Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation

Nikolaus Hansen & Andreas Ostermeier

Fachgebiet für Bionik und Evolutionstechnik

Technische Universität Berlin

Ackerstr. 71–76, 13355 Berlin, Germany

E-mail: {hansen,ostermeier}@fb10.tu-berlin.de

In: *Proceedings of the 1996 IEEE Intern. Conf. on Evolutionary Computation (ICEC '96)*: 312–317

Erratum: p.314, right column: $c_{\mathrm{u}} = \sqrt{c \cdot (2 - c)}$

Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation

Nikolaus Hansen & Andreas Ostermeier
Fachgebiet für Bionik und Evolutionstechnik
Technische Universität Berlin
Ackerstr. 71–76, 13355 Berlin, Germany
E-mail: {hansen,ostermeier}@fb10.tu-berlin.de

***Abstract*—A new formulation for coordinate system independent adaptation of arbitrary normal mutation distributions with zero mean is presented. This enables the evolution strategy (ES) to adapt the correct scaling of a given problem and also ensures invariance with respect to any rotation of the fitness function (or the coordinate system). Especially rotation invariance, here resulting directly from the coordinate system independent adaptation of the mutation distribution, is an essential feature of the ES with regard to its general applicability to complex fitness functions. Compared to previous work on this subject, the introduced formulation facilitates an interpretation of the resulting mutation distribution, making sensible manipulation by the user possible (if desired). Furthermore it enables a more effective control of the *overall* mutation variance (expected step length).**

***KeyWords*—Adaptation, covariance matrix, derandomized adaptation, evolutionary algorithms, evolution strategy, mutation distribution, self-adaptation, strategy parameters.**

I. Introduction

Self-adaptation of the mutation distribution is an important feature in evolution strategies (ESs). Without changing the mutation distribution over the generation sequence, no adequate progress can be expected in general.[1] To carry out a self-adaptation process, we assume no other problem-specific knowledge than revealed by selection. **Selection information** consists of all points so far selected in object parameter space.[2] If self-adaptation is applied (cf. e.g. [8; 7; 1]), it is usually assumed implicitly, that it is useful to favour reproduction of successful (i.e. selected) mutation steps in the future — at first glance a quite reasonable approach.

We suggest not to look at *single* mutation steps only, but to consider a path the population takes over a number of generations. We will call such a path an **evolution path** of the ES. Using this information for step size adaptation was proposed in [4], and it was consistently found to be useful for the adaptation of the mutation distribution in our research.

The evolution path mainly reveals information on *correlations* between mutation steps successively selected in the generation sequence. If successively selected mutation steps are parallel correlated (scalar product greater zero), the evolution path will be comparatively long. If successively selected mutation steps are anti-parallel correlated (scalar product less than zero), the evolution path will be comparatively short. Roughly speaking, parallel correlation means that successive steps are going into the same direction, and thus the same distance could be covered by fewer but longer steps. Anti-parallel correlation means, that the steps cancel each other out. Both is inefficient with respect to the single mutation step. Consequently, to make single mutation steps most efficient, it is best to have no correlation between the selected mutation steps in the evolution path. It is important to notice that exactly the same reasoning holds for any given *projection* of mutation steps and evolution path as well. Projections may relate to certain strategy parameters, e.g. to individual step sizes.

The geometrical interpretation is, that successively selected mutation steps should be perpendicular to each other (apart from stochastic deviations).

To remove correlations between successively selected mutation steps, we suggest a **fundamental adaptation principle** for the adaptation of the mutation distribution: Reasonable adaptation has to *reduce the difference between the distributions of the actual evolution path and an evolution path under random selection*, with respect to the parameters adapted. This reduces the correlation between selected mutation steps, because they *are* uncorrelated under *random* selection. The distribution of the actual evolution path can only be estimated using the selection information properly. Due to special properties of the normal distribution,[3] the distribution of the evolution path under random selection looks like the mutation distribution with larger variance.

Let us apply the principle to the self-adaptation of one global step size δ. In this case, δ^2 determines the overall variance of the mutation distribution. If parallel correlation between selected mutation steps makes the actual evolution path longer than expected under random selection, δ is enlarged. If anti-parallel correlation between selected mutation steps makes the actual evolution path shorter than expected under random selection, δ is reduced. On the one hand, this follows our fundamental principle and reduces the difference between the distributions discussed above, with respect to their overall variance. On the other hand, as substantiated by experiments, this leads to selected steps being uncorrelated and adapts optimal step size precisely. Using only the length of *single* mutation steps for the adaptation (e.g. with mutative step size control) usually leads to a global step size smaller than opti-

[1] The mutation is regarded to be the main operator in the ES.

[2] Object parameter space is a subspace of $\mathbb{R}^n$.

[3] The mutation distribution is assumed to be a normal distribution with zero mean.

mal.

