_k \sim N (0, I)
\]  

(2.88)

can be derived [38]. Expression

\[
\langle x \rangle_{\mu}^{(g+1)} \sim \langle x \rangle_{\mu}^{(g)} + \delta^{(g)} \frac{1}{\mu} \sum_{i=1}^{\mu} z_i
\]  

(2.89)

is derived from (2.76) and (2.77). The expressions (2.88), (2.89) and (2.90) are derived under the assumption of a random selection. Inserting (2.88) into (2.89) leads to (2.90), which shows that the difference vector of the centroids between the generations \( g \) and \( g + 1 \) is normally distributed:

\[
\frac{\sqrt{\mu}}{\delta^{(g)}} \left( \langle x \rangle_{\mu}^{(g+1)} - \langle x \rangle_{\mu}^{(g)} \right) \sim N (0, I).
\]  

(2.90)

As (2.90) is normally distributed under random selection, \( s^{(g+1)} \) is also equally distributed, if the constant \( c_u \) is chosen according to (2.91) and \( s^{(0)} \sim N (0, I) \)
2.4. EVOLUTIONARY STRATEGIES

\[ s^{(g+1)} \sim s^{(g)} \sim N(0, I) \Rightarrow (1 - c)^2 + c_u^2 = 1^2. \]  \hspace{1cm} (2.91)

This expression is derived by applying (2.87). An extension of the introduced cumulative step-size adaptation is shown in the next section, which is not restricted to an isotropic mutation operator.

2.4.7 Covariance matrix adaptation

The best adaptation of the mutation operator with respect to the fitness landscape is obtained by deploying a covariance matrix implying a global scaling, a scaling for each eigenvector and an orthogonal rotation of the coordinate system defined by the eigenvectors. As introduced above, the related covariance matrix can be derived from a set of strategy parameters. Another approach is to learn the covariance matrix directly from the optimization sequence. The adaptation of it is determined by the history of successful steps and the selected individuals of the current population. This procedure is referred to as covariance matrix adaptation (CMA). The introduction of the CMA is mainly taken from [33].

The offspring is drawn from a Normal distribution, as given by

\[ x^{(g+1)}_k \sim \sigma^{(g)} N\left(m^{(g)}, C^{(g)}\right). \] \hspace{1cm} (2.92)

For the numerical computation of the random vector, expression (2.41) is applied. \( \sigma^{(g)} \in \mathbb{R}^+ \) is the global mutation strength. \( x^{(g+1)}_k \) the k-th offspring of the generation \( g + 1 \), whereas \( m^{(g)} \in \mathbb{R}^n \) is the centroid of the generation \( g \) and \( C^{(g)} \) the covariance matrix of the generation \( g \).

The following section treats the update procedures for \( m, C \) and \( \sigma \). Hansen applies an extended intermediate recombination, which implies additional weights for each parent

\[ m^{(g+1)} = \sum_{i=1}^{\mu} w_i x^{(g+1)}_{i:}\lambda \] \hspace{1cm} (2.93)

for updating \( m \). All parents are involved in the recombination process, therefore \( \rho \) is equal to \( \mu \). The expression \( i : \lambda \) ranks the individuals according to their fitness value as shown by

\[ f\left(x^{(g+1)}_{1:}\lambda\right) \leq f\left(x^{(g+1)}_{2:}\lambda\right) \leq \cdots \leq f\left(x^{(g+1)}_{\lambda:}\lambda\right). \] \hspace{1cm} (2.94)

The higher the fitness of the parents the higher their weight for the recombination is. Expression (2.95) defines the weight factors:

\[ \sum_{i=1}^{\mu} w_i = 1, \ w_1 \geq w_2 \geq \cdots \geq w_\mu, \ w_i \in \mathbb{R}^+. \] \hspace{1cm} (2.95)
The expression (2.96) is a useful definition for the subsequent presentation of the CMA-ES:

\[
\mu_{\text{eff}} = \left( \frac{\sum_{i=1}^{\mu} w_i}{\|w\|} \right)^2 = \left( \sum_{i=1}^{\mu} w_i^2 \right)^{-1}.
\]  (2.96)

The lowest possible value of \(\mu_{\text{eff}}\) is 1.0. In this case, one parent has a weight of 1.0 and the remaining parents a weight of 0.0. If all parents have the same weight \(\frac{1}{\mu}\), \(\mu_{\text{eff}}\) is equal to \(\mu\), which is the highest possible value of \(\mu_{\text{eff}}\).

Generally, a fast search is desired, which implies a small population size \(\lambda\). Under this condition, it is difficult to obtain a reliable estimation of the covariance matrix. Hansen suggested deploying information from previous generations in order to improve the computation of the covariance matrix. Expression (2.97) illustrates the estimation of the covariance matrix within the generation \(g+1\):

\[
C^{(g+1)}_{\mu} = \sum_{i=1}^{\mu} w_i \left( x_{i;\lambda}^{(g+1)} - m^{(g)} \right) \left( x_{i;\lambda}^{(g+1)} - m^{(g)} \right)^T.
\]  (2.97)

The same weights are applied as for the extended intermediate recombination. Hansen [33] interprets (2.97) as an estimation of the variances of the sampled steps, because the mean is taken from the previous generation. Expression (2.98) demonstrates the assignment of prior information to the covariance matrix:

\[
C^{(g+1)} = (1 - c_{\mu}) C^{(g)} + \frac{c_{\mu}}{\sigma^{(g)^2}} C^{(g+1)}_{\mu}.
\]  (2.98)

The recent generations possess a higher weight than the older ones. The exogenous parameter \(c_{\mu}\) controls the significance of the older information with respect to the current covariance matrix and is referred to as learning rate. If \(c_{\mu}\) is chosen to be zero, the initial covariance matrix is preserved. A value of \(c_{\mu} = 1\) means that any prior information is discarded. Inserting (2.97) into (2.98) and applying

\[
y_{i;\lambda}^{(g+1)} = \frac{x_{i;\lambda}^{(g+1)} - m^{(g)}}{\sigma^{(g)}}
\]  (2.99)

gives

\[
C^{(g+1)} = (1 - c_{\mu}) C^{(g)} + c_{\mu} \sum_{i=1}^{\mu} w_i y_{i;\lambda}^{(g+1)} y_{i;\lambda}^{(g+1)^T}.
\]  (2.100)

The parameter \(c_{\mu}\) balances between two extreme cases. If \(c_{\mu}\) is chosen too big, the covariance matrix could degenerate and a small value of \(c_{\mu}\) leads to slow learning. Usually, as an initial covariance matrix \(C^{(0)}\) the unity matrix \(I\) is utilized.
2.4. EVOLUTIONARY STRATEGIES

A drawback of the presented estimation of the covariance matrix is that the sign information of the steps is not reflected \( yy^T = (-y)(-y)^T \). Hansen proposed to overcome this shortcoming by introducing a so-called evolution path. Successful steps are accumulated and stored in a vector \( p_c \). Expression (2.101) defines this vector:

\[
p^{(g+1)}_c = (1 - c_c) p^{(g)}_c + \sqrt{c_c (2 - c_c) \mu_{eff}} \left( \frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \right).
\]

(2.101)

The factor of the second summand on the right hand side is a normalization constant and \( c_c \) defines the significance of the previous generations similar as \( c_\mu \). Expression (2.102) shows the generation of a covariance matrix on the basis of an evolution path:

\[
C^{(g+1)} = (1 - c_1) C^{(g)} + c_1 p^{(g+1)}_c p^{(g+1)T}_c.
\]

(2.102)

Both approaches for adapting the covariance matrix can be combined as given by

\[
C^{(g+1)} = (1 - c_1 - c_\mu) C^{(g)} + c_1 p^{(g+1)}_c p^{(g+1)T}_c + c_\mu \sum_{i=1}^{\mu} w_i \lambda^{(g+1)T} y_{i;\lambda}^{(g+1)}.
\]

(2.103)

Thereby the information of each generation (estimation of the covariance matrix) and the correlations between the generations (evolution path) are efficiently used for updating the covariance matrix. The constants \( c_1 \) and \( c_\mu \) control the significance of the information of the previous generations. Hansen [33] recommends to choose \( c_1 \) and \( c_\mu \) according to

\[
c_1 \approx \frac{2}{n^2}
\]

(2.104)

and

\[
c_\mu \approx \min \left( \mu_{eff} \frac{1}{n^2}, 1 - c_1 \right).
\]

(2.105)

Additionally, the CMA-ES comprises an overall scale of the mutation operator, which is similar to the above introduced cumulative step-size adaptation. This mechanism is introduced, as the largest reliable learning rate for the covariance matrix update (2.103) is too slow for achieving the desired change rates of the global mutation strength. Expression

\[
p^{(g+1)}_\sigma = (1 - c_\sigma) p^{(g)}_\sigma + \sqrt{c_\sigma (2 - c_\sigma) \mu_{eff}} C^{(g)-1} \frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}}
\]

(2.106)

shows a slightly modified version of (2.78), which is adapted to the weighted intermediate recombination and contains a rescaling of the eigenvalues. For the
exogenous parameter $c_\sigma$, which weights the influence of the previous generations on the evolution path, Hansen recommends the same range as for $c$. 

The matrix $C(g)^{-\frac{1}{2}}$, defined as

$$C(g)^{-\frac{1}{2}} = B(g)D(g)^{-1}B(g)^T,$$  \hspace{1cm} (2.107)

eliminates the scaling of the axes, which is caused by the covariance matrix $C(g)$ within the mutation. However, the rotation of the axis is not affected by $C(g)^{-\frac{1}{2}}$. On the basis of this rescaling, it is possible to compare the length of the evolution path with the expected length of a $N(\mathbf{0}, \mathbf{I})$ distributed normal vector. The fundamental idea of the adaption of the global mutation strength is the same as mentioned in the section about the cumulative step-size adaptation. A successful adaptation of the global mutation strength is obtained, if the vectors, connecting the centroids of each generation are in expectation perpendicular to each other in the rescaled space. If the centroids of each generation are considered in the search space, expression

$$\left( m(g) - m(g-1) \right)^T C(g)^{-1} \left( m(g+1) - m(g) \right) \approx 0$$ \hspace{1cm} (2.108)

holds. In other words, the vectors $m(g+1) - m(g)$ and $m(g) - m(g-1)$ are in expectation $C^{-1}$ conjugate. The derivation of (2.108) can be found in [33]. Apart from the constant $c_\sigma$ and the formula signs, the adaptation of the global mutation strength is identically computed as (2.79):

$$\sigma^{(g+1)} = \sigma^{(g)} exp \left[ \frac{c_\sigma}{d_\sigma} \left( \frac{\| p_{\sigma}^{(g+1)} \|_E}{\| N(\mathbf{0}, \mathbf{I}) \|} - 1 \right) \right].$$ \hspace{1cm} (2.109)

2.4.8 $(1+\lambda)$-CMA-ES

Subsequently, a $(1+\lambda)$-CMA-ES is introduced, which is developed by Igel et al. [20]. The offspring is precreated in the same way, as shown for the $(\mu/\mu, \lambda)$-CMA-ES (Expression (2.92)). The best fitness value of each generation, comprising $\lambda$ offspring, is denoted by the symbol $x_{1,\lambda}^{(g+1)} \in \mathbb{R}^n$. For the adaptation of the step size $\sigma$, a procedure similar to the 1/5th rule (Rechenberg 1973) is applied. Thereby, $\lambda_{\text{suc}}^{(g+1)}$ is defined as the number of offspring concerning a generation, whose fitness is better than the one of the parent. The success probability $p_{\text{suc}}^{(g+1)} = \frac{\lambda_{\text{suc}}^{(g+1)}}{\lambda}$ is smoothed, as given by

$$\overline{p}_{\text{suc}}^{(g+1)} = (1-c_p)p_{\text{suc}}^{(g)} + c_p p_{\text{suc}}^{(g+1)}.$$ \hspace{1cm} (2.110)

The adaptation of the step size $\sigma$ (2.111) is performed multiplicatively:
2.4. EVOLUTIONARY STRATEGIES

\[ \sigma^{(g+1)} = \sigma^{(g)} \exp \left( \frac{1}{d} \frac{p^{(g+1)}_{\text{succ}} - p^{\text{target}}_{\text{succ}}}{1 - p^{\text{target}}_{\text{succ}}} \right). \] (2.111)

In order to limit the adaptation of the step size by the damping factor, the result of the sub term \( \frac{p^{(g+1)}_{\text{succ}} - p^{\text{target}}_{\text{succ}}}{1 - p^{\text{target}}_{\text{succ}}} \) must lie within the interval \( [-1; 1] \). For obtaining the desired interval, \( p^{\text{target}}_{\text{succ}} < 0.5 \) must hold. This restriction of the mentioned sub term leads to an argument of the exponential function, which is larger than \( -\frac{1}{d} \) and smaller than \( \frac{1}{d} \).

Provided \( x^{(g+1)}_{1,\lambda} \) corresponds to a better fitness value than the parent \( x^{(g)}_{\text{parent}} \), this offspring is taken as a parent of the next generation and the covariance matrix is updated. The update procedure of the covariance matrix also depends on the smoothed success probability \( p^{(g+1)}_{\text{succ}} \), as shown by the expressions

\[
\begin{align*}
\text{if} & \ p^{(g+1)}_{\text{succ}} < p_{\text{thresh}} \\
& \quad p^{(g+1)}_c = (1 - c_c) p^{(g)}_c + \sqrt{c_c (2 - c_c)} \left[ \frac{x^{(g+1)}_{\text{parent}} - x^{(g)}_{\text{parent}}}{\sigma^{(g)}_{\text{parent}}} \right]
\end{align*}
\] (2.112)

and

\[
\begin{align*}
\text{if} & \ p^{(g+1)}_{\text{succ}} \geq p_{\text{thresh}} \\
& \quad p^{(g+1)}_c = (1 - c_c) p^{(g)}_c
\end{align*}
\]

\[
\begin{align*}
\text{if} & \ p^{(g+1)}_{\text{succ}} < p_{\text{thresh}} \\
& \quad C^{(g+1)} = (1 - c_{\text{cov}}) C^{(g)} + c_{\text{cov}} p^{(g+1)}_c p^{(g)}_c T^{(g+1)} \\
\text{if} & \ p^{(g+1)}_{\text{succ}} \geq p_{\text{thresh}} \\
& \quad C^{(g+1)} = (1 - c_{\text{cov}}) C^{(g)} + c_{\text{cov}} (p^{(g+1)}_c p^{(g)}_c T^{(g+1)} + c_c (2 - c_c) C^{(g)})
\end{align*}
\] (2.113)

Thereby, the update of the covariance matrix takes the actual step into account, if the smoothed success probability is below the threshold. In this case, the evolution path and the adaptation of the covariance matrix is computed in the same way as shown for the \( (\mu/\mu, \lambda) \)-CMA-ES. The term \( \sqrt{c_c (2 - c_c)} \) normalizes the variance of \( p_c \), which is considered to be a random variable (see [20]).

According to (2.112), the evolution path shrinks, if the smoothed success probability exceeds the threshold. For preserving the variance of \( C \) in this case, which is affected by the shrinking evolution path, the update of the covariance matrix is complemented by the term \( c_c (2 - c_c) C^{(g)} \). The threshold \( p_{\text{thresh}} \) limits the adaptation of the covariance matrix in the case of small step sizes. Table 2.2 summarizes recommendations regarding the choice of the exogenous parameters, as given in [20]. The suggestion of the success probability \( p^{\text{target}}_{\text{succ}} \) lies in the same dimension as 1/5th rule of Rechenberg.
Table 2.2: Recommended values of the exogenous parameters of the \((1+\lambda)\)-CMA-ES algorithm.

<table>
<thead>
<tr>
<th>Adaptation Step Size</th>
<th>(d = 1 + \frac{n}{2\lambda})</th>
<th>(p_{\text{target}} = \frac{1}{\sqrt{4+\lambda}})</th>
<th>(e_p = \frac{p_{\text{target}}}{\sqrt{n^2+6}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptation Covariance Matrix</td>
<td>(c_c = \frac{2}{n+2})</td>
<td>(c_{\text{cov}} = \frac{2}{n^2+6})</td>
<td>(p_{\text{thresh}} = 0.44)</td>
</tr>
</tbody>
</table>

2.5 **Newton method**

The Newton method is an optimization procedure, which is based on the second derivative of the objective function. The application of this algorithm is limited to differentiable and unconstrained minimization problems as given by:

\[
\text{minimize: } f(x), \ x \in \mathbb{R}^n, \ f \in \mathbb{R}.
\]  

(2.114)

Thereby, the optimization procedure identifies the root of the first derivative

\[
\nabla f(x) = 0.
\]  

(2.115)

It has to be mentioned that this root could also correspond to a local maximum or a saddle point of \(f(x)\). For investigating the computed stationary point, the Hessian matrix \(\nabla^2 f(x)\) can be calculated. Provided this matrix is positive definite, the considered stationary point is a local minimum. The expressions

\[
\nabla f = \frac{\partial f(x)}{\partial x} = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1}, & \frac{\partial f(x)}{\partial x_2}, & \ldots, & \frac{\partial f(x)}{\partial x_n} \end{bmatrix}^T
\]  

(2.116)

and

\[
\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_i \partial x_k} \end{bmatrix}_{i,k=1,\ldots,n}
\]  

(2.117)

define the first and second derivative respectively. The root of the first derivative is computed based on a linearization of the stationary condition with respect to the current position \(x^{(n)}\) (2.115). The desired stationary point is iteratively determined. Expression

\[
\nabla f(x^{(n)}) + \nabla^2 f(x^{(n)}) \Delta x^{(n+1)} = 0
\]  

(2.118)

shows the linearization of the \(n\)th iteration. This expression is solved for the vector \(\Delta x^{(n+1)}\), which gives, in consideration of
2.5. NEWTON METHOD

\[ x^{(n+1)} = x^{(n)} + \Delta x^{(n+1)}, \quad (2.119) \]
the update of the solution \( x^{(n+1)} \) regarding the \( n \)th iteration

\[ x^{(n+1)} = x^{(n)} - \left[ \nabla^2 f(x^{(n)}) \right]^{-1} \nabla f(x^{(n)}). \quad (2.120) \]

In order to apply the Newton method for constrained problems, as defined by

\[
\begin{align*}
\text{minimize:} & \quad f(x), \quad f \in \mathbb{R}, \quad x \in \mathbb{R}^n, \\
g_j(x) & \leq 0, \quad j = 1, ..., n_g, \\
h_i(x) & = 0, \quad i = 1, ..., n_h,
\end{align*}
\]

(2.121)

the objective function can be exchanged by a Lagrange function. This function, as defined by the expression (2.122), enables to transform the constrained problem to an unconstrained one:

\[ L(x, \lambda, \mu) = f(x) + \sum_{j=1}^{n_g} \lambda_j g_j(x) + \sum_{k=1}^{n_h} \mu_k h_k(x). \quad (2.122) \]

Provided, the problem comprises only equality constraints \( n_g = 0; n_h > 0 \), the stationary point of the Lagrange function

\[ \nabla L(x, \bar{\lambda}, \bar{\mu}) = 0 \quad (2.123) \]

is a necessary condition of a local minimum. However, the stationary point is only a local minimum, if the Hessian matrix is positive definite.

In the general case, comprising equality and inequality constraints \( (n_g > 0; n_h > 0) \), the Kuhn-Tucker optimality conditions (2.124), (2.125), (2.126) and (2.127) have to be applied for the determination of a local minimum:

\[
\begin{align*}
\bar{\lambda}_j & \geq 0, \quad (2.124) \\
\nabla_x L(x, \bar{\lambda}, \bar{\mu}) & = 0, \quad (2.125) \\
\bar{\lambda}_j g_j(x) & = 0, \quad (2.126) \\
h_i(x) & = 0. \quad (2.127)
\end{align*}
\]

Expression

\[ \nabla_x L(x, \lambda, \mu) = \nabla_x f(x) + \sum_{j=1}^{n_g} \lambda_j \nabla x g_j(x) + \sum_{i=1}^{n_h} \mu_i \nabla x h_i(x) \quad (2.128) \]
gives the partial derivative of $L(x, \lambda, \mu)$ with respect to $x$. For solving the optimization problem on the basis of the Kuhn-Tucker optimality conditions, the equations (2.125), (2.126) and (2.127) need to be linearized as given by (2.129), (2.130) and (2.131) [26]:

$$\nabla_x L(x^{(n)}, \lambda^{(n)}, \mu^{(n)}) + (\nabla^2_x f(x^{(n)}) + \sum_{j=1}^{n_g} \lambda_j^{(n)} \nabla^2_x g_j(x^{(n)}) + \sum_{i=1}^{n_h} \mu_i^{(n)} \nabla^2_x h_i(x^{(n)}) \Delta x^{(n+1)} + \sum_{j=1}^{n_g} \nabla_x g_j(x) \Delta \lambda_j^{(n+1)} + \sum_{i=1}^{n_h} \nabla_x h_i(x) \Delta \mu_i^{(n+1)} = 0, \quad (2.129)$$

$$\lambda_j^{(n)} g_j(x^{(n)}) + g_j(x^{(n)}) \Delta \lambda_j^{(n+1)} + \lambda_j^{(n)} \nabla_x g_j(x^{(n)})^T \Delta x^{(n+1)} = 0, \quad (2.130)$$

$$h_i(x^{(n)}) + \nabla_x h_i(x^{(n)})^T \Delta x_i^{(n+1)} = 0. \quad (2.131)$$

For the linearization, the variables $\Delta x, \Delta \lambda_j$ and $\Delta \mu_i$ are introduced. Now, apart from $x$, also the Lagrange multipliers $\lambda_j$ and $\mu_j$ need to be updated as given by the expressions:

$$x^{(n+1)} = x^{(n)} + \Delta x^{(n+1)}, \quad (2.132)$$

$$\lambda_j^{(n+1)} = \lambda_j^{(n)} + \Delta \lambda_j^{(n+1)}, \quad (2.133)$$

$$\mu_i^{(n+1)} = \mu_i^{(n)} + \Delta \mu_i^{(n+1)}. \quad (2.134)$$

On the basis of the shown equation system (2.129), (2.130) and (2.131) the desired solution can be computed iteratively.

### 2.6 Sequential quadratic programming

Unfortunately, the Newton method can converge to a solution, which violates the constraints [25]. In order to avoid this shortcoming of the Newton method, Sequential Quadratic Programming algorithms can be deployed. The introduction of these algorithms is mainly taken from [25]. As a first step, additional constraints are defined:

$$\lambda_i^{(n+1)} \geq 0, \quad g_j(x^{(n)}) + \nabla_x g_j(x^{(n)})^T \Delta x^{(n+1)} \leq 0. \quad (2.135)$$
2.6. SEQUENTIAL QUADRATIC PROGRAMMING

By applying the relation \( \lambda_j^{(n+1)} = \lambda_j^{(n)} + \Delta \lambda_j^{(n+1)} \) expression (2.130) can be rearranged as given by

\[
\lambda_j^{(n+1)} g_j(x^{(n)}) + \lambda_j^{(n)} \nabla_x g_j(x^{(n)})^T \Delta x^{(n+1)} = 0. 
\] (2.136)

Additionally, if \( \lambda_j^{(n)} \) of the second summand of (2.136) is exchanged by \( \lambda_j^{(n+1)} \), expression

\[
\lambda_j^{(n+1)}(g_j(x^{(n)}) + \nabla_x g_j(x^{(n)})^T \Delta x^{(n+1)}) = 0 
\] (2.137)

is obtained. Expression

\[
\nabla_x f(x^n) + \sum_{j=1}^{n_g} \lambda_j^{(n)} \nabla_x g_j(x^{(n)}) + \sum_{i=1}^{n_h} \mu_i^{(n)} \nabla_x h_i(x^{(n)}) + B^{(n)} \Delta x^{(n+1)} + \\
\sum_{j=1}^{n_g} \nabla_x g_j(x^{(n)}) \Delta \lambda_j^{(n+1)} + \sum_{i=1}^{n_h} \nabla_x h_i(x^{(n)}) \Delta \mu_i^{(n+1)} = 0 
\] (2.138)

is derived by inserting (2.128) in (2.129) and introducing the abbreviation \( B^{(n)} \) for the terms comprising the second derivative of \( f(x) \) with respect to \( x \). The relations \( \lambda_j^{(n+1)} = \lambda_j^{(n)} + \Delta \lambda_j^{(n+1)} \) and \( \mu_i^{(n+1)} = \mu_i^{(n)} + \Delta \mu_i^{(n+1)} \) allow to write (2.138) in the form given by

\[
\nabla_x f(x^n) + B^{(n)} \Delta x^{(n+1)} + \sum_{j=1}^{n_g} \nabla_x g_j(x^{(n)}) \lambda_j^{(n+1)} + \sum_{i=1}^{n_h} \nabla_x h_i(x^{(n)}) \mu_i^{(n+1)} = 0. 
\] (2.139)

For a better overview, the additional constraints (2.135) and the rearranged equations (2.139), (2.137) are summarized by

\[
\lambda_j^{(n+1)} \geq 0, \\
\nabla_x f(x^n) + B^{(n)} \Delta x^{(n+1)} + \sum_{j=1}^{n_g} \nabla_x g_j(x^{(n)}) \lambda_j^{(n+1)} + \sum_{i=1}^{n_h} \nabla_x h_i(x^{(n)}) \mu_i^{(n+1)} = 0, \\
g_j(x^{(n)}) + \nabla_x g_j(x^{(n)})^T \Delta x_j^{(n+1)} \leq 0, \\
h_i(x^{(n)}) + \nabla_x h_i(x^{(n)})^T \Delta h_i^{(n+1)} = 0. 
\] (2.140)

Equation (2.131) is not affected by the additional constraints and is therefore repeated. Due to the linearization of the original problem and the introduced
additional constraints (2.135), the equations (2.140) describe the Kuhn-Tucker optimality conditions of a quadratic function as given by

\[
\begin{align*}
\text{minimize:} & \quad \nabla_x f(x^{(n)})^T s + \frac{1}{2} s^T B^{(n)} s, \\
g_j(x^{(n)}) + \nabla_x g_j(x^{(n)})^T s & \leq 0, \\
h_i(x^{(n)}) + \nabla_x h_i(x^{(n)})^T s & = 0.
\end{align*}
\]

(2.141)

The shown considerations allow formulating an optimization algorithm which is referred to as Sequential Quadratic Programming (SQP). Starting from a solution \(x^{(n)}\), it is possible to formulate an optimization problem on the basis of a quadratic function, which is also called quadratic program, for improving the solution \(x^{(n)}\). Thereby, each optimization step comprises the solution of a quadratic sub program. A drawback of the method is the determination of the Hessian matrix, which is computational expensive. Quasi-Newton methods give an approximation of the Hessian matrix by a reduced cost. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) is a well known member of the class of the quasi-Newton methods. Thereby, [39] and [25] are given as references for this method. Provided the Hessian matrix \(B^{(n)}\) is positive semidefinite \(\Delta x^T B^{(n)} \Delta x \geq 0\), the quadratic problem (2.141) is convex and a solution, which complies the Kuhn-Tucker conditions is a global minimum. For this case efficient solution strategies, like the interior-point methods, exist, which are described in [25]. In order to enforce a positive semidefinite Hessian matrix, the BFGS method can be modified as suggested by Powell [25].

### 2.7 Summary

Generally, evolutionary strategies are characterized by low demands regarding the structure of the fitness landscape. These optimization procedures are able to search for an optimum based on zero-order information. As derivatives of the fitness function are not needed, evolutionary strategies are suitable for discontinuous optimization problems. The ability to overcome local minima is a further property of these optimization algorithms. For the analysis of the constitutive laws, real-valued as well as integer-valued parameters have to be considered. Therefore, for this task an evolutionary strategy comprising the introduced derandomized self adaptation is applied, as this algorithm is able to treat mixed-integer optimization problems. However, some investigations of this thesis are solely based on real-valued object variables. In these cases also the \((\mu/\mu, \lambda)\)-CMA-ES, whose field of application is limited to real-valued object variables, is applied and compared with the \((1, \lambda)\)-DR-ES. Provided the corresponding landscape of the optimization problem is unimodal, it is expected that the \((1 + 1)\)-CMA-ES algorithm shows a better performance in comparison with the \((\mu/\mu, \lambda)\)-CMA-ES [20]. This property
of the (1 + 1)-CMA-ES can give additional insight into the optimization task. As long as the search spaces of the optimization tasks are real-valued in this thesis, the performance of the evolutionary strategies is compared with the deterministic SQP algorithm.