Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES)

Nikolaus Hansen hansen@bionik.tu-berlin.de
Fachgebiet für Bionik, Technische Universität Berlin, Ackerstr. 71-76, 13355 Berlin, Germany

Sibylle D. Müller muellers@inf.ethz.ch
Institute of Computational Science, ETHZ Computational Laboratory (CoLab), Swiss Federal Institute of Technology, 8092 Zürich, Switzerland

Petros Koumoutsakos petros@inf.ethz.ch
Institute of Computational Science, ETHZ Computational Laboratory (CoLab), Swiss Federal Institute of Technology, 8092 Zürich, Switzerland

Abstract

This paper presents a novel evolutionary optimization strategy based on the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). This new approach is intended to reduce the number of generations required for convergence to the optimum. Reducing the number of generations, i.e., the time complexity of the algorithm, is important if a large population size is desired: (1) to reduce the effect of noise; (2) to improve global search properties; and (3) to implement the algorithm on (highly) parallel machines. Our method results in a highly parallel algorithm which scales favorably with large numbers of processors. This is accomplished by efficiently incorporating the available information from a large population, thus significantly reducing the number of generations needed to adapt the covariance matrix. The original version of the CMA-ES was designed to reliably adapt the covariance matrix in small populations but it cannot exploit large populations efficiently. Our modifications scale up the efficiency to population sizes of up to $10n$, where n is the problem dimension. This method has been applied to a large number of test problems, demonstrating that in many cases the CMA-ES can be advanced from quadratic to linear time complexity.

Keywords

Evolution strategy, derandomized self-adaptation, covariance matrix adaptation, evolution path, parallelization.

1 Introduction

One of the commonly proposed advantages of evolution strategies (ESs) is that they can be easily parallelized, see e.g. Schwefel (1995) or Bäck, Hammel, and Schwefel (1997). ESs with λ offspring per generation (population size λ) are usually parallelized by distributing the function evaluation for each of the offspring on a different processor. However, when the number of offspring is smaller than the number of available processors, the advantage of using evolution strategies in parallel cannot be fully exploited. Consequently, when large numbers of processors are available, it is desirable to develop an algorithm that can handle a large population efficiently.

Our approach is based on a derandomized ES with covariance matrix adaptation (CMA-ES). Experimental results (Hansen and Ostermeier, 1997, 2001) have shown the advantageous convergence properties of the CMA-ES when compared to several other evolution strategies for a wide class of problems. The primary feature of the CMA-ES is its reliability in adapting an arbitrarily oriented scaling of the search space in small populations. In particular, the algorithm is—apart from the initialization—independent of any linear transformation of the coordinate system.

For our purposes, adaptation time is defined as the *number of generations* needed to realize adaptive changes to get a (nearly) optimal covariance matrix under the given function topology. When optimizing considerably complex (e.g., highly nonseparable) functions, the adaptation time becomes the limiting factor for the performance of the CMA-ES if the problem dimension n exceeds a certain threshold, typically $n \geq 10$. The number of generations needed to adapt the covariance matrix of the search distribution to the function topology is the prominent factor for the overall performance in this case. The reason is that in the CMA-ES $(n^2+n)/2$ elements of the symmetric covariance matrix $\boldsymbol{C}$ need to be adapted while the search process itself needs to adjust only n variables.[1] It is interesting to note that for population sizes greater than 20 the adaptation time becomes practically independent of the population size. Thus, the performance in number of function evaluations decreases linearly with increasing population. In other words, the implementation of the original CMA-ES on highly parallel computer architectures, e.g., on Beowulf clusters with hundreds of processors, results in no substantial advantage compared to the use of about twenty processors. On several complex functions, the time complexity (Beyer 1996), i.e., the observed number of generations to reach f_{stop}, is proportional to n^2, independent of the population size and the processor number.

Hence, by simply increasing the number of offspring λ, it is not possible to increase the efficiency of the standard CMA-ES and consequently to use efficiently parallel computer architectures. In this paper, we present a method for handling larger populations efficiently so as to increase the adaptation speed of the covariance matrix adaptation. This should be possible as a larger population can contain more information available to be exploited to obtain a reduced adaptation time. Compared with the original adaptation mechanism, the proposed modification would usually require much fewer generations when λ is large but could be slightly less effective within small populations.

The paper is organized as follows. First, Section 2 outlines the working principles of the original algorithm, the CMA-ES. Then Section 3 presents the algorithmic modifications. Section 4 outlines the method of obtaining new strategy parameters. In Section 5, we present tests for the proposed methodology. We discuss the simulation results in Section 6 and give our conclusions in Section 7.