The principle still works, considering the *shape* of the normal mutation distribution. If, for example, the evolution path is long with respect to a certain direction, the distribution is elongated in this direction, making the mutation distribution more similar to the evolution path. If the evolution path is short, with respect to a certain direction, the distribution is flattened in this direction. Both distribution changes tend to decrease correlations between successively selected mutation steps.

In this paper we apply the principle to the **adaptation of *arbitrary* normal mutation distributions** with zero mean. One such adaptation scheme has been introduced in [8], where the mutation distribution is changed by mutating $n\,(n-1)/2$ rotation angles and n variances. We found this scheme to be dependent on orientation and permutation(!) of the coordinate axes by experiments [2]. The rotations, performed in canonical planes, are the obvious reason for that dependency. The generating set adaptation (**GSA**), proposed in [2], adapts arbitrary normal mutation distributions *independently* of the chosen coordinate system.

What do we expect from an adaptation scheme like the CMA? On the one hand, with respect to the optimization process, quadratic problems can in principle be transformed into the hypersphere problem by the CMA [5]. Therefore, we expect the CMA to reliably adapt such problems. On the other hand, the adaptation process obviously cannot look beyond the horizont of the distribution variance. Because overall variance is usually small compared to the distance to optimum,[4] adaptation of the mutation distribution is a pretty *local* process, and we do not expect to improve *global* performance properties of the ES, especially with respect to multimodal objective functions.

In the next section, we present the covariance matrix adaptation (**CMA**). The CMA is closely related to the GSA as discussed in Sect. III. Section IV shows simulation results for different (1,10)-ESs, including CMA and GSA, while Sect. V gives a conclusion.

II. The Covariance Matrix Adaptation (CMA)

Mutation steps of object and strategy variables will be explained for the (1,λ)-ES. The extension to the (μ,λ)-ES without recombination is straightforward. Because recombination disturbs the adaptation process, the applicability of any standard recombination scheme (cf. e.g. [1; 7]) on strategy parameters seems questionable.

A. Object Variables

To carry out an arbitrarily normally distributed mutation step with expectation $\mathbf{0}$, the $\mathsf{N}(\mathbf{0}, \boldsymbol{I})$ distributed vector $\boldsymbol{z}$, where $\boldsymbol{I}$ denotes the identity matrix, is linearly transformed by a $n \times n$-matrix $\boldsymbol{B}$. $\boldsymbol{Bz}$ is then $\mathsf{N}(\mathbf{0}, \boldsymbol{BB}^t)$ distributed.[5] Choosing $\boldsymbol{B}$ in a suitable way, any normal distribution with zero mean can be generated by this transformation. To control overall variance of the mutation distribution, $\boldsymbol{Bz}$ is multiplied by the scalar δ. The mutation step for the object variable vector $\boldsymbol{x}$ for each descendant $k = 1, \ldots, \lambda$ reads

$$\boldsymbol{x}^{\mathrm{N}_k} = \boldsymbol{x}^{\mathrm{E}} + \delta^{\mathrm{E}} \boldsymbol{B}^{\mathrm{E}} \boldsymbol{z}_k \ , \tag{1}$$

where

- $\boldsymbol{x} = (x_1, \ldots, x_n)^t \in \mathbb{R}^n$ Object variable vector to be optimized. n is the dimension of the problem.
- E Index for the parent (*E*lder).
- N_k $k = 1, \ldots, \lambda$ Index for the descendant (*N*ewer) k.
- $\boldsymbol{z} = (z_1, \ldots, z_n)^t \sim \mathsf{N}(\mathbf{0}, \boldsymbol{I})$ z_i are independent $\mathsf{N}(0,1)$ distributed. $\boldsymbol{z}_k \in \mathbb{R}^n$ $(k = 1, \ldots, \lambda)$ are independent realizations of $\boldsymbol{z}$.
- $\delta > 0$ Global step size.
- $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ Matrix, which linearly transforms $\boldsymbol{z}$. $\boldsymbol{B}$ can be seen as the basis on which the normal distribution works. See also Sect. II.B.

B. Strategy Variables

The adaptation of the mutation distribution is separated into two parts. First, a covariance matrix is adapted, without taking into account overall variance[6]. Speaking about the "covariance matrix of the mutation distribution" we always refer to this matrix, which represents all correlations and variance *quotients* of the mutation distribution. Second, overall variance is adapted. The reason for applying two adaptation mechanisms are the fairly different time scales on which these mechanisms should work: Overall variance should be able to change as fast as required on the hypersphere problem. The covariance matrix can only be adapted on a larger time scale, because selection information for the adaptation of $n\,(n+1)/2$ parameters must be collected, before doing notable changes. Otherwise the mutation distribution will degenerate into subspaces due to stochastic fluctuations. *Both* time scale requirements can be met by the suggested separated adaptation.

