Chapter 1

Evolution Strategies

1.1 Background

The paradigm of Evolutionary Computation (EC), which is gleaned from the model of organic evolution, studies populations of candidate solutions undergoing variations and selection, and aims at benefiting from the collective phenomena of their generational behavior. The term Evolutionary Algorithms (EAs) essentially refers to the collection of such generic methods, inspired by the theory of natural evolution, that encode complex problems into an artificial biological environment, define its genetic operators, and simulate its propagation in time. Motivated by the basic principles of the Darwinian theory, it is suggested that such simulation would yield an optimal solution for the given problem.

Evolutionary Algorithms [1] have three main streams, rooted either in the United States or in Germany, during the 1960s: Evolutionary Programming (EP), founded by L. Fogel in San-Diego [2], Genetic Algorithms (GAs) founded by J. Holland in Ann Arbor [3, 4], and Evolution Strategies (ES), founded by P. Bienert, H.P. Schwefel and I. Rechenberg, three students to that time at the Technical University of Berlin (see, e.g., [5, 6, 7]).

Evolution Strategies for global parameter optimization, the general framework of this study, is reviewed in this chapter. We start with laying out the basic foundations and definitions.

1.1.1 The Framework: Global Optimization

Let us introduce the elementary terminology of a continuous real-valued parameter optimization problem [8]. The following definition excludes discrete and mixed-integer problems. Given an objective function, also called the target function,

\[ f : S \subseteq \mathbb{R}^n \rightarrow \mathbb{R}, \quad S \neq \emptyset \]
where $S$ is the set of feasible solutions

$$S = \{ \bar{x} \in \mathbb{R}^n \mid g_j(\bar{x}) \geq 0 \; \forall j \in \{1, \ldots, q\} \}, \quad g_j(\bar{x}) : \mathbb{R}^n \to \mathbb{R}$$

subject to $q$ inequality constraints $g_j(\bar{x})$, the goal is to find a vector $\bar{x}^* \in S$ which satisfies

$$\forall \bar{x} \in S : f(\bar{x}) \geq f(\bar{x}^*) = f^*$$

Then, $f^*$ is defined as the global minimum and $\bar{x}^*$ is the global minimum location.

Due to

$$\min \{ f(\bar{x}) \} = -\max \{ -f(\bar{x}) \},$$

it is straightforward to convert every minimization problem into a maximization problem. Thus, without loss of generality, we shall assume a minimization problem, unless specified otherwise.

A local minimum $\bar{f} = f(\bar{x})$ is defined in the following manner:

$$\exists \epsilon > 0 \; \forall \bar{x} \in S : \| \bar{x} - \bar{x}^* \| < \epsilon \Rightarrow \bar{f} \leq f(\bar{x})$$

Unimodality vs. Multimodality A landscape is said to be unimodal if it has only a single minimum, and multimodal otherwise. It is called multi-global if there are several minima with equal function values as the global minimum.

Global Minimum in Practice: Characterization While there exists a general criterion for the automatic identification of a local minimum, such as the zero gradient criterion, in practice there is no equivalent general criterion for the global minimum [8]. The attempt to characterize it is essentially equivalent to posing the multimodal optimization problem and differentiating de facto between global and local minima. We outline here a theoretical attempt to accomplish this characterization, by means of the important concept of level sets [9, 10]. Given a level set,

$$L_f(\alpha) = \{ \bar{x} | \bar{x} \in S, \; f(\bar{x}) \leq \alpha \}, \quad (1.2)$$

it is subject to level set mapping, which defines its effective domain:

$$G_f = \{ \alpha | \alpha \in \mathbb{R}, \; L_f(\alpha) \neq \emptyset \}.$$  \quad (1.3)

Assuming that $G_f$ is compact and closed, $L_f(\alpha)$ is said to be lower semicontinuous (lsc) at the point $\hat{\alpha} \in G_f$ if $\bar{x} \in L_f(\hat{\alpha})$, $\{ \alpha^i \} \subset G_f$, $\{ \alpha^i \} \to \hat{\alpha}$ imply the existence of $K \in \mathbb{N}$ and a sequence $\{ \bar{x}^i \}$ such that $\{ \bar{x}^i \} \to \bar{x}$ and $\bar{x}^i \in L_f(\alpha^i)$ for $i \geq K$.

Given this, the following is a sufficient condition for characterizing a global minimum:
1.1. Background

**Theorem 1.1.1.** Let \( f \) be a real-valued function on \( S \subset \mathbb{R}^n \). If every \( \bar{x} \in S \) satisfying \( f(\bar{x}) = \alpha \) is either a global minimum of \( f(\cdot) \) on \( S \) or it is not a local minimum of \( f(\cdot) \), then \( L_f(\alpha) \) is lsc at \( \alpha \).

Törn and Zilinskas concluded that the extension to multimodal domains makes the optimization problem unsolvable in the general case, i.e., there is no efficient solution technique for obtaining the global minimum value (see [8] pp. 6).

**The Hessian and the Condition Number** Given a real-valued twice differentiable \( n \)-dimensional function \( f \), the Hessian matrix of \( f(\bar{x}) \) is defined as the matrix

\[
H(f(\bar{x})) = \begin{pmatrix}
\frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\
\frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2}
\end{pmatrix}
\]  

(1.4)

If the second derivatives of \( f \) are all continuous, a condition which we shall assume here, the order of differentiation does not matter, and thus the Hessian matrix is symmetric. It is then worthwhile to introduce the condition number of the Hessian, a scalar which characterizes its degree of complexity, and typically determines the difficulty of a problem to be solved by optimization methods. Let \( \{\Lambda_i^H\}_{i=1}^n \) denote the eigenvalues of the Hessian \( H \), and let \( \Lambda_{\min}^H \) and \( \Lambda_{\max}^H \) denote its minimal and maximal eigenvalues, respectively. The condition number of the Hessian matrix is defined by:

\[
\text{cond}(H) = \frac{\Lambda_{\max}^H}{\Lambda_{\min}^H} \geq 1
\]  

(1.5)

Ill-conditioned problems are often classified as such due to large condition numbers (e.g., 10\(^{14}\) of the Hessian on their landscapes.

**Separability** Another defining property of problem difficulty is the separability of the objective function (see, e.g., [11]). A function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is called separable if it can be optimized by solving \( n \) 1-dimensional problems separately:

\[
\arg\min_{\bar{x}} f(\bar{x}) = \left( \arg\min_{x_1} f(x_1, \ldots), \ldots, \arg\min_{x_n} f(\ldots, x_n) \right)
\]

1.1.2 Evolutionary Algorithms

Whereas ES and EP are similar algorithms and share many basic characteristics [12], the principal difference between them and GAs is the encoding of
Chapter 1. Evolution Strategies

Algorithm 1 An Evolutionary Algorithm

1: $t \leftarrow 0$
2: $P_t \leftarrow \text{Init}() \{ P_t \in S^\mu; \text{Set of solutions} \}$
3: Evaluate($P_t$)
4: while $t < t_{\text{max}}$ do
5: $G_t \leftarrow \text{Generate}(P_t) \{ \text{Generate } \lambda \text{ variations} \}$
6: Evaluate($G_t$)
7: $P_{t+1} \leftarrow \text{Select}(G_t \cup P_t) \{ \text{Rank and select } \mu \text{ best} \}$
8: $t \leftarrow t + 1$
9: end while

The genetic information. Traditional GAs encode the genome with discrete values (as in nature), whereas ES as well as EP do that with continuous real-values. Moreover, ES and EP focused more on development of mutation operators, while in classical GA research the recombination operator received most attention. Today, GA, ES, and EP subsume under the term Evolutionary Algorithms (EAs).