[1] The optimization needs $\mathcal{O}(n)$ generations, independently of the chosen covariance matrix. The adaptation of the complete covariance matrix needs $\mathcal{O}(n^2)$. Therefore, for very large problem dimensions the optimization goal can be reached before an effective adaptation of the distribution shape is achieved. In a somewhat realistic scenario, let for example the number of generations required to reach f_{stop} be equal to $10^4 n$ without adaptation. And let the optimization with adaptation take n^2 generations for the adaptation itself and additionally n generations to reach f_{stop} once the covariance matrix has been adapted, yielding $n^2 + n$ generations in total. What happens, if $n = 10, 10^3$, or 10^5? f_{stop} is reached after the minimum of $n^2 + n$ or $10^4 n$ generations, that is after about $110, 10^6$, and 10^9 generations, respectively. The first two numbers reflect mainly the adaptation time n^2 for $n = 10; 10^3$. For $n = 10^5$ the adaptation time is 10^{10} and therefore the optimization goal is reached already with the original covariance matrix after $10^4 n = 10^9$ generations, before an effective adaptation has taken place. Empirical results for problems, where the condition number of the covariance matrix (that is the ratio of the largest to the smallest eigenvalue) is 10^6, have shown that the latter situation appears only if $n > 1000$ (as in the example) or does not happen at all, if the adaptation time happens to be linear with n (Hansen and Ostermeier 2001).

2 Algorithm of the CMA-ES

Following Hansen and Ostermeier (2001), in the $(\mu_{\mathrm{I}}, \lambda)$-CMA-ES the λ offspring of generation $g+1$ are computed by

$$\boldsymbol{x}_k^{(g+1)} = \langle \boldsymbol{x} \rangle_\mu^{(g)} + \sigma^{(g)} \underbrace{\boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} \boldsymbol{z}_k^{(g+1)}}_{\sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{C}^{(g)}\right)} \quad , \quad k = 1, \ldots, \lambda, \tag{1}$$

where

$$\langle \boldsymbol{x} \rangle_\mu^{(g)} = \frac{1}{\mu} \sum_{i \in I_{sel}^{(g)}} \boldsymbol{x}_i^{(g)} \tag{2}$$

represents the center of mass of the selected individuals of generation g, and $I_{sel}^{(g)}$ is the set of indices of the selected individuals of generation g with $|I_{sel}^{(g)}| = \mu$. $\sigma^{(g)}$ is the global step size.

The random vectors $\boldsymbol{z}_k$ from Equation (1) are $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ distributed (n-dimensional normally distributed with expectation zero and the identity covariance matrix) and serve to generate offspring for generation $g+1$. Similar to Equation (2), we can calculate their center of mass as:

$$\langle \boldsymbol{z} \rangle_\mu^{(g+1)} = \frac{1}{\mu} \sum_{i \in I_{sel}^{(g+1)}} \boldsymbol{z}_i^{(g+1)}. \tag{3}$$

The covariance matrix $\boldsymbol{C}^{(g)}$ of the random vectors $\boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} \boldsymbol{z}_k^{(g+1)}$ is a symmetrical positive $n \times n$-matrix. The columns of the orthogonal matrix $\boldsymbol{B}^{(g)}$ represent normalized eigenvectors of the covariance matrix. $\boldsymbol{D}^{(g)}$ is a diagonal matrix whose elements are the square roots of the eigenvalues of $\boldsymbol{C}^{(g)}$. Hence, the relation of $\boldsymbol{B}^{(g)}$ and $\boldsymbol{D}^{(g)}$ to $\boldsymbol{C}^{(g)}$ can be expressed by

$$\boldsymbol{C}^{(g)} = \boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} \left(\boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} \right)^{\mathrm{T}} \quad \text{and} \quad \boldsymbol{C}^{(g)} \boldsymbol{b}_i^{(g)} = \left(d_{ii}^{(g)} \right)^2 \cdot \boldsymbol{b}_i^{(g)} \tag{4}$$

where $\boldsymbol{b}_i^{(g)}$ represents the i-th column of $\boldsymbol{B}^{(g)}$ and $\|\boldsymbol{b}_i^{(g)}\| = 1$ and $d_{ii}^{(g)}$ are the diagonal elements of $\boldsymbol{D}^{(g)}$. Surfaces of equal probability density of the random vectors $\boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} \boldsymbol{z}_k^{(g+1)} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{C}^{(g)}\right)$ are (hyper-)ellipsoids whose main axes correspond to the eigenvectors of the covariance matrix. The squared lengths of the axes are equal to the eigenvalues of the covariance matrix.