We call these adaptation mechanisms **derandomized**, because on the one hand, strategy parameters are not subject to *direct* mutations, but to the same (although transformed) stochastic variations as the object variables: The random number $\boldsymbol{z}$ is the only stochastic source and the same realization of $\boldsymbol{z}$ is used in (1), (2) and (4). On the other hand, adaptation speed is suited to the number of parameters to be adapted. This essentially reduces stochastic fluctuations of the strategy parameters, which may ruin the adaptation process as well as the whole optimization. For an introduction to the derandomized approach to self-adaptation see [3].

We will first discuss the mutation of the **covariance matrix** $\boldsymbol{C}$, which determines $\boldsymbol{B}$. The vector $\boldsymbol{Bz}$ is the basic source for changing the covariance matrix. Actually, we use a summation vector $\boldsymbol{s}$, which is calculated by a weighted sum of the mutation steps, the individual's ancestors made over the passed generations. $\boldsymbol{s}$ represents the

[4] This is true *especially* in high dimensional search spaces.

[5] $\boldsymbol{BB}^t$ is the covariance matrix of the distribution.

[6] Overall variance of the $\mathsf{N}(\mathbf{0}, \boldsymbol{A})$ distribution means $\sum_i a_{ii}$.

selected mutation steps over the generation sequence or, in other words, an evolution path.

The normal distribution with zero mean, which most probably (re-)produces $\boldsymbol{s}$, is a line distribution along $\boldsymbol{s}$ with variance $\|\boldsymbol{s}\|^2$. This is the $\mathsf{N}(\boldsymbol{0}, \boldsymbol{s}\boldsymbol{s}^t)$ distribution. To follow the fundamental principle, stated in Sect. I, we adapt the covariance matrix $\boldsymbol{C}$ of the mutation distribution making the vector $\boldsymbol{s}$ more likely to appear. Summation in $\boldsymbol{s}$, also referred to as "cumulation", and adaptation of $\boldsymbol{C}$ read

$$\boldsymbol{s}^{\mathrm{N}_k} = (1-c)\cdot\boldsymbol{s}^{\mathrm{E}} + c_{\mathrm{u}}\cdot\boldsymbol{B}^{\mathrm{E}}\boldsymbol{z}_k \quad (2)$$

$$\boldsymbol{C}^{\mathrm{N}_k} = (1-c_{\mathrm{cov}})\cdot\boldsymbol{C}^{\mathrm{E}} + c_{\mathrm{cov}}\cdot\boldsymbol{s}^{\mathrm{N}_k}\left(\boldsymbol{s}^{\mathrm{N}_k}\right)^t \quad (3)$$

where

$c \in]0,1]$ determines the accumulation time for $\boldsymbol{s}$.

$c_{\mathrm{u}} = \sqrt{c\cdot(2-c)}$ normalizes the variance of $\boldsymbol{s}$ by solving the equation $1^2 = (1-c)^2 + c_{\mathrm{u}}^2$.

$\boldsymbol{s}^{\mathrm{start}} = \boldsymbol{0}$.

$\boldsymbol{C} \in \mathbb{R}^{n\times n}$ Covariance matrix of the mutation distribution. $\boldsymbol{C}$ determines $\boldsymbol{B}$, so that $\boldsymbol{B}\boldsymbol{z} \sim \mathsf{N}(\boldsymbol{0}, \boldsymbol{C})$ holds, that is $\boldsymbol{C} = \boldsymbol{B}\boldsymbol{B}^t$. $\boldsymbol{C}^{\mathrm{start}} = \boldsymbol{I}$.

$c_{\mathrm{cov}} \in]0,1[$ determines the time of averaging the distributions $\boldsymbol{s}\boldsymbol{s}^t$ over the generation sequence.

δ does not appear in (2), because a possibly fast change of δ would distort the result of the summation otherwise. For example, as long as δ gets smaller by a factor less than $(1-c)$ each generation, new information would not effect $\boldsymbol{s}$ considerably.

Equation (3) is quite similar to the update rule of quasi-Newton methods in classical optimization. In both cases second order estimation of the problem topology is done.

While $\boldsymbol{C}$ determines $\boldsymbol{B}$ through $\boldsymbol{C} = \boldsymbol{B}\boldsymbol{B}^t$, the solution of this equation is not unique. For the reason of global step size adaptation (see below) we choose as column vectors of $\boldsymbol{B}$ the eigenvectors of $\boldsymbol{C}$, their lengths given by the square root of the corresponding eigenvalues. The eigenvectors can be seen as the result of a principle component analysis of (exponentially decreasing weighted) evolution paths.

We give a **geometrical idea** of the resulting distribution change, ignoring the initial distribution: A mutation step at generation $g+1$ is composed by g line mutations along certain vectors $\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(g)}$. More precisely, $\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(g)}$ are multiplied each with a $\mathsf{N}(0,1)$ distributed random number and the results are added up. To produce the mutation steps of the next generation, $\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(g)}$ are shortened by multiplication with $(1-c)$ and a new vector $\boldsymbol{s}^{(g+1)}$ is added to the vector tupel. The GSA implements a similar geometrical idea *directly* (see also Sect. III).