Here, we offer an introductory generic description of an EA. The latter considers a population (i.e., set) of individuals (i.e., trial solutions), and models its collective learning process. Each individual in the population is initialized according to an algorithm-dependent procedure, and may carry not only a specific search point in the landscape, but also some environmental information concerning the search. A combination of stochastic as well as deterministic processes such as mutation, recombination, and selection, dictate the propagation in time towards successively better individuals, corresponding to better regimes of the landscape. The quality of an individual, or alternatively the merit of a trial solution, are determined by a so-called fitness function, which is typically the objective function or its rescaling. Thus, certain individuals are favored over others during the selection phase, which is based upon the fitness evaluation of the population. The selected individuals become the candidate solutions of the next generation, while the others die out.

More explicitly, an EA starts with initializing the generation counter $t$. After generating the initial population with $\mu$ individuals in $\mathcal{S}$, a set $G_t$ of $\lambda$ new solutions is generated by means of mutation and possibly recombination. The new candidate solutions are evaluated and ranked in terms of their quality (fitness value). The $\mu$ best solutions in $G_t \cup P_t$ are selected to form the new parent population $P_{t+1}$.

A generalized EA pseudocode is outlined in Algorithm 1.
1.2 The Standard Evolution Strategy

Evolution Strategies were originally developed at the Technical University of Berlin as a procedure for automated experimental design optimization, rather than a global optimizer for continuous landscapes. Following a sequence of successful applications (e.g., shape optimization of a bended pipe, drag minimization of a joint plate, and hardware design of a two-phase flashing nozzle), a diploma thesis [13] and a dissertation [14] laid out the solid foundations for ES as an optimization methodology. There has been extensive work on ES analysis and algorithmic design since then [7, 15, 16].

This section, which is mostly based on [1] and [7], will describe the standard ES in detail. Section 1.2.1 will introduce notation and basic terminology. Section 1.2.2 will present the (1 + 1) algorithm, which was originally analyzed for theoretical purposes, but continued to play an important role in several aspects of Evolution Strategy design. The self-adaptation principle will be described in Section 1.2.3, while Section 1.2.4 will outline the ES algorithm.

1.2.1 Notation and Terminology

The typical application domain of Evolution Strategies is the minimization of non-linear objective functions of signature $f : \mathcal{S} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$. Given a search problem of dimension $n$, let $\vec{x} := (x_1, x_2, ..., x_n)^T \in \mathbb{R}^n$ denote the set of decision parameters or object variables to be optimized: It is defined as an individual associated with a trial solution. In optimization problems, which are of our main interest, it is then straightforward to define the fitness of that individual: It is the objective function(s) value(s) of $\vec{x}$, i.e., $f(\vec{x})$.

Evolution Strategies consider a population of candidate solutions of the given problem. This population undergoes stochastic as well as deterministic variations, with the so-called mutation operator, and possibly with the recombination operator. The mutation operator is typically equivalent to sampling a random variation from a normal distribution. Due to the continuous nature of the parameter space, the biological term mutation rate can be associated here with the actual size of the mutation step in the decision space, also referred to as the mutation strength.

Explicitly, an individual is represented by a tuple of continuous real-values, sometimes referred to as a chromosome, which comprises the decision parameters to be optimized, $\vec{x}$, their fitness value, $f(\vec{x})$, as well as a set of endogenous (i.e., evolvable) strategy parameters, $\vec{s} \in \mathbb{R}^m$.

The $k^{th}$ individual of the population is thus denoted by:

$$\vec{a}_k = (\vec{x}_k, \vec{s}_k, f(\vec{x}_k))$$

The dimension $m$ of the strategy parameter space is subject to the desired parameter control approach, to be discussed shortly. The endogenous pa-
rameters are a unique concept for ES, in particular in the context of the mutation operator, and they play a crucial role in the so-called self-adaptation principle (see Section 1.2.3).

Strategy-specific parameters, such as the population characteristic parameters $\mu$, $\lambda$, and the so-called mixing number $\nu$, are called exogenous strategy parameters, as they are kept constant during the simulated evolution. The mixing number determines the number of individuals involved in the application of the recombination operator.

1.2.2 Motivation: The $(1+1)$ Evolution Strategy

Rechenberg [6] considered a simple $(1+1)$ Evolution Strategy, with a fixed mutation strength $\sigma$, in order to investigate analytically two basic objective functions, namely the corridor model and the sphere model. From the historical perspective, that study laid out the foundations for the theory of Evolution Strategies.

Rechenberg derived explicitly the expressions for the convergence rate of his $(1+1)$ ES for the two models. By definition, neither self-adaptation nor recombination were employed in this strategy. Given the probability of the mutation operator to cover a distance $k'$ towards the optimum, $p(k')$, the convergence rate $\varphi$ is defined as the expectation of the distance $k'$ covered by the mutation:

$$\varphi = \int_0^\infty p(k') \cdot k' \, dk'$$

(1.6)

The expression for the optimal step-size for the two models was first derived. It was observed to depend on the so-called success probability $p_s$,

$$p_s = \mathcal{P} \{ f(Mutate \{ \vec{x} \}) \leq f(\vec{x}) \} .$$

(1.7)

By setting

$$\left. \frac{d\varphi}{d\sigma} \right|_{\sigma^*} = 0,$$

(1.8)

the optimal step-sizes for the two models were calculated, yielding also the optimal success probabilities. The obtained values were both close to $1/5$, regardless of the search space dimensionality. This led to the formulation of the well-known $1/5th$-success rule:

The ratio of successful mutations to all mutations should be $1/5$. If it is greater than $1/5$, increase the standard deviation, if it is smaller, decrease the standard deviation.

For more details see [1]. The implementation of the $1/5th$-success rule within the $(1+1)$-ES is given as Algorithm 2. As practical hints, $p_s$ can be calculated over intervals of $10 \cdot n$ trials, and the adaptation constant should be set between the boundaries $0.817 \leq c \ll 1$. 
Algorithm 2 The \((1 + 1)\) Evolution Strategy

1: \(t \leftarrow 0\)
2: \(P_t \leftarrow \text{Init}()\) \(\{P_t \in S: \text{Set of solutions}\}\)
3: Evaluate\((P_t)\)
4: \textbf{while} \(t < t_{\max}\) \textbf{do}
5: \(\bar{x}(t) := \text{Mutate} \{\bar{x}(t - 1)\}\) with step-size \(\sigma\)
6: Evaluate\((P'(t) := \{\bar{x}(t)\}) : \{f(\bar{x}(t))\}\)
7: Select \(\{P'(t) \cup P(t)\}\)
8: \(t \leftarrow t + 1\)
9: \textbf{if} \(t \mod n = 0\) \textbf{then}
10: \[\sigma = \begin{cases} \sigma(t - n)/c & \text{if } p_s > 1/5 \\ \sigma(t - n) \cdot c & \text{if } p_s < 1/5 \\ \sigma(t - n) & \text{if } p_s = 1/5 \end{cases}\]
11: \textbf{else}
12: \(\sigma(t) = \sigma(t - 1)\)
13: \textbf{end if}
14: \textbf{end while}

It should be noted that \(1/5\text{th}\)-success rule has been kept alive, and continued to play an important role in several aspects, including the construction of the elitist strategy of the Covariance Matrix Adaptation ES algorithm ([17] and also see Section 1.4).

1.2.3 The Self-Adaptation Principle

Section 1.2.2 provided us with the motivation to adapt the endogenous strategy parameters during the course of evolution, e.g., tuning the mutative step-size according to the \(1/5\text{th}\)-success rule. The basic idea of the self-adaptation principle is to consider the strategy parameters as endogenous parameters, that undergo an evolutionary process themselves. The idea of coupling endogenous strategy parameters to the object variables can be found in organisms, where self-repair mechanisms exist, such as repair enzymes and mutator genes [18]. This allows an individual to adapt to the changing environment of its trajectory in the landscape, while keeping the potentially harmful effect of mutation within reasonable boundaries. Hence, when mutative self-adaptation is applied, there is no deterministic control in the hands of the user with respect to the mutation strategy.