In the following, the two adaptation mechanism of the CMA-ES are described: (i) the adaptation of the covariance matrix $\boldsymbol{C}^{(g)}$ and (ii) the adaptation of the global step size $\sigma^{(g)}$.

First, the evolution path $\boldsymbol{p}_c^{(g+1)}$ is calculated by

$$\boldsymbol{p}_c^{(g+1)} = (1 - c_c) \cdot \boldsymbol{p}_c^{(g)} + \sqrt{c_c \cdot (2 - c_c)} \cdot \underbrace{\frac{\sqrt{\mu}}{\sigma^{(g)}} \left(\langle \boldsymbol{x} \rangle_\mu^{(g+1)} - \langle \boldsymbol{x} \rangle_\mu^{(g)} \right)}_{= \sqrt{\mu}\, \boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} \langle \boldsymbol{z} \rangle_\mu^{(g+1)}} \tag{5}$$

and is used to build the covariance matrix of generation $g+1$

$$\boldsymbol{C}^{(g+1)} = (1 - c_{\mathrm{cov}}) \cdot \boldsymbol{C}^{(g)} + c_{\mathrm{cov}} \cdot \boldsymbol{p}_c^{(g+1)} \left(\boldsymbol{p}_c^{(g+1)} \right)^{\mathrm{T}}. \tag{6}$$

$\boldsymbol{C}$ is updated with a symmetric matrix of rank one (right summand in Equation (6)). Note here that using $c_c = 1$ in Equation (5) reduces the evolution path to $\sqrt{\mu}\boldsymbol{B}\boldsymbol{D}\langle \boldsymbol{z} \rangle_\mu$ which is the mean mutation step of the last generation multiplied by $\sqrt{\mu}$. Choosing $c_c < 1$ makes the adaptation usually faster and more reliable as the correlation between successive steps is exploited (Hansen and Ostermeier 2001). We even found it advantageous to choose $c_c \propto \frac{1}{n}$ because this setting achieves a linear scaling of the strategy on f_{cigar} (see Table 1 and Figure 4 below).

Second, to adapt the global step size σ, the evolution path $\boldsymbol{p}_\sigma^{(g+1)}$ is computed in analogy to the evolution path $\boldsymbol{p}^{(g+1)}$. The difference between the two evolution paths is that $\boldsymbol{p}_\sigma^{(g+1)}$ is not scaled by $\boldsymbol{D}^{(g)}$, resulting in

$$\boldsymbol{p}_\sigma^{(g+1)} = (1-c_\sigma)\cdot \boldsymbol{p}_\sigma^{(g)} + \sqrt{c_\sigma \cdot (2-c_\sigma)}\cdot \underbrace{\sqrt{\mu}\,\boldsymbol{B}^{(g)}\langle \boldsymbol{z} \rangle_\mu^{(g+1)}}_{= \boldsymbol{B}^{(g)}\left(\boldsymbol{D}^{(g)}\right)^{-1}\left(\boldsymbol{B}^{(g)}\right)^{-1}\frac{\sqrt{\mu}}{\sigma^{(g)}}\left(\langle \boldsymbol{x} \rangle_\mu^{(g+1)} - \langle \boldsymbol{x} \rangle_\mu^{(g)}\right)}. \tag{7}$$

The length of the evolution path determines the step size for generation $g+1$

$$\sigma^{(g+1)} = \sigma^{(g)} \cdot \exp\left(\frac{1}{d_\sigma}\frac{\|\boldsymbol{p}_\sigma^{(g+1)}\| - \hat{\chi}_n}{\hat{\chi}_n}\right), \tag{8}$$

where $\hat{\chi}_n = \mathrm{E}\left[\|\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})\|\right]$ is the expected length of a $(\boldsymbol{0}, \boldsymbol{I})$-normally distributed random vector and $d_\sigma > 1$ is the damping parameter. $\hat{\chi}_n$ is approximated by $\hat{\chi}_n \approx \sqrt{n}\left(1 - \frac{1}{4n} + \frac{1}{21n^2}\right)$ (Ostermeier, 1997).

Equation (7) reveals that the lengths of the axes of the mutation ellipsoid do not affect the global step size adaptation.

The strategy parameter setting is discussed in Hansen and Ostermeier (2001) in detail and the default setting is used as follows:

$$c_c = \frac{4}{n+4}, \quad c_{\mathrm{cov}} = \frac{2}{(n+\sqrt{2})^2}, \quad c_\sigma = \frac{4}{n+4}, \quad d_\sigma = c_\sigma^{-1} + 1 \tag{9}$$

Initial values are $\boldsymbol{p}^{(0)} = \boldsymbol{0}$, $\boldsymbol{p}_\sigma^{(0)} = \boldsymbol{0}$ and the initial covariance matrix $\boldsymbol{C}^{(0)}$ is the identity matrix $\boldsymbol{I}$. The problem dependent initial values for $\langle \boldsymbol{x} \rangle_\mu^{(0)}$ and $\sigma^{(0)}$ are given in Section 5.

3 Modified Algorithm

The adaptation of the $\frac{n^2+n}{2}$ elements of the covariance matrix requires a number of function evaluations in the CMA-ES that scales between $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$ (Hansen and Ostermeier (2001)), depending on the function topology. In the original CMA-ES, the covariance matrix is updated in every generation by adding the (weighted) outer product of the evolution path which is a symmetrical $n \times n$ matrix of rank one.

The idea of the modification of the algorithm is to adapt the covariance matrix by exploiting more of the information contained in larger populations. Instead of updating the covariance matrix with rank one information, we include higher rank information. This is achieved by modifying Equation (6), which describes the change of the covariance matrix, (i.e., the change of the mutation distribution shape), and by changing the strategy parameter c_{cov}. The global step size adaptation (Equations (7) and (8)) remains unchanged.

The modification is obtained by adding to Equation (6) the following term:

$$\begin{aligned} \boldsymbol{Z}^{(g+1)} &= \frac{1}{\mu} \sum_{i \in I_{sel}^{(g+1)}} \boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} \boldsymbol{z}_i^{(g+1)} \left(\boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} \boldsymbol{z}_i^{(g+1)} \right)^{\mathrm{T}} \\ &= \boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} \left(\frac{1}{\mu} \sum_{i \in I_{sel}^{(g+1)}} \boldsymbol{z}_i^{(g+1)} \left(\boldsymbol{z}_i^{(g+1)} \right)^{\mathrm{T}} \right) \left(\boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} \right)^{\mathrm{T}} \end{aligned} \tag{10}$$

that is a symmetrical $n \times n$ matrix with rank $\min(\mu, n)$.

Equation (6) is replaced by the new adaptation of the covariance matrix:

$$\boldsymbol{C}^{(g+1)} = (1 - c_{\mathrm{cov}}) \cdot \boldsymbol{C}^{(g)} + c_{\mathrm{cov}} \left(\alpha_{\mathrm{cov}} \cdot \boldsymbol{p}_c^{(g+1)} \left(\boldsymbol{p}_c^{(g+1)} \right)^{\mathrm{T}} + (1 - \alpha_{\mathrm{cov}}) \cdot \boldsymbol{Z}^{(g+1)} \right) \tag{11}$$

where $0 \leq \alpha_{\mathrm{cov}} \leq 1$. Note that for $\alpha_{\mathrm{cov}} = 1$ Equation (11) and Equation (6) are identical and the original CMA-ES is recovered. Decreasing α_{cov} changes the parameterized algorithm continuously. In Section 6, results are presented for $\alpha_{\mathrm{cov}} = 0$ and $\alpha_{\mathrm{cov}} = \frac{1}{\mu}$ and compared with the original CMA algorithm, where $\alpha_{\mathrm{cov}} = 1$.

The addition of $\boldsymbol{Z}^{(g+1)}$, a term with rank greater than one, for the update of $\boldsymbol{C}$ allows to give a higher weight to the second term in Equation (11) by enlarging the learning rate c_{cov}. The heuristic analysis to obtain the new c_{cov} is described in the next section, resulting in

$$c_{\mathrm{cov}} = \alpha_{\mathrm{cov}} \frac{2}{(n + \sqrt{2})^2} + (1 - \alpha_{\mathrm{cov}}) \min \left(1, \frac{2\mu - 1}{(n+2)^2 + \mu} \right) \quad . \tag{12}$$

Note, that for $\alpha_{\mathrm{cov}} = 1$ the original c_{cov} value is recovered.

Next, Equations (10) and (11) are analyzed in order to motivate the coefficients $\frac{1}{\mu}$ in Equation (10) and α_{cov} in conjunction with $(1 - \alpha_{\mathrm{cov}})$ in Equation (11). To this effect, two equivalent selection models can be considered: First, one can assume random selection, that is, the selection is independent of the realized $\boldsymbol{z}$-vectors. This selection model tests the stationarity of parameters, here the elements of $\boldsymbol{C}^{(g)}$, and reveals systematic (i.e., non random) drifts unrelated to selection. We assume such drifts to be undesirable in general. Second, one can assume $\boldsymbol{C}^{(g)}$ to be optimal in the sense that the *selected* vectors $\boldsymbol{B}\boldsymbol{D}\boldsymbol{z}_i, i \in I_{sel}$, are $\mathcal{N}(\boldsymbol{0}, \boldsymbol{C}^{(g)})$ distributed just as before selection. This model is suitable when the covariance matrix is properly adapted and it represents a fix point of the adaptation iteration. Under both selection assumptions, the symmetric matrix

$$\sum_{i \in I_{sel}} \boldsymbol{z}_i (\boldsymbol{z}_i)^{\mathrm{T}} = \sum_{i \in I_{sel}} \begin{pmatrix} z_{i1}^2 & z_{i1} z_{i2} & \cdots & z_{i1} z_{in} \\ z_{i2} z_{i1} & z_{i2}^2 & \cdots & z_{i2} z_{in} \\ \vdots & \vdots & \ddots & \vdots \\ z_{in} z_{i1} & z_{in} z_{i2} & \cdots & z_{in}^2 \end{pmatrix} \tag{13}$$

from Equation (10) has diagonal elements that are χ_μ^2 distributed and off-diagonal elements with expectation zero (Grimmett and Stirzaker 1998). With $\mathrm{E}\left[\sum_{i \in I_{sel}} z_{ij}^2\right] = \mu$, $j = 1, \ldots n$, we have that

$$\mathrm{E}\left[\boldsymbol{Z}^{(g+1)}\right] = \frac{1}{\mu} \boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} \mu \boldsymbol{I} \left(\boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} \right)^{\mathrm{T}} = \boldsymbol{C}^{(g)} \tag{14}$$

under the given selection models. Equation (14) is the reason for choosing the coefficient $\frac{1}{\mu}$ in Equation (10). With $\mathrm{E}\left[\boldsymbol{p}_c^{(g+1)}(\boldsymbol{p}_c^{(g+1)})^{\mathrm{T}}\right] = \boldsymbol{C}^{(g)}$ (Hansen 1998) and $\mathrm{E}\left[\boldsymbol{Z}^{(g+1)}\right] = \boldsymbol{C}^{(g)}$, we conclude from Equation (11) that $\mathrm{E}\left[\boldsymbol{C}^{(g+1)}\right] = \boldsymbol{C}^{(g)}$. This is the reason for choosing the coefficients $(1 - \alpha_{\mathrm{cov}})$ in conjunction with α_{cov} in Equation (11). Their sum is equal to one.

4 Strategy Parameter Evaluation

The reliability and adaptation speed of the covariance matrix adaptation procedure can be influenced by tuning the learning rate for the covariance matrix c_{cov}. To enable the comparison of algorithms with different α_{cov} values, c_{cov} has to be chosen appropriately.

The following procedure is used to find a suitable c_{cov} as a function of two variables $c_{\mathrm{cov}}(n, \mu)$: Experiments on ellipsoidal functions (f_{elli}, see Table 1) are performed where $n = [2, 3, 5, 10, 20, 40, 80]$, $\mu = [2, n, n^2]$, and $\lambda = 4\mu$. In each discrete point (n_i, μ_i), the number of function evaluations to reach f_{stop} is minimized as a function of the change rate c_{cov}, using nested intervals on the scaled parameter axis $\log \frac{1}{c_{\mathrm{cov}}}$. In our experience, this graph is unimodal and smooth and therefore easy to optimize. Using the number of function evaluations to reach f_{stop} as optimization criterion yields similar results as minimizing the adaptation time. Additionally, only the convergence phases (e.g. the rapid approaching to the optimum after the adaptation has taken place, see Figure 1) are measured and the convergence rates are comparatively insensitive to c_{cov} once adaptation has taken place. Measuring the adaptation time is more difficult to implement because the automatic identification when the adaptation is successfully completed is not straightforward.

Because the graph that depicts the needed number of function evaluations versus $\log \frac{1}{c_{\mathrm{cov}}}$ is considerably asymmetric (the left branch becomes infinitely steep), we set the appropriate c_{cov} to be a factor of two to three smaller than the optimum c_{cov}. This is the conservative and more reliable parameter choice. A function fitted through these appropriate $c_{\mathrm{cov}}(n_i, \mu_i)$ values roughly results in the default $c_{\mathrm{cov}} = \frac{2}{(n+\sqrt{2})^2}$ for $\alpha_{\mathrm{cov}} = 1$. For $\alpha_{\mathrm{cov}} = 0$, the function

$$c_{\mathrm{cov}} = \min\left(1, \frac{2\mu - 1}{(n+2)^2 + \mu}\right) \tag{15}$$

is found by trial and error and fits the points reasonably well. To get a closed expression for c_{cov}, we choose Equation (12) to set c_{cov}.

5 Test Functions

For the comparison of the two strategies, a test bed consisting of the functions shown in Table 1 is used. Initial values are set to $\langle \boldsymbol{x} \rangle_\mu^{(0)} = \mathbf{1}$ and $\sigma^{(0)} = 1$ for all functions except for Rosenbrock's case where $\langle \boldsymbol{x} \rangle_\mu^{(0)} = \mathbf{0}$ and $\sigma^{(0)} = 0.1$.

Note that the functions f_{elli}, f_{cigar}, f_{tablet}, and f_{twoax} have an axis ratio of $1 : 1000$, and f_{cigtab} has an axis ratio of $1 : 1000\ 0$.

Except for f_{diffpow} and f_{Rosen}, the functions as shown in Table 1 are completely separable. Except for f_{sphere}, they can be transformed into highly nonseparable functions with identical topology using an orthogonal coordinate system transformation. All results in this paper are independent of any orthogonal, i.e., angle preserving transformation of the coordinate system. Therefore, all functions, except for f_{sphere}, must

Name	Function	f_{stop}
Sphere	$f_{\text{sphere}} = \sum_{i=1}^{n} x_i^2$	10^{-10}
Ellipsoid	$f_{\text{elli}} = \sum_{i=1}^{n} 10^{6\frac{i-1}{n-1}} x_i^2$	10^{-10}
Cigar	$f_{\text{cigar}} = x_1^2 + \sum_{i=2}^{n} 10^6 x_i^2$	10^{-10}
Tablet	$f_{\text{tablet}} = 10^6 x_1^2 + \sum_{i=2}^{n} x_i^2$	10^{-10}
Cigar-Tablet	$f_{\text{cigtab}} = x_1^2 + \sum_{i=2}^{n-1} 10^4 x_i^2 + 10^8 x_n^2$	10^{-10}
Two-Axes	$f_{\text{two ax}} = \sum_{i=1}^{\lfloor n/2 \rfloor} 10^6 x_i^2 + \sum_{i=\lfloor n/2 \rfloor}^{n} x_i^2$	10^{-10}
Different powers	$f_{\text{diffpow}} = \sum_{i=1}^{n} \lvert x_i \rvert^{2+10\frac{i-1}{n-1}}$	10^{-15}
Rosenbrock	$f_{\text{Rosen}} = \sum_{i=1}^{n-1} (100 \cdot (x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	10^{-10}
Parabolic ridge	$f_{\text{parabR}} = -x_1 + 100 \sum_{i=2}^{n} x_i^2$	-10^{10}
Sharp ridge	$f_{\text{sharpR}} = -x_1 + 100 \sqrt{\sum_{i=2}^{n} x_i^2}$	-10^{10}

Table 1: Test functions and stopping criteria. While the first six functions are convex quadratic f_{diffpow}, f_{Rosen}, and f_{sharpR} differ remarkably from a convex quadratic model.

be interpreted as highly nonseparable. For an arbitrary orthogonal coordinate system transformation, which implements the actually nonseparable versions of these functions, the reader is referred to Hansen and Ostermeier (2001).

Tests are carried out in the dimensions $n = [2, 3, 5, 10, 20, 40, 80]$ and for parent numbers $\mu = [2, \lceil n/4 \rceil, \lceil n/2 \rceil, n, 2n, 4n, \lceil n^2/4 \rceil, \lceil n^2/2 \rceil, n^2]$ where the population size λ is set to 4μ. For $n \to \infty$ and $\lambda \to \infty$, one known theoretical optimum for μ is $\lambda/3.70$, while for $\lambda \not\ll n < \infty$ optimal values can be considerably smaller (Beyer 1996, 2001). To make the investigation as meaningful as possible, we choose the large but sensible $\mu = \lambda/4$.

6 Discussion of the Results

Three different strategy variants are presented: New-CMA, where $\alpha_{\text{cov}} = 0$; Orig-CMA, where $\alpha_{\text{cov}} = 1$; and Hybr-CMA, where $\alpha_{\text{cov}} = \frac{1}{\mu}$. Note that for $\mu = 1$, (i) Hybr-CMA is identical to Orig-CMA, and (ii) New-CMA is identical to Orig-CMA with $c_c = 1$ and a smaller c_{cov}. The simulation results for the various strategies were analyzed using two different perspectives:

Serial performance. We analyzed the total number of function evaluations to reach f_{stop}. This is the appropriate measure if optimization is performed on a single processor or on a small number of processors that does not exceed the smallest sensible population size λ. In this case the number of functions evaluations is an appropriate measure for the time to reach f_{stop}.

Parallel performance. We analyzed the number of generations to reach f_{stop}. This corresponds to the time complexity of the algorithm (Beyer 1996). When a larger number of processors is available, it becomes interesting to evaluate the time complexity, especially if λ is equal to or smaller than the number of processors. In this case, the number of generations is an appropriate measure for the time to reach f_{stop} in a single run.

We will first discuss single optimization runs as shown in Figure 1. All three strategies are shown on f_{sphere}, f_{cigar}, and f_{elli}, where $\mu = n = 10$, and $\lambda = 40$. The standard

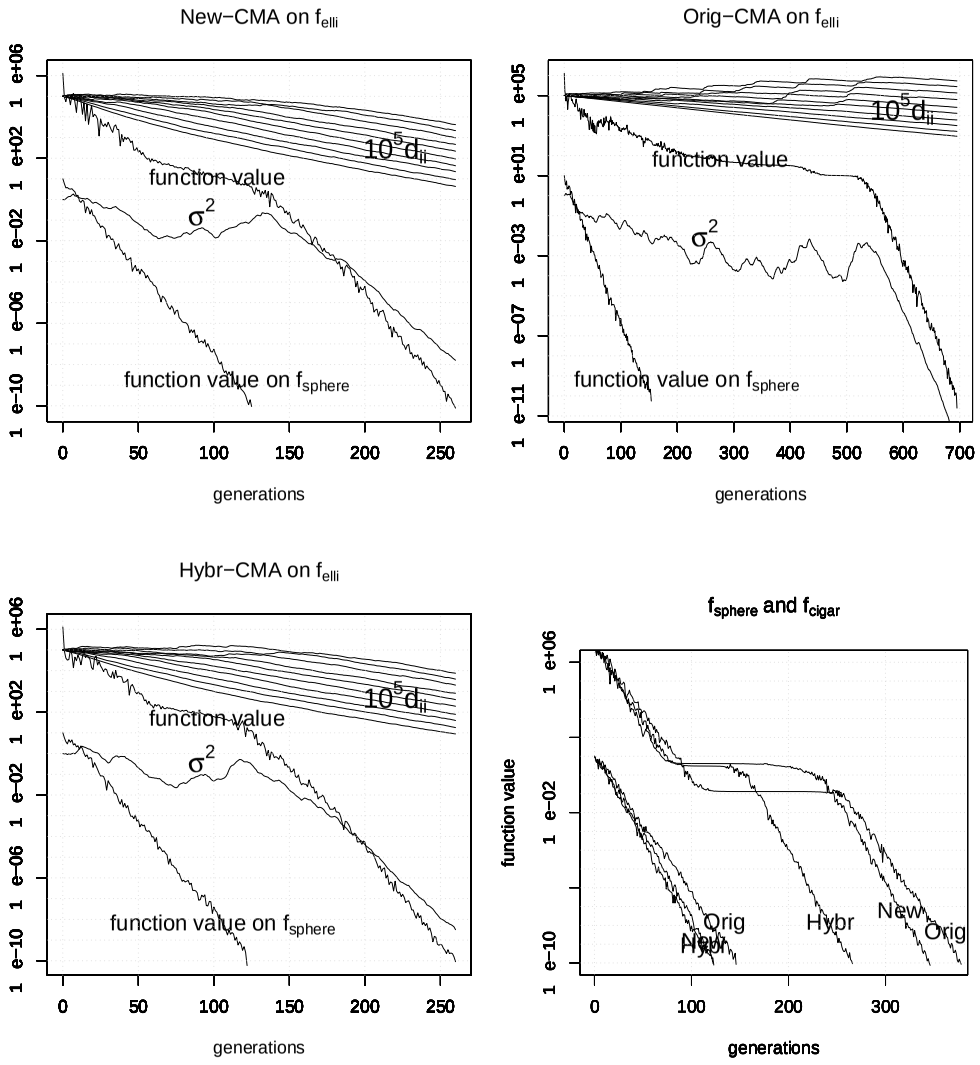

Figure 1: Single simulation runs of different $(10, 40)$-ESs on f_{sphere} and f_{cigar} (lower right) and on f_{elli} (other figures), where $n = 10$. Shown are the function values, ${\sigma^{(g)}}^2$ and the sorted diagonal elements of the matrix $\boldsymbol{D}^{(g)}$ multiplied by 10^5 (reflecting the main axis lengths of the mutation ellipsoid) versus the number of generations. Notice the different x-axis ranges. The same strategy without adaptation of the covariance matrix takes roughly 10^6 generations on f_{elli} and 10^7 generations on f_{cigar}.

deviations for the number of function evaluations needed to reach $f_{\text{stop}} = 10^{-10}$ are quite small[2] and all observed differences prove to be statistically significant by evaluating ten runs, while New-CMA and Hybr-CMA perform equally well on f_{sphere} and

[2]E.g., the one standard deviation intervals to reach f_{stop} for New-CMA and Orig-CMA on f_{cigar} (right most runs in the lower right of Figure 1) do not overlap.

f_{elli}.

The optimization using Orig-CMA on f_{elli} (upper right) shows a sharp transition from the adaptation phase to a convergence phase after 520 generations. During the following convergence phase, the same progress as on f_{sphere} is achieved. On New-CMA and Orig-CMA, the transition from adaptation to convergence is much less sharp. After about 150 generations, the progress becomes quite close to the progress on f_{sphere}; after about 200 generations, the adaptation is perfect, as revealed by the regular configuration of the graphs representing the diagonal elements d_{ii}. Comparing the runs on f_{elli} with the runs on f_{sphere}, the additional adaptation time is about 150 generations for New-CMA and Hybr-CMA while it is about 600 generations for Orig-CMA. The introduced modification reduces the adaptation time by a factor of four (for the given $n = \mu = 10$, and $\lambda = 40$). New-CMA and Hybr-CMA perform equally well.

On f_{cigar} (lower right of Figure 1), Hybr-CMA outperforms the other strategies, as in this case it combines their advantages: Compared to Orig-CMA, it uses additional vectors of the same generation for the update of the covariance matrix $\boldsymbol{C}$. Compared to New-CMA, it uses additional information about correlations between vectors of past generations (i.e., the evolution path) for the update of $\boldsymbol{C}$. Either of these additions reduces the adaptation time by a factor of roughly 1.5 (for the given $n = \mu = 10$, and $\lambda = 40$). Note that the level of the plateaus where the function values remain constant for a while is independent of the strategy variant (and has a comparatively large variance). Orig-CMA shows a slightly slower convergence speed in the convergence phase than Hybr-CMA and New-CMA for both f_{sphere} and f_{cigar}. This difference increases with increasing μ and will be discussed in the next section, where we evaluate the performance differences on f_{sphere} in detail.

6.1 Serial Performance on Sphere

The differences in serial performance between Orig-CMA and Hybr-CMA on f_{sphere} are shown in Figure 2, upper left.[3] For $\lambda < 10n$ the performance of Orig-CMA and Hybr-CMA are similar as expected. For larger population sizes Hybr-CMA becomes faster than Orig-CMA by a factor of up to three. This effect cannot be attributed to a faster adaptation of the distribution *shape* because on f_{sphere} the shape is already optimal at the beginning. The reason for the better performance of Hybr-CMA is the faster adaptation of the overall step length, i.e., the overall variance of the distribution. Originally, the cumulative path length control (Equations (7) and (8)) facilitates the adaptation of the global step size, that is, the overall step length. The parameter that tunes the adaptation speed (d_σ in Equation (8)) was (i) chosen conservatively resulting in a somewhat slower but more robust algorithm (Hansen and Ostermeier 2001, Section 5.1) and (ii) chosen for small population sizes that realize smaller progress rates per generation than larger populations and therefore demand slower adaptation rates. Consequently, for $\mu > n$ and $\alpha_{\text{cov}} \ll 1$ the distribution adaptation in Equation (11) can successfully contribute to the adaptation speed of the overall step length in Hybr-CMA, because the change rate $c_{\text{cov}} \not\ll \frac{1}{n}$. This is the presumable reason for the observed speed-up on f_{sphere}.

Even though this effect appears to be advantageous at first sight, it is in principle disadvantageous if the distribution adaptation significantly influences the magnitude

[3] Note that one expects linear graphs for any function that can be expressed by $\alpha\, n^\beta$, where α, β constant. β determines the slope of the graph and α the vertical position. For $\beta = 1$ (linear scaling) and $\beta = 2$ (quadratic scaling), dotted lines are drawn in all figures. When we interpret the slopes, we neglect the irregular effects observed in small dimensions.