The **global step size** δ is adapted in a quite similar way, originally introduced in [4] and here referred to as "cumulative step size adaptation". Following the principle of Sect. I, an evolution path (here the sum of "normalized" mutation steps selected in the generation sequence) is, with respect to its length, compared to the expected value under random selection. Summation of mutation steps and adaptation of δ read

$$\boldsymbol{s}_\delta^{\mathrm{N}_k} = (1-c)\cdot\boldsymbol{s}_\delta^{\mathrm{E}} + c_{\mathrm{u}}\cdot\boldsymbol{B}_\delta^{\mathrm{E}}\boldsymbol{z}_k \quad (4)$$

$$\delta^{\mathrm{N}_k} = \delta^{\mathrm{E}}\cdot\exp\left\{\beta\cdot\left(\|\boldsymbol{s}_\delta^{N_k}\| - \widehat{\chi}_n\right)\right\} \quad (5)$$

where

$\boldsymbol{B}_\delta^{\mathrm{E}} \in \mathbb{R}^{n\times n}$ equals $\boldsymbol{B}^{\mathrm{E}}$ with normalized columns.

$c \in]0,1]$ determines the accumulation time for $\boldsymbol{s}$.

$c_{\mathrm{u}} = \sqrt{c\cdot(2-c)}$, see also c_{u} in (2).

$\boldsymbol{s}_\delta^{\mathrm{start}} = \boldsymbol{0}$.

β Parameter for damping the step size variation between successive generations.

$\widehat{\chi}_n = \sqrt{n}\left(1 - \frac{1}{4\,n} + \frac{1}{21\,n^2}\right)$ approximates the expectation of the χ_n-distribution, which is the distribution of the length of a $\mathsf{N}(\boldsymbol{0}, \boldsymbol{I})$ distributed random vector in $\mathbb{R}^n$.[7]

Equation (4) looks almost exactly like (2). The only difference is the normalization of the columns in $\boldsymbol{B}_\delta$, which is important to derive the *expected* length of $\boldsymbol{s}_\delta$: Because the columns of $\boldsymbol{B}_\delta$ are normalized eigenvectors[8], $\boldsymbol{B}_\delta\,\boldsymbol{z}$ is $\mathsf{N}(\boldsymbol{0}, \boldsymbol{I})$ distributed. In this way, before selection takes place, $\boldsymbol{s}_\delta^{\mathrm{N}_k}$ is $\mathsf{N}(\boldsymbol{0}, \boldsymbol{I})$ distributed, if $\boldsymbol{s}_\delta^{\mathrm{E}}$ is $\mathsf{N}(\boldsymbol{0}, \boldsymbol{I})$ distributed and its expected length is well-known. Parallel/anti-parallel correlation between successively selected steps, as well as selection of large/small steps enlarge/reduce the variance of $\boldsymbol{s}_\delta^{\mathrm{N}_{sel}}$, if $sel = 1, \ldots, \lambda$ denotes the index of the selected descendant.

Equation (5) adapts δ, to reduce the difference between the length of the actual "normalized" evolution path $\|\boldsymbol{s}_\delta\|$ and its expected length.

C. Discussion of Introduced External Parameters

$\beta > 0$ determines the step size changing rate. Small values lead to small changing rates and scale down stochastic fluctuations of the step size. We choose β as small as possible, but as large as necessary to allow nearly optimal step size changes on the hypershere problem. Smaller values can be sensible to prevent premature decrease of the step size. In any case, β has to be chosen considerably smaller than c, to prevent oscillations due to the cumulation.

$c_{\mathrm{cov}} \in]0,1[$ determines the time horizont of $\boldsymbol{C}$ and $1/c_{\mathrm{cov}}$ can be regarded as life span. Roughly speaking, after $1/c_{\mathrm{cov}}$ generations about $2/3$ of the original information has vanished. In general, the larger c_{cov}, the faster is the adaptation, and the smaller c_{cov}, the more reliable it becomes. Because of the number of free parameters in $\boldsymbol{C}$, c_{cov} should be smaller than $10/n^2$. Otherwise, the amount of selection information is not sufficient for the adaptation and the distribution degenerates into a subspace. To prevent oscillations due to the cumulation, c_{cov} has to be chosen clearly smaller than c.

$c \in]0,1]$ determines the accumulation time for the evolution paths $\boldsymbol{s}$ and $\boldsymbol{s}_\delta$. If $c = 1$, no accumulation takes

[7] In implementations, obtain $\mathsf{E}(\|\boldsymbol{z}\|)$ *by simulation* and make sure that $\widehat{\chi}_n = \mathsf{E}(\|\boldsymbol{z}\|)$.

[8] and because eigenvectors are orthogonal

place. In this case correlation information is not utilized, and step size adaptation declines with increasing problem dimension, while adaptation of the distribution usually still works, even though on a larger time scale. Small values of c make $\boldsymbol{s}$ more accurate, because a lot of selection information is accumulated then. On the other hand, if β or c_{cov} must be chosen smaller than optimal due to a small c, adaptation time increases.

As a compromise, we found $c = 1/\sqrt{n}$ useful, while we choose $\beta = 1/n$ and $c_{\text{cov}} = 2/n^2$.

III. Relation to Previous Work

The CMA, introduced in Sect. II, is closely related to the generating set adaptation (GSA), described in [2]. The algorithms, as described, yield only two differences. The first relates to the global step size adaptation. In the CMA, cumulative global step size adaptation can be used in a sensible way. Choosing $\boldsymbol{B}_\delta$ with orthonormal columns and accumulating the mutation steps $\boldsymbol{B}_\delta \boldsymbol{z}$ (rather than $\boldsymbol{z}$), correlation between successive steps is meaningful *and* the expectation of the step length $\|\boldsymbol{B}_\delta \boldsymbol{z}\|$ is known. In the GSA the global step size adaptation is mutative (cf. Sect. IV).

Second, different weights are used for summing up the distributions, which relate to the vector $\boldsymbol{s}$ in (3) or to the generating vectors, respectively. In both schemes, the weights are chosen mainly for the sake of implementational simplicity. While the GSA uses equal weights, the exponentially decreasing weights, as used in the CMA, seem to be more natural, because more recent selection information has a comparatively higher influence on the mutation distribution here. The covariance matrices of the mutation distributions of the two algorithms at generation $g+1 \geq m$, where m is the number of vectors in the generating set, read

$$\boldsymbol{C}_{\text{CMA}}^{(g+1)} = c_{\text{cov}} \sum_{i=1}^{g} (1 - c_{\text{cov}})^{g-i} \boldsymbol{S}^{(i)} + (1 - c_{\text{cov}})^{g} \boldsymbol{I} \tag{6}$$

$$\boldsymbol{C}_{\text{GSA}}^{(g+1)} = \frac{1}{m} \sum_{i=g-m+1}^{g} \boldsymbol{S}^{(i)} \ , \tag{7}$$

where $\boldsymbol{S}^{(i)} \in \mathbb{R}^{n \times n}$ equals $\boldsymbol{s}^{\text{N}_{sel}} \left(\boldsymbol{s}^{\text{N}_{sel}}\right)^t$ at generation i. Choosing parameter $m \approx n^2$ relates to $c_{\text{cov}} \approx 1/n^2$. In Fig. 1 the weights at different generations are shown for $c_{\text{cov}} = 1/10$ and $m = 20$.

The disadvantage of the CMA is, that eigenvalues and –vectors of $\boldsymbol{C}$ have to be found in a numerical process.[9] This process is $\mathcal{O}(n^3)$, which is also the case for the GSA.

Advantages compared to the GSA are

- The required storage capacity is $\mathcal{O}(n^2)$ rather than $\mathcal{O}(n^3)$.
- Cumulative global step size adaptation can be applied easily.
- Threatening degeneration of the distribution can be detected and avoided, for example by restricting the relation between largest and smallest eigenvalue or setting parameter c_{cov} to smaller values.
- Searching for eigenvalues and –vectors every n^{th} generation merely has no substantial (negative) consequences, because the time scale for notable changes of $\boldsymbol{C}$ is n^2. This makes CMA's computational effort $\mathcal{O}(n^2)$, while there is no comparable technique available for the GSA.
- The use of the covariance matrix and its eigensystem makes a further development of specialized adaptation control mechanisms possible — or at least more convenient.

[9] We used the C-routines tred2.c and tqli.c of [6] in double precision. Allowing up to 300 iterations, we always had stable and consistent results for matrix conditions up to 10^{14}.

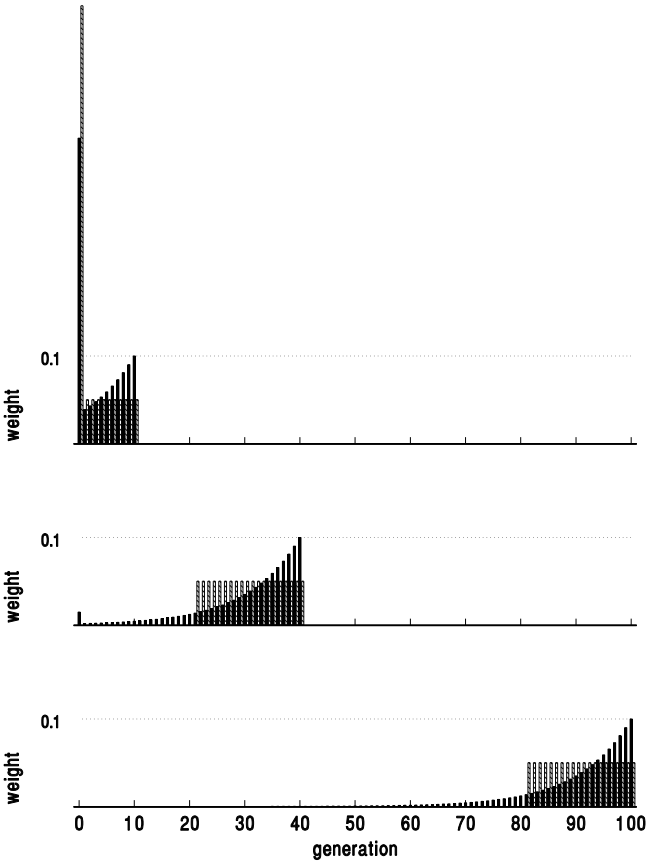

Fig. 1. Weight coefficients in (6) and (7), which constitute the covariance matrix for the covariance matrix adaptation (CMA, solid) and the generating set adaptation (GSA, hatched) at generation 10, 40 and 100. The first bar reflects the weight for the initial distribution (CMA) or the sum of all weights, which correspond to initialization vectors (GSA). $c_{\text{cov}} = 1/10$ and $m = 20$.

IV. Simulations

Simulations were done on two different objective functions with $n = 20$:

1. An arbitrarily orientated hyperellipsoid [2]

$$f_{\text{elli}}(\boldsymbol{x}) = \sum_{i=1}^{n} \left(1000^{\frac{i-1}{n-1}} \langle \boldsymbol{x}, \boldsymbol{o}_i \rangle\right)^2 ,$$

where $\langle ., . \rangle$ denotes the canonical scalar product and $\boldsymbol{o}_1, \ldots, \boldsymbol{o}_n$ is an arbitrarily orientated orthonormal basis. The axis ratio between i^{th} and $(i+1)^{\text{th}}$ axis is con-

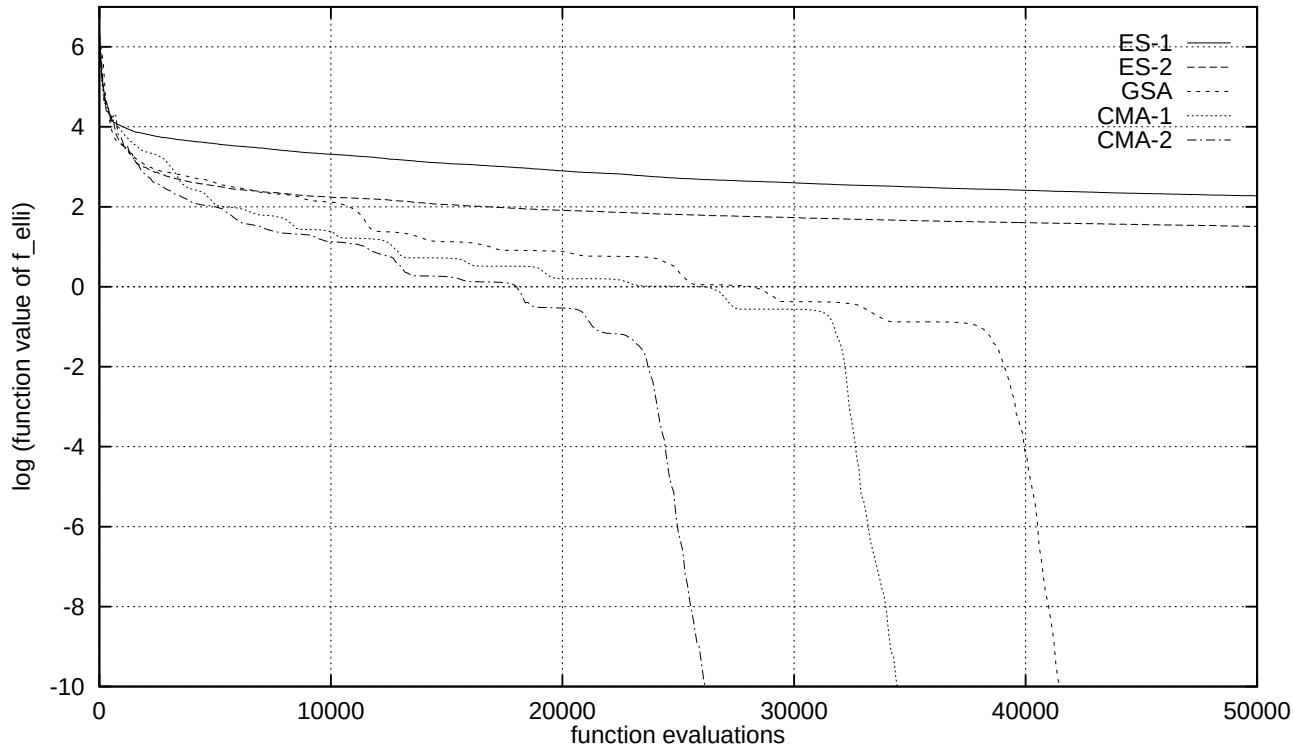

Fig. 2. Simulations on the arbitrarily orientated hyperellipsoid f_{elli}, where $n = 20$. Starting point is $(1, \ldots, 1)^t$, $\delta^{\text{start}} = 0.1$. The different $(1, 10)$-ESs shown are described in the text.

stant, the condition of the hyperellipsoid is 10^6. Vector $\boldsymbol{o}_i$ is first generated with (0,1)-normally distributed components. Then the projections on the previously generated vectors $\boldsymbol{o}_1, \ldots, \boldsymbol{o}_{i-1}$ are subtracted and normalization is done.

2. The generalized Rosenbrock's function

$$f_{\text{rosen}}(\boldsymbol{x}) = \sum_{i=1}^{n-1} \left(100 \left({x_i}^2 - x_{i+1}\right)^2 + (x_i - 1)^2\right),$$

where multiple dependencies between parameters arise.

Both functions are nonlinear, non-separable, scalable and resistant to simple hill-climbing — the most desirable properties of test problems, as pointed out in [9].

We compared the following **five** $(1, 10)$-evolution **strategies**:

- ES–1 — A simple evolution strategy with damped mutative global step size adaptation only (see below). No further adaptation takes place and the mutation distribution is isotropic.
- ES–2 — A strategy like ES–1, here with *cumulative* global step size adaptation.
- GSA — The Generating Set Adaptation (GSA) as described in [2].
- CMA–1 — The CMA with damped mutative adaptation of the global step size (see below). The only difference between CMA–1 and GSA are the summation weights as discussed in Sect. III.
- CMA–2 — The CMA with cumulative global step size adaptation as described in Sect. II (external parameter setting see end of Sect. II.C). The only difference between CMA–2 and CMA–1 is the global step size adaptation.

ES–1 and ES–2 are shown for comparison merely. The **damped mutative step size control** of ES–1, GSA and CMA–1 works as follows: The mutation step, for example $\delta \boldsymbol{B}^{\text{E}} \boldsymbol{z}_k$ in (1), is multiplied by the step size changing factor ξ_k, and step size adaptation is done by $\delta^{\text{N}_k} = \delta^{\text{E}}(\xi_k)^{\beta}$, replacing (5). ξ_k equals 1.5 or 1/1.5 with equal probability, and the damping parameter β is chosen $1/\sqrt{n}$.

On the **hypersphere** problem (not shown here) ES–2 performs best. Progresses of the other strategies are at least 75% of the progress of ES–2 here.[10]

Simulation runs on the arbitrarily orientated hyperellipsoid f_{elli} are shown in **Fig. 2**. The standard deviations for reaching the function value 10^{-10} for GSA, CMA–1 and CMA–2 are about 1000 function evaluations. Consequently, the difference between these strategies is highly significant. The difference between ES–1 and ES–2 as can be seen in the figure, is not significant. In a (much) later stage of optimization, a difference will become obvious and ES–2 reaches function value 10^{-10} after about $4 \cdot 10^7$ function evaluations, ES–1 after about 10^8 function evaluations.

Furthermore the simulation shows, that the adaptation processes of the CMA and the GSA are looking quite similar. The reason for the faster adaptation of CMA–1 than GSA may not only be the different method of weighting. Life spans $1/c_{\text{cov}} = n^2/2$ and $m = 1.5n^2$ (cf. Sect. III) are not exactly comparable either.

[10] Progress may be defined as expectation of $\log\left(f^{\text{evalstart}}/f^{\text{evalstop}}\right)$.

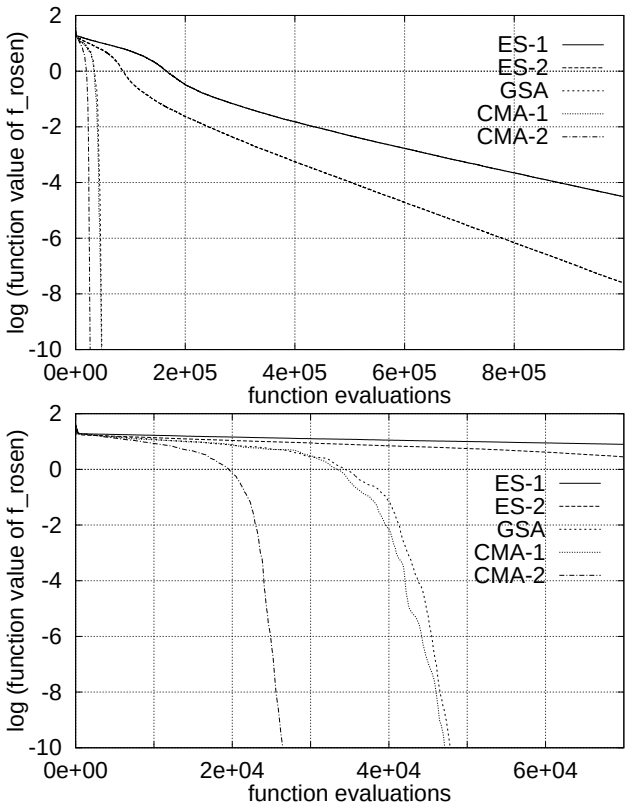

Fig. 3. Simulations on the generalized Rosenbrock's function f_{rosen}, where $n = 20$. Starting point is $\mathbf{0}$, $\delta^{\text{start}} = 0.1$. The different $(1, 10)$-ESs shown are described in the text. The lower diagram is an enlarged detail of the upper one.

Cumulative global step size adaptation makes the adaptation process of the CMA about 25% faster (CMA–2 vs. CMA–1).

In principle, adaptation time on f_{elli} scales with n^2, as could be confirmed in simulations without cumulation, not shown here.

After adaptation has been completed, the three algorithms which adapt arbitrary distributions, realize progress values comparable to those on the hypersphere problem.

Simulation runs on the generalized Rosenbrock's function f_{rosen} are shown in **Fig. 3** on two different time scales. Again, the standard deviations for reaching the function value 10^{-10} for CMA–1, CMA–2 and GSA are about 10^3 function evaluations. In this case, there is no significant difference between CMA–1 and GSA. Cumulative global step size adaptation makes the adaptation process of the CMA about 50% faster (CMA–2 vs. CMA–1) and significantly speeds up the strategy with adaptation of just one global step size as well (ES–2 vs. ES–1).

V. Conclusion

This paper aimed at two goals. First, we argued to utilize the evolution path, i.e. a *sum* of successively selected mutation steps, for the adaptation of the mutation distribution, as proposed in [4]. This usually makes the adaptation faster *and* more reliable. Second, we introduced the covariance matrix adaptation (CMA), which uses the evolution path to adapt arbitrary normal mutation distributions with zero mean. The CMA is a new formulation of the generating set adaptation (GSA) proposed in [2] and reliably adapts hyperellipsoids with high axis ratio. The adaptation is independent of the chosen coordinate system. Cumulative global step size adaptation can be applied easily, which can speed up the CMA by the factor two. Furthermore, the CMA facilitates control and manipulation of important distribution parameters. In conclusion we state

1. Adapting as many as $n\,(n+1)/2$ free distribution parameters with constant population size usually leads to adaptation times of $\mathcal{O}(n^2)$.
2. In general, the CMA should be preferred to the GSA for several reasons.

Acknowledgements

This work was supported by the *Bundesministerium für Bildung und Forschung* under grant 01 IB 404 A. We would like to gratefully thank Andreas Gawelczyk and Iván Santibáñez-Koref for their persistent support of our work.

References

[1] Bäck, T., Schwefel, H.-P. (1993). An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation*, 1(1): 1–23.

[2] Hansen, N., Ostermeier, A. & Gawelczyk, A. (1995). On the Adaptation of Arbitrary Normal Mutation Distributions in Evolution Strategies: The Generating Set Adaptation. In: Eshelman (ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*: 57–64. San Francisco: Morgan Kaufmann.

[3] Ostermeier, A., Gawelczyk, A. & Hansen, N. (1994). A Derandomized Approach to Self-Adaptation of Evolution Strategies. *Evolutionary Computation*, 2(4): 369–380.

[4] Ostermeier, A., Gawelczyk, A. & Hansen, N. (1994). Step-size Adaptation Based on Non-local Use of Selection Information. In: Davidor, Y., Schwefel, H.-P. & Männer, R. (eds.), *Parallel Problem Solving from Nature – PPSN III, Proceedings*: 189–198. Berlin: Springer.

[5] Rudolph, G. (1992). On Correlated Mutations in Evolution Strategies. In: Männer, R. & Manderick, B. (eds.), *Parallel Problem Solving from Nature, 2, Proceedings*: 105–114. Amsterdam: North-Holland.

[6] Press, W.H., Teukolsky, S.A., Vetterling, W.T. & Flannery, B.P. (1988). *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press.

[7] Rechenberg, I. (1994), *Evolutionsstrategie '94*, Stuttgart: frommann-holzboog.

[8] Schwefel, H.-P. (1981). *Numerical optimization of computer models.* Chichester: Wiley.

[9] Whitley, D., Mathias, K. & Dzubera, J. (1995). Building Better Test Functions. In: Eshelman (ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*: 239–246. San Francisco: Morgan Kaufmann.