The crucial claim regarding ES is that self-adaptation of strategy parameters works [19]. It succeeds in doing so by applying the mutation, recombination and selection operators in the strategy, and without the use of any exogenous control. The link between strategy and decision parameters is exploited, even if it is only indirect. Experiments upon which this claim was
based had found several boosting conditions for self-adaptation to work, such as recombination on strategy parameters, selection pressure within certain bounds, and others.

1.2.4 The Canonical $(\mu/\nu \pm \lambda)$-ES Algorithm

We describe here the specific operators for the standard Evolution Strategy, sometimes referred to as the Schwefel approach, and provide the reader with the implementation details.

\textbf{Mutation}

The mutation operator is the dominant variation operator within ES, and thus we choose to elaborate in this section on its characteristics. As a retrospective analysis, we choose to begin with the outline of some general rules for the design of mutation operators, as suggested by Beyer [15]:

1. \textbf{Reachability}. Given the current generation of individuals, any other search point in the landscape should be reached within a finite number of mutation operations.

2. \textbf{Unbiasedness}. Variation operators in general, and the mutation operator in particular, should not introduce any bias, and satisfy the maximum entropy principle. In the case of continuous unconstrained landscapes, this would suggest the use of the \textit{normal distribution}.

3. \textbf{Scalability}. The mutation strength should be adaptive with respect to the landscape.

The ES mutation operator considers stochastic continuous variations, which are based on the \textit{multivariate normal distribution}. Given a normally-distributed random vector, denoted by $\vec{z} = (z_1, z_2, \ldots, z_n)^T$, the mutation operator is then defined as follows:

$$\vec{x}^{\text{NEW}} = \vec{x}^{\text{OLD}} + \vec{z}$$  \hspace{1cm} (1.9)

A \textit{multivariate normal distribution} is uniquely defined by a \textit{covariance matrix}, $\mathbf{C} \in \mathbb{R}^{n \times n}$, which is a symmetric positive semi-definite matrix, as well as by a \textit{mean vector} $\vec{m} \in \mathbb{R}^n$. Its \textit{probability density function} (PDF) is given by:

$$\Phi^{\text{pdf}}(\vec{z}) = \frac{1}{\sqrt{(2\pi)^n \det \mathbf{C}}} \cdot \exp \left( -\frac{1}{2} (\vec{z} - \vec{m})^T \cdot \mathbf{C}^{-1} \cdot (\vec{z} - \vec{m}) \right)$$ \hspace{1cm} (1.10)

A random vector $\vec{z}$ drawn from a multivariate normal distribution, is denoted by

$$\vec{z} \sim \mathcal{N}(\vec{m}, \mathbf{C}).$$
1.2. The Standard Evolution Strategy

The ES mutation operator always considers a distribution with zero mean, i.e., $\vec{m} = 0$, and thus the covariance matrix $C$ is the defining component of this operator. It is characterized by its $(n \cdot (n - 1)) / 2$ covariance elements,

$$c_{ij} = \text{cov}(x_i, x_j) = \text{cov}(x_j, x_i) = c_{ji},$$

as well as by its $n$ variances,

$$c_{ii} \equiv \sigma_i^2 = \text{var}(x_i).$$

Overall, we have,

$$C = \begin{pmatrix}
\text{var}(x_1) & \text{cov}(x_1, x_2) & \cdots & \text{cov}(x_1, x_n) \\
\text{cov}(x_2, x_1) & \text{var}(x_2) & \cdots & \text{cov}(x_2, x_n) \\
\vdots & \vdots & \ddots & \vdots \\
\text{cov}(x_n, x_1) & \text{cov}(x_n, x_2) & \cdots & \text{var}(x_n)
\end{pmatrix}$$

Essentially, the $(n \cdot (n + 1)) / 2$ independent elements of the covariance matrix are the endogenous strategy parameters that evolve along with the individual:

$$\vec{s} \leftarrow C,$$

i.e., the strategy parameter vector $\vec{s}$ represents the covariance matrix $C$ in this case.

For the definition of the update rule for the strategy parameters, it is convenient to represent the off-diagonal elements of $C$ by means of the rotational angles between the principal axes of the decision parameters. Let $\alpha_{ij}$ denote these angles,

$$c_{ij} = \text{cov}(x_i, x_j) = \frac{1}{2}(\text{var}(x_i) - \text{var}(x_j)) \cdot \tan(2\alpha_{ij}) \quad (1.11)$$

According to the self-adaptation principle, the covariance matrix elements also evolve every generation. The adaptation of the covariance matrix elements is dictated by non-linear update rules: The diagonal terms, $c_{ii} = \sigma_i^2$, are updated according to the log-normal distribution:

$$\sigma_i^{NEW} = \sigma_i^{OLD} \cdot \exp \left( \tau' \cdot N(0, 1) + \tau \cdot N_i(0, 1) \right) \quad (1.12)$$

and the off-diagonal terms are updated through the rotational angles:

$$\alpha_{ij}^{NEW} = \alpha_{ij}^{OLD} + \beta \cdot N_\ell(0, 1) \quad (1.13)$$

where $N(0, 1)$, $N_i(0, 1)$, and $N_\ell(0, 1)$ ($\ell = 1, \ldots, (n \cdot (n - 1)) / 2$) denote independent random variables, and where $\tau \sim 1 / \sqrt{2n}$, $\tau' \sim 1 / \sqrt{2n}$, and $\beta = \frac{5}{3n} \pi$ are constants. After those two update steps, the covariance matrix can be updated (off-diagonal terms are calculated by means of Eq. 1.11).
Geometrical Interpretation  The equal probability density contour lines of a multivariate normal distribution are ellipsoids, centered about the mean. The principal axes of the ellipsoids are defined by the eigenvectors of the covariance matrix $C$. The lengths of the principal axes are proportionate to the corresponding eigenvalues. Figure 1.1 provides an illustration for mutation ellipsoids in the case of $n = 2$.

Correlated Mutations: Strategy Considerations  Given a decision parameter space of dimension $n$, a general mutation-control mechanism considers the covariance matrix $C$, but may apply various different strategies, for computational considerations. There are three common approaches:

1. A covariance matrix proportionate to the identity matrix, i.e., having a single free strategy parameter $\sigma$, often referred to as the global step-size:

$$C_1 = \sigma^2 \cdot I$$

2. A diagonalized covariance matrix, i.e., having a vector of $n$ free strategy parameters, $(\sigma_1^2, \sigma_2^2, ..., \sigma_n^2)^T$, typically referred to as the individual step-sizes:

$$C_2 = \text{diag} (\sigma_1^2, \sigma_2^2, ..., \sigma_n^2)$$
3. A general non-singular covariance matrix, with arbitrary \((n \cdot (n + 1))/2\) free strategy parameters:

\[ C_3 = (c_{ij}) \]  \hspace{1cm} \text{(1.16)}

Thus, the three approaches propose orders of \(O(1)\), \(O(n)\), or \(O(n^2)\) strategy parameters to be learned, respectively, at the cost of different invariance properties. Obviously, a single global step-size approach is very limited in its ability to generate successful moves on a generic landscape. The generalization into individual step-sizes assigns different variances to each coordinate axis, achieving an invariance with respect to translation, but still having dependency on the coordinate system (no invariance with respect to rotation). Finally, the most general approach with an arbitrary normal mutation distribution introduces complete invariance with respect to translation and rotation. Figure 1.2 offers an illustration for the three different approaches, on a given 2D landscape.

Recombination

Inspired by the organic mechanism of a meiotic cell division, where the genetic material is reordered by means of crossover between the chromosomes, the ES recombination operator considers sharing the information from up to \(\nu\) parent individuals [21]. When \(\nu > 2\), it is usually referred to as multi-recombination. Unlike other Evolutionary Algorithms (e.g., GAs), the ES recombination operator obtains only a single offspring.

Due to the continuous nature of the parameters at hand, decision as well as strategy parameters, there are two fundamental ways to recombine
parents:

- **Discrete recombination**: one of the alleles is randomly chosen among \( \nu \) parents. Given a parental matrix of the old generation, \( \mathbf{A}^O = (\tilde{a}_1^O, \tilde{a}_2^O, \ldots, \tilde{a}_\nu^O) \), the new *recombinant* \( \tilde{a}^N \) is constructed by:

\[
(\tilde{a}^N)_i := (\mathbf{A}_{m_i}^O)_i, \quad m_i := \text{rand}\{1, \ldots, \nu\}
\]

- **Intermediate recombination**: the values of \( \nu \) parents are averaged, typically with uniform weights. Essentially, this is equivalent to calculating the centroid of the \( \nu \) parent vectors:

\[
(\tilde{a}^N)_i := \frac{1}{\nu} \sum_{j=1}^{\nu} (\tilde{a}_j^O)_i \tag{1.17}
\]

The recombination operator in the standard ES could be applied as follows:

1. For each *object variable* choose \( \nu \) parents, and apply *discrete recombination* on the corresponding variables.

2. For each *strategy parameter* choose \( \nu \) parents, and apply *intermediate recombination* on the corresponding variables.

It should be noted that there are no generally known best settings of the recombination operator, and the above are typical implementations of it.

Within the GA research, the *building block hypothesis* (BBH) (see, e.g., [22]) offered an explanation for the working mechanism of the crossover: The combination of good, but yet different, building blocks, i.e., specific portions of the genetic encoding from different parents, is supposed to be the key role for propagating high fitness. The debate over this hypothesis has been kept alive. In ES populations, the diversity decreases rapidly. Therefore, BBH is unlikely to fit in a similar way it does in GA populations.

On the other hand, ES research has given rise to the *genetic repair* hypothesis [23], stating that the *common* good properties of the different parents, rather than their different features, are the key role in the working mechanism of recombination. Also, recombination would typically decrease the harmful effect of mutation and would allow for high step-sizes while achieving the same convergence rates.

**Selection**

Natural selection is the driving force of organic evolution: Clearing-out an old generation, and allowing its individuals with the fitness advantage to increase their representation in the genetic pool of future generations. As dramatic as it might sound, *death* is an essential part in this process.
1.2. The Standard Evolution Strategy

Algorithm 3 The $(\mu/\nu + \lambda)$ Evolution Strategy

1: \( t \leftarrow 0 \)
2: \( P_t \leftarrow \text{Init()} \) \( \{ P_t \in S^\mu : \text{Set of solutions} \} \)
3: Evaluate\( (P_t) \)
4: \textbf{while} \( t < t_{\text{max}} \) \textbf{do}
5: \quad \text{Select} \( \nu \) mating parents from \( P_t \) \{Marriage\}
6: \quad \widehat{\alpha}_k(t) := \text{Recombine} \{ P(t) \} \quad \forall k \in \{1, \ldots, \lambda \} \quad \{\text{Recombination} \} \)
7: \quad \widehat{\beta}_k(t) := \text{Mutate} \{ \widehat{\alpha}_k(t) \} \quad \forall k \in \{1, \ldots, \lambda \} \quad \{\text{Mutation} \} \)
8: \quad Evaluate\( (P^n(t) := \{ \widehat{\alpha}^n_1(t), \ldots, \widehat{\alpha}^n_\lambda(t) \}) (\{f (\widehat{x}^n_1(t)), \ldots, f (\widehat{x}^n_\lambda(t))\}) \)
9: \quad \textbf{if} \ (\mu, \lambda)\text{-ES} \ \textbf{then}
10: \quad \text{Select} \{ P^n(t) \}
11: \quad \textbf{else if} \ (\mu + \lambda)\text{-ES} \ \textbf{then}
12: \quad \text{Select} \{ P^n(t) \cup P(t) \}
13: \quad \textbf{end if}
14: \quad t \leftarrow t + 1
15: \quad \textbf{end while}

Evolution Strategies adopt this principle, and employ deterministic operators in order to select the best \( \mu \) individuals with the highest fitness, e.g., minimal objective function values, to be transferred into the next generation. Two selection operators are introduced in the standard ES using an elegant notation due to Schwefel. The notation characterizes the selection mechanism, as well as the number of parents and offspring involved:

- (\( \mu + \lambda \))-selection: the next generation of parents will be the best \( \mu \) individuals selected out of the union of current parents and \( \lambda \) offspring.

- (\( \mu, \lambda \))-selection: the next generation of parents will be the best \( \mu \) individuals selected out of the current \( \lambda \) offspring.

In the case of comma selection, it is rather intuitive that setting \( \mu < \lambda \) would be a necessary condition for an efficient convergence. In plus selection, however, any \( \mu > 0 \) can be chosen in principle. In the latter, the so-called elitist selection occurs, when the survival of the best individual found so far is guaranteed, leading to a possible scenario of a parent surviving for the entire process.

We are now in a position to introduce a pseudocode of the Standard Evolution Strategy (Algorithm 3).

A Note on Population Sizes One of the important topics in ES research is the study of optimal population sizes. By definition, the magnitude of \( \lambda \) determines the number of function evaluations per generation, which should preferably be kept small.
Typical population sizes in ES keep a ratio of $\frac{1}{\lambda}$ between the parent and the offspring populations; a popular choice is $\mu = 15$ and $\lambda = 100$ (see, e.g., [1] and [20]). Based on experimental observations, when individual step-sizes are chosen as strategy parameters (Eq. 1.15), $\lambda$ has to scale linearly with $n$. In the case of arbitrary normal mutations (Eq. 1.16), Rudolph [24] showed that successful adaptation to the landscape (i.e., learning successfully the Hessian matrix) can be achieved with an upper bound of $\mu + \lambda = (n^2 + 3n + 4)/2$, but it is certainly not likely to be achieved with the typical population sizes of $\{\mu = 15, \lambda = 100\}$.

1.3 Derandomized Evolution Strategies (DES)

Mutative step-size control (MSC) tends to work well in the Standard-ES for the adaptation of a single global step-size (Eq. 1.14), but tends to fail when it comes to the individual step-sizes or arbitrary normal mutations (Eq. 1.15 or Eq. 1.16). Schwefel claimed that the adaptation of the strategy parameters in those cases is impossible within small populations [19], and suggested larger populations as a solution to the problem.

Due to the crucial role that the mutation operator plays within Evolution Strategies, its mutative step-size control was investigated intensively. In particular, the disruptive effects to which the MSC is subject, were studied at several levels [25, 16], and are reviewed here:

- **Indirect selection.** By definition, the goal of the mutation operator is to apply a stochastic variation to an object variable vector, which will increase its selection probability. The selection of the strategy parameters setting is indirect, i.e., the vector of a successful mutation is not used to adapt the step-size parameters, but rather the parameters of the distribution that led to this mutation vector.

- **Realization of parameter variation.** Due to the sampling from a random distribution, the realization of the parameter variation does not necessarily reflect the nature of the strategy parameters. Thus, the difference de facto between good and bad strategy settings of strategy parameters is only reflected in the difference between their probabilities to be selected - which can be rather small. Essentially, this means that the selection process of the strategy parameters is strongly disturbed.

- The strategy parameter change rate is defined as the difference between strategy parameters of two successive generations. Hansen and Ostermeier [16] argue that the change rate is an important factor, as it gives an indication concerning the adaptation speed, and thus it has a direct influence on the performance of the algorithm. The principal claim is that this change rate basically vanishes in the standard-ES.
1.3. Derandomized Evolution Strategies (DES)

The change rate depends on the mutation strength to which the strategy parameters are subject. While aiming at attaining the maximal change rate, the latter is underposed to an upper bound, due to the finite selection information that can be transferred between generations. Change rates that exceed the upper bound would lead to a stochastic behavior. Moreover, the mutation strength that obtains optimal change rate is typically smaller than the one that obtains good diversity among the mutants - a desired outcome of the mutation operator, often referred to as selection difference. Thus, the conflict between the objective of optimal change rate versus the objective of optimal selection difference cannot be resolved at the mutation strength level [25]. A possible solution to this conflict would be to unlink the change rate from the mutation strength.

The so-called derandomized mutative step-size control aims to treat those disruptive effects, regardless of the problem dimensionality, population size, etc.

1.3.1 (1, λ) Derandomized ES Variants

The concept of derandomized Evolution Strategies has been originally introduced by scholars at the Technical University of Berlin in the beginning of the 1990’s. It was followed by the release of a new generation of successful ES variants by Hansen, Ostermeier, and Gawelczyk [26, 27, 28, 29].

The first versions of derandomized ES algorithms introduced a controlled global step-size in order to monitor the individual step-sizes by decreasing the stochastic effects of the probabilistic sampling. The selection disturbance was completely removed with later versions by omitting the adaptation of strategy parameters by means of probabilistic sampling. This was combined with individual information from the last generation (the successful mutations, i.e., of selected offspring), and then adjusted to correlated mutations. Later on, the concept of adaptation by accumulated information was introduced, aiming to use wisely the past information for the purpose of step-size adaptation: Instead of using the information from the last generation only, it was successfully generalized to a weighted average of the previous generations.

Note that the different derandomized-ES variants strictly follow a (1, λ) strategy, postponing the treatment of recombination or plus-strategies for later stages\(^1\). In this way, the question how to update the strategy parameters when an offspring does not improve its ancestor is not relevant here.

Moreover, the different variants hold different numbers of strategy parameters to be adapted, and this is a factor in the learning speed of the

\(^1\)When asked about comma versus plus strategies, Hansen states that “with a good enough algorithm at hand, employing the plus strategy is unnecessary, as your algorithm should be able to revisit the best attainable solution”.
optimization routine. The different algorithms hold a number of strategy parameters scaling either linearly ($O(n)$ parameters responsible for individual step-sizes) or quadratically ($O(n^2)$ parameters responsible for arbitrary normal mutations) with the dimensionality $n$ of the search space.

1.3.2 First Level of Derandomization

The so-called first level of derandomization achieved the following desired effects:

- A degree of freedom with respect to the mutation strength of the strategy parameters.
- Scalability of the ratio between the change rate and the mutation strength.
- Independence of population size with respect to the adaptation mechanism.

We choose to review the implementation of the first level of derandomization through three particular derandomized ES variants:

DR1

The first derandomized attempt [26] coupled the successful mutations to the selection of decision parameters, and learned the mutation step-size as well as the scaling vector based upon the successful variation. The mutation step is formulated for the $k^{th}$ individual, $k = 1, \ldots, \lambda$

$$x^{(g+1)} = x^{(g)} + \xi_k \delta^{(g)} \eta_\text{scal} \tilde{z}_k$$

$$\tilde{z}_k \in \{-1, +1\}^n$$ \hspace{1cm} (1.18)

Note that $\tilde{z}_k$ is a random vector of ±1, rather than a normally distributed random vector, while $\xi_k \sim \mathcal{N}(0,1)$, i.e., distributed over the positive part of the normal distribution. The evaluation and selection are followed by the adaptation of the strategy parameters (subscripts $\text{sel}$ refer to the selected individual):

$$\delta^{(g+1)} = \delta^{(g)} \cdot (\xi_\text{sel})^{\beta}$$ \hspace{1cm} (1.19)

$$\delta_\text{scal}^{(g+1)} = \delta_\text{scal}^{(g)} \cdot (\xi_\text{scal}^{\text{sel}} + b)^{\beta_\text{scal}}$$ \hspace{1cm} (1.20)

$\mathcal{P} \left( \xi_k = \frac{7}{3} \right) = \mathcal{P} \left( \xi_k = \frac{5}{7} \right) = \frac{1}{2}$; $\beta = \sqrt{1/n}$; $\beta_\text{scal} = 1/n$, $b = 0.35$, and $\xi_k \in \left\{ \frac{7}{3}, \frac{5}{7} \right\}$ are constants. Note that the multiplication in Eq. 1.20 is between two vectors and carried out as element-by-element multiplication, yielding a vector of the same dimension $n$.  

1.3. Derandomized Evolution Strategies (DES)

DR2

The second derandomized ES variant [27] aimed to accumulate information about the correlation or anti-correlation of past mutation vectors in order to adapt the global step-size as well as the individual step-sizes - by introducing a quasi-memory vector. This accumulated information allowed omitting the stochastic element in the adaptation of the strategy parameters - updating them only by means of successful variations, rather than with random steps.

The mutation step for the \( k \)-th individual, \( k = 1, \ldots, \lambda \), reads:

\[
\vec{x}^{(g+1)} = \vec{x}^{(g)} + \delta^{(g)} \delta_{\text{scal}} \vec{z}_k \\
\vec{z}_k \sim \mathcal{N}(0, 1) \tag{1.21}
\]

Introducing a quasi-memory vector \( \vec{Z} \):

\[
\vec{Z}^{(g)} = c \vec{z}_{\text{sel}} + (1-c) \vec{Z}^{(g-1)} \tag{1.22}
\]

The adaptation of the strategy parameters according to the selected offspring:

\[
\delta^{(g+1)} = \delta^{(g)} \cdot \left( \exp \left( \frac{\|\vec{Z}^{(g)}\|}{\sqrt{n} \sqrt{c} } - 1 + \frac{1}{5n} \right) \right)^{\beta} \tag{1.23}
\]

\[
\delta_{\text{scal}}^{(g+1)} = \delta_{\text{scal}}^{(g)} \cdot \left( \frac{\|\vec{Z}^{(g)}\|}{\sqrt{\frac{c}{2-c} } } + b \right)^{\beta_{\text{scal}}} \\
\|\vec{Z}^{(g)}\| = (|Z_1^{(g)}|, |Z_2^{(g)}|, \ldots, |Z_n^{(g)}|) \tag{1.24}
\]

with \( \beta = \sqrt{1/n} \), \( \beta_{\text{scal}} = 1/n \), \( b = 0.35 \), and the quasi-memory rate \( c = \sqrt{1/n} \) as constants. Note that the multiplication in Eq. 1.24 is between two vectors and carried out as element-by-element multiplication, yielding a vector of the same dimension \( n \).

DR3

This third variant [28], usually referred to as the Generation Set Adaptation (GSA), considered the derandomization of arbitrary normal mutations for the first time, aiming to achieve invariance with respect to the scaling of variables and the rotation of the coordinate system. This naturally came with the cost of a quasi-memory matrix, \( B \in \mathbb{R}^{n \times n} \), setting the dimension of the strategy parameters space to \( n^2 \leq m \leq 2n^2 \). The adaptation of the global step-size is mutative with stochastic variations, just like in the DR1.

The mutation step is formulated for the \( k \)-th individual, \( k = 1, \ldots, \lambda \):

\[
\vec{x}^{(g+1)} = \vec{x}^{(g)} + \delta^{(g)} \xi_k \vec{y}_k \tag{1.25}
\]

\[
\vec{y}_k = c_m B^{(g)} \cdot \vec{z}_k \\
\vec{z}_k \sim \mathcal{N}(0, 1) \tag{1.26}
\]
The update of the memory matrix is formulated as:
\[
B^{(g)} = \begin{pmatrix} \overline{b}^{(g)}_1, \ldots, \overline{b}^{(g)}_m \end{pmatrix}
\]
\[
\overline{b}^{(g+1)}_i = (1 - c) \cdot \overline{b}^{(g)}_i + c \cdot (c_u \xi_{sel} \overline{y}_{sel}) , \quad \overline{b}^{(g+1)}_{i+1} = \overline{b}^{(g)}_i
\]  \hspace{1cm} (1.27)

The step-size is updated as follows:
\[
\delta^{(g+1)} = \delta^{(g)} (\xi_{sel})^\beta
\]  \hspace{1cm} (1.28)

where \( \mathcal{P} (\xi_k = \frac{2}{3}) = \mathcal{P} (\xi_k = \frac{3}{4}) = \frac{1}{2} ; \beta = \sqrt{1/n} , \ c_m = (1/\sqrt{m})(1 + 1/m) , \ c = \sqrt{1/n} , \ \xi_k \in \{ \frac{3}{4}, \frac{2}{3} \} , \) and \( c_u = \sqrt{(2 - c)/c} \) are constants.

### 1.4 The Covariance Matrix Adaptation ES

Following a series of successful derandomized ES variants addressing the first level of derandomization, and a continuous effort at the Technical University of Berlin, the so-called Covariance Matrix Adaptation (CMA) Evolution Strategy was released in 1996 [29], as a completely derandomized Evolution Strategy - the fourth generation of derandomized ES variants.

**Second Level of Derandomization** The so-called second level of derandomization targeted the following effects:

- The probability to regenerate the same mutation step is increased.
- The change rate of the strategy parameters is subject to explicit control.
- Strategy parameters are stationary when subject to random selection.

The second level of derandomization was implemented by means of the CMA.

The CMA combines the robust mechanism of ES with powerful statistical learning principles, and thus it is sometimes subject to informal criticism for not being a genuine Evolution Strategy. In short, it aims at satisfying the maximum likelihood principle by applying Principle Components Analysis (PCA) to the successful mutations, and it uses cumulative global step-size adaptation.

### 1.4.1 Preliminary

One of the goals of the CMA is to achieve a successful statistical learning process of the optimal mutation distribution, which is equivalent to learning a covariance matrix proportional to the inverse of the Hessian matrix (see, e.g., [30]), without calculating the actual derivatives:

\[
C \propto H^{-1}
\]
1.4. The Covariance Matrix Adaptation ES

Rather than representing a mutation step with a normal variation with zero mean (Eq. 1.9), it is convenient to refer to the original notation of the normal distribution. Thus, in the notation we use here, the vector \( \vec{m} \) represents the mean of the mutation distribution, but is also associated with the favorite solution at present (i.e., \( \vec{x}^{\text{OLD}} \) of Eq. 1.9), \( \sigma \) denotes the global step-size, and the covariance matrix \( \mathbf{C} \) determines the shape of the distribution ellipsoid:

\[
\vec{x}^{\text{NEW}} \sim \mathcal{N}(\vec{m}, \sigma^2 \mathbf{C}) = \vec{m} + \sigma \cdot \mathcal{N}(\vec{0}, \mathbf{C}) = \vec{m} + \sigma \cdot \vec{z}
\]

Different principles dictate the adaptation of the covariance matrix, \( \mathbf{C} \), versus the adaptation of the global step-size \( \sigma \):

- The mean \( \vec{m} \) and the covariance matrix \( \mathbf{C} \) of the normal distribution are updated according to the maximum likelihood principle, such that good mutations are likely to appear again. \( \vec{m} \) is updated such that

\[
P(\vec{x}_{\text{sel}}|\mathcal{N}(\vec{m}, \sigma^2 \mathbf{C})) \rightarrow \max
\]

and \( \mathbf{C} \) is updated such that

\[
P\left(\frac{\vec{x}_{\text{sel}} - \vec{m}_{\text{old}}}{\sigma} \bigg| \mathcal{N}(\vec{0}, \mathbf{C})\right) \rightarrow \max
\]

considering the prior \( \mathbf{C} \). This is implemented through the so-called Covariance Matrix Adaptation (CMA) mechanism.

- \( \sigma \) is updated such that it is conjugate perpendicular to the consecutive steps of \( \vec{m} \). This is implemented through the so-called Cumulative Step-size Adaptation (CSA) mechanism.

The Evolution Path

The most intuitive way to update the covariance matrix would be to construct an \( n \times n \) matrix analogue to the DR2 mechanism (see Eq. 1.22), with the outer-product of the selected mutation vector \( \vec{z}_{\text{sel}} \):

\[
\mathbf{C} \leftarrow (1 - c_{\text{cov}}) \mathbf{C} + c_{\text{cov}} \vec{z}_{\text{sel}} \vec{z}_{\text{sel}}^T
\]

However, the sign information of \( \vec{z}_{\text{sel}} \) is lost due to \( \vec{z}_{\text{sel}} \vec{z}_{\text{sel}}^T = -\vec{z}_{\text{sel}} (-\vec{z}_{\text{sel}})^T \). The solution lies within the definition of the so-called evolution path, which accumulates the history information using an exponentially weighted moving average:

\[
\vec{p}_c \propto \sum_{i=0}^{g} (1 - c_c)^{g-i} \vec{z}_{\text{sel}}^{(i)}
\]

And now the covariance matrix adaptation step reads:

\[
\mathbf{C} \leftarrow (1 - c_{\text{cov}}) \mathbf{C} + c_{\text{cov}} \vec{p}_c \vec{p}_c^T
\]
The Path Length Control

The covariance matrix update is not likely to increase the variance in all directions simultaneously, and thus a global step-size control is much needed. The basic idea of the so-called path length control is to measure the length of the evolution path, which is also the consecutive steps of \( \vec{m} \), and adapt the step-size according to the following argument: If the evolution path is longer than expected, the steps are likely parallel, and thus the step-size should be increased; Alternatively, if it is shorter than expected, the steps are probably anti-parallel, and the step-size should be decreased accordingly. The expected length is defined in a straightforward manner as the expected length of a normally distributed random vector.

The actual measurement is done by means of the "conjugate" evolution path:

\[
\vec{p}_\sigma \propto \sum_{i=0}^{g} (1 - c_\sigma)^{g-i} \mathbf{C}^{(i)} - \frac{1}{2} \tilde{z}^{(i)}_{sel}
\]

where the factorization of \( \mathbf{C} \) is required in order to align all directions within the \( \text{rotated frame} \). Then, the update of the step-size depends on the comparison between \( \| \vec{p}_\sigma \| \) and the expected length of a normally distributed random vector, \( E [ \| \mathcal{N} (0, \mathbf{I}) \| ] \):

\[
\sigma \leftarrow \sigma \cdot \exp \left( \frac{\| \vec{p}_\sigma \|}{E [ \| \mathcal{N} (0, \mathbf{I}) \| ]} - 1 \right)
\]

### 1.4.2 The \((1, \lambda)\) Rank-One CMA

We are now in a position to introduce the explicit formulation of the rank-one update with cumulation Covariance Matrix Adaptation Evolution Strategy, following the notation introduced in Section 1.4.1. Additionally, consider the diagonalization of the covariance matrix, denoted by

\[
\mathbf{C}^{(g)} = \mathbf{B}^{(g)} \mathbf{D}^{(g)} \left( \mathbf{B}^{(g)} \mathbf{D}^{(g)} \right)^T
\]

where \( \mathbf{B}^{(g)} \) is an orthonormal rotation matrix which defines the coordinate system, and \( \mathbf{D}^{(g)} = \text{diag} \left( \sqrt{\Lambda_1}, \sqrt{\Lambda_2}, ..., \sqrt{\Lambda_n} \right) \) holds the square-roots of the eigenvalues.

The mutation step for the \( k \)-th individual, \( k = 1, \ldots, \lambda \), is then defined as:

\[
\vec{x}_k^{(g+1)} = \vec{x}^{(g)} + \sigma^{(g)} \mathbf{B}^{(g)} \mathbf{D}^{(g)} \tilde{z}_k^{(g+1)}
\]

with \( \tilde{z}_k \sim \mathcal{N} \left( \vec{0}, \mathbf{I} \right) \).

The evolution path, initialized \( \vec{p}_c^{(0)} = \vec{0} \), is explicitly updated as follows:

\[
\vec{p}_c^{(g+1)} = (1 - c_c) \cdot \vec{p}_c^{(g)} + \sqrt{c_c (2 - c_c)} \cdot \mathbf{B}^{(g)} \mathbf{D}^{(g)} \tilde{z}_{sel}^{(g+1)}
\]
and then the covariance matrix, initialized as identity $C^{(0)} = I$, is adapted accordingly:

$$
C^{(g+1)} = (1 - c_{cow}) \cdot C^{(g)} + c_{cow} \cdot \tilde{p}_c^{(g+1)} \left( \tilde{p}_c^{(g+1)} \right)^T
$$  \hspace{1cm} (1.32)

The calculation of the "conjugate" evolution path, initialized $\tilde{p}_c^{(0)} = \overline{0}$, reads:

$$
\tilde{p}_c^{(g+1)} = (1 - c_n) \cdot \tilde{p}_c^{(g)} + \sqrt{c_n(2 - c_n)} \cdot B^{(g)} \tilde{z}_{sel}^{(g+1)}
$$  \hspace{1cm} (1.33)

and then followed by the update of the global step-size:

$$
\sigma^{(g+1)} = \sigma^{(g)} \cdot \exp \left( \frac{c_n}{d_n} \cdot \left( \frac{\|\tilde{p}_c^{(g+1)}\|}{E[\|N(0, I)\|]} - 1 \right) \right)
$$  \hspace{1cm} (1.34)

The various learning coefficients are typically set as $c_n = 4/(n + 4), c_{cow} = 2/(n + 1)^2, c_n = 3/(n + 4)$, and $d_n = 1 + c_n$. The expectation of the length of a normally distributed random vector is given by:

$$
E[\|N(0, I)\|] = \sqrt{2} \cdot \frac{\Gamma \left( \frac{n + 1}{2} \right)}{\Gamma \left( \frac{n}{2} \right)}
$$  \hspace{1cm} (1.35)

where the Gamma function is defined by:

$$
\Gamma(n) = \int_0^\infty x^{n-1} \exp(-x) dx
$$  \hspace{1cm} (1.36)

but may also be approximated by $E[\|N(0, I)\|] \approx \sqrt{n} \left( 1 - \frac{1}{4n} + \frac{1}{2n^2} \right)$.

**Implementation** Additional implementation remarks are outlines here:

- Arnold offered\(^2\) a dramatic simplification to the global step-size update (Eq. 1.34) with replacing $\left( \frac{\|\tilde{p}_c^{(g+1)}\|}{E[\|N(0, I)\|]} - 1 \right)$ by $\left( \frac{\|\tilde{p}_c^{(g+1)}\|^2}{2n} \right)$. This was reported to perform equally well [16].

- The update of the *evolution path* (Eq. 1.31) is usually implemented with a conditional threshold as follows:

$$
\tilde{p}_c^{(g+1)} = (1 - c_n) \cdot \tilde{p}_c^{(g)} + H^{(g+1)} \sqrt{c_n(2 - c_n)} \cdot B^{(g)} \tilde{z}_{sel}^{(g+1)}
$$  \hspace{1cm} (1.37)

$$
H^{(g+1)} = \begin{cases} 
1 & \text{if} \quad \frac{\|\tilde{p}_c^{(g+1)}\|}{\sqrt{1 - (1 - c_n)^2}} < H_{thresh} \\
0 & \text{otherwise}
\end{cases}
$$  \hspace{1cm} (1.38)

where $H_{thresh} = \left( 1.5 + \frac{1}{n - 0.5} \right) \cdot E[\|N(0, I)\|]$.

\(^2\)Hansen et al. cite this source of information as *personal communications.*
1.4.3 The \((\mu_W, \lambda)\) Rank-\(\mu\) CMA

The Rank-\(\mu\) Covariance Matrix Adaptation [31] is an extension of the original update rule for larger population sizes. The idea is to use \(\mu > 1\) vectors in order to update the covariance matrix \(\mathbf{C}\) in each generation, based on weighted intermediate recombination.

Let \(\bar{x}_{i,\lambda}\) denote the \(i^{th}\) ranked solution point, such that

\[
f(\bar{x}_{1,\lambda}) \leq f(\bar{x}_{2,\lambda}) \leq \cdots \leq f(\bar{x}_{\lambda,\lambda})
\]

The updated mean is now defined as follows:

\[
\bar{m} \leftarrow \sum_{i=1}^{\mu} w_i \bar{x}_{i,\lambda} = \bar{m} + \sigma \sum_{i=1}^{\mu} w_i \bar{z}_{i,\lambda} \equiv \langle \bar{x} \rangle_W
\]

with a set of weights:

\[
w_1 \geq w_2 \geq \cdots \geq w_\mu > 0, \quad \sum_{i=1}^{\mu} w_i = 1
\]

Essentially, this is a generalization of the intermediate recombination concept (Eq. 1.17), suggested by Rechenberg\(^3\).

By setting \(\forall i : w_i = \frac{1}{\mu}\), the original recombination is restored, which is then noted by \((\mu_I, \lambda)\) (note, however, that the \((\mu/\mu_I, \lambda)\) notation is also used [32]).

The covariance matrix update can now be formalized by means of rank-\(\mu\) update, using an outer product of the weighted mutation vectors:

\[
\mathbf{C} \leftarrow (1 - c_{\text{cov}})\mathbf{C} + c_{\text{cov}} \sum_{i=1}^{\mu} w_i \bar{z}_{i,\lambda} \bar{z}_{i,\lambda}^T
\]

It can be even furthermore combined with the rank-one update:

\[
\mathbf{C} \leftarrow (1 - c_{\text{cov}})\mathbf{C} + \frac{c_{\text{cov}}}{\mu_{\text{cov}}} \bar{p}_{C} \bar{p}_{C}^T + c_{\text{cov}} \left(1 - \frac{1}{\mu_{\text{cov}}}\right) \sum_{i=1}^{\mu} w_i \bar{z}_{i,\lambda} \bar{z}_{i,\lambda}^T
\]

We shall now present the \((\mu_W, \lambda)\) rank-\(\mu\) CMA characteristic equations:

\[
\bar{x}_{k}^{(g+1)} = \langle \bar{x} \rangle_W^{(g)} + \sigma^{(g)}\mathbf{B}^{(g)}\mathbf{D}^{(g)}\bar{z}_{k}^{(g+1)}, \quad k = 1, \ldots, \lambda
\]

\[
\bar{p}_{C}^{(g+1)} = (1 - c_c) \cdot \bar{p}_C^{(g)} + \sqrt{c_c(2 - c_c) \cdot c_W}\mathbf{B}^{(g)}\mathbf{D}^{(g)}\langle \bar{z} \rangle_W^{(g+1)}
\]

\[
\mathbf{C}^{(g+1)} = (1 - c_{\text{cov}})\mathbf{C}^{(g)} + \frac{c_{\text{cov}}}{\mu_{\text{cov}}^2} \bar{p}_{C}^{(g+1)} \cdot \bar{p}_{C}^{(g+1)} \cdot ^T + c_{\text{cov}} \left(1 - \frac{1}{\mu_{\text{cov}}}\right) \sum_{i=1}^{\mu} w_i \bar{z}_{i,\lambda} \bar{z}_{i,\lambda}^T
\]

\(^3\)Reported as personal communications between Hansen, Ostermeier and Rechenberg.
\[ \vec{p}_g^{(g+1)} = (1 - c) \cdot \vec{p}_g^{(g)} + \sqrt{c(2 - c)} \cdot \vec{B}\left(D^{(g)}\right)^{-1} T \cdot c W \cdot \tilde{z}_W^{(g+1)} \]  
\[ \sigma^{(g+1)} = \sigma^{(g)} \cdot \exp \left( \frac{c \cdot \left\| \vec{p}_g^{(g+1)} \right\|}{E\left[\left\| N(0, I) \right\|\right] - 1} \right) \]  

The weights are typically set to:
\[ w_{i=1,..,\mu} = \frac{\ln (\mu + 1) - \ln (i)}{\sum_{j=1}^{\mu} \ln (\mu + 1) - \ln (j)} \]  

The constant \( c W \) is defined such that \( c W \tilde{z}_W^{(g+1)} \) and \( \tilde{z}_k^{(g+1)} \) are identically distributed with the same variance under random selection:
\[ c W = \frac{\sum_{i=1}^{\mu} w_i}{\sqrt{\sum_{i=1}^{\mu} w_i^2}} \]  

The special rank-\( \mu \) constant, \( \mu_{\text{cov}} \), is the variance effective selection mass:
\[ \mu_{\text{cov}} = \frac{1}{\sum_{i=1}^{\mu} w_i^2} \]  

which becomes \( \mu_{\text{cov}} = \mu \) in the special case of \((\mu_1, \lambda)\).

The rest of the constants are set as in the \((1, \lambda)\) rank-one CMA.

**Population Size**  
Given a search space of dimension \( n \), the default CMA population sizes introduced a *revolutionary order of magnitude* into the ES field, \( \mathcal{O}(\log (n)) \), especially when we take into account the goal to learn the full covariance matrix of the decision parameters space.

The explicit suggested values are as follows:
\[ \lambda = 4 + [3 \cdot \ln (n)] \quad \mu = [\lambda/2] \]  

### 1.4.4 The \((1 + \lambda)\) CMA

This elitist version [17] of the CMA-ES algorithm, which had been originally derived for the sake of a *multi-objective* CMA algorithm [33], combined the classical concept of the \((1 + 1)\) ES strategy, and in particular the *success probability* and *success rule* components (see Eq. 1.7 as well as Section 1.2.2), with the Covariance Matrix Adaptation concept. The so-called *success rule based step size control* replaces the *path length control* of the CMA-comma strategy. The same notation as in Section 1.4.2 is used here:
\[ \tilde{x}_k^{(g+1)} = \tilde{x}^{(g)} + \sigma^{(g)} \vec{B}^{(g)} \cdot \vec{D}^{(g)} \cdot \tilde{z}_k^{(g+1)} \quad k = 1, \ldots, \lambda \]  

\[ \vec{p}_g^{(g+1)} = (1 - c) \cdot \vec{p}_g^{(g)} + \sqrt{c(2 - c)} \cdot \vec{B}^{(g)} \left(\vec{D}^{(g)}\right)^{-1} T \cdot \sigma^{(g+1)} \cdot \exp \left( \frac{c \cdot \left\| \vec{p}_g^{(g+1)} \right\|}{E\left[\left\| N(0, I) \right\|\right] - 1} \right) \]  

\[ c W = \frac{\sum_{i=1}^{\mu} w_i}{\sqrt{\sum_{i=1}^{\mu} w_i^2}} \]  

\[ \mu_{\text{cov}} = \frac{1}{\sum_{i=1}^{\mu} w_i^2} \]  

\[ \lambda = 4 + [3 \cdot \ln (n)] \quad \mu = [\lambda/2] \]  

The rest of the constants are set as in the \((1, \lambda)\) rank-one CMA.
After the evaluation of the new generation, the success rate is updated

\[ p_{\text{succ}} = \frac{\lambda^{g+1}}{\lambda}, \]

where:

\[ \bar{p}_{\text{succ}} = (1 - c_p) \cdot \bar{p}_{\text{succ}} + c_p \cdot p_{\text{succ}} \]

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

The covariance matrix is updated only if the selected offspring is better than the parent. Then,

\[ \tilde{p}_c = \begin{cases} (1 - c_c) \tilde{p}_c + \sqrt{c_c (2 - c_c)} \cdot \frac{\bar{p}_{\text{target}}}{\sigma^{(g)}_{\text{parent}}} & \text{if } \bar{p}_{\text{succ}} < p_\Theta \\ (1 - c_c) \tilde{p}_c & \text{otherwise} \end{cases} \]

\[ C^{(g+1)} = \begin{cases} (1 - c_{\text{cov}}) \cdot C^{(g)} + c_{\text{cov}} \cdot \tilde{p}_c \tilde{p}_c^T & \text{if } \bar{p}_{\text{succ}} < p_\Theta \\ (1 - c_{\text{cov}}) \cdot C^{(g)} + c_{\text{cov}} \cdot (\tilde{p}_c \tilde{p}_c^T + c_c (2 - c_c) C^{(g)}) & \text{otherwise} \end{cases} \]

The default parameters are set as follows:

- \( d = 1 + \frac{n}{2} \), \( p_{\text{target}} = \frac{2}{n}, c_p = \frac{1}{12} \), \( c_c = \frac{2}{n+2} \), \( c_{\text{cov}} = \frac{2}{n^2+6} \), and \( p_\Theta = 0.44 \).

**A Note on Usage** As mentioned earlier, this plus-strategy version was constructed for multi-objective optimization. Unofficially, it is not recommended to use it otherwise. In this work, we will restrict the use of the CMA+ to the niching framework exclusively, and thus we will not consider it upon the employment of the DES variants to single-criterion Quantum Control optimization tasks in Chapter 7.

1.4.5 Constraints Handling

The broad topic of *constraints handling* [34] is certainly not of a major concern in this study, but it does have an indirect impact on the niching techniques to be introduced here, as will become more clear in the following chapters. We thus choose to specify here, in short, the general approach to handle constraints when derandomized-ES are in use, in light of the rule of thumb suggested by Hansen and Ostermeier for the CMA (see [16], pp. 21).

A possible way to handle constraints would be to repeat the generation step (e.g., Eq. 1.30) until \( \lambda \), or at least \( \mu \) feasible solutions are generated. This should be strictly enforced, before the following update equations are applied. It is claimed that this method should perform in a satisfying manner, if a sufficient number of feasible solutions are initially generated - due to the symmetry of the mutation distribution. However, if the global minimum is located at the edge of the feasible domain, it is suggested that other constraints handling techniques should be used.
1.4.6 Discussion

The Covariance Matrix Adaptation Evolution Strategy is a state-of-the-art optimization routine, which combines classical deterministic concepts (e.g., Hessian or Covariance matrices learning) and statistical learning tools (e.g., Principal Components Analysis) with the powerful stochastic mechanism of Evolution Strategies. In terms of standard performance criteria, it was ranked as the best Evolutionary Algorithm at hand [35].

The CMA-ES has been informally criticized for not being a genuine evolution strategy, since it incorporates those non-evolutionary components. Even as such, and despite its considerable success-rate as a global optimizer, we would like to stress that it certainly has a nature of a local search routine. The fact that it learns a unimodal distribution in the search space - no matter how well it does so - makes it a local search. We believe that this provides us with some motivation to use the CMA-ES, as well as other derandomized-ES routines, as algorithmic kernels for a multi-distribution approach - which would construct a niching algorithm. The idea would be essentially to use multiple CMAs in parallel, aiming to achieve a good coverage of the landscapes with local-searchers. This idea would become more clear in the next chapter, when we introduce the gateway to niching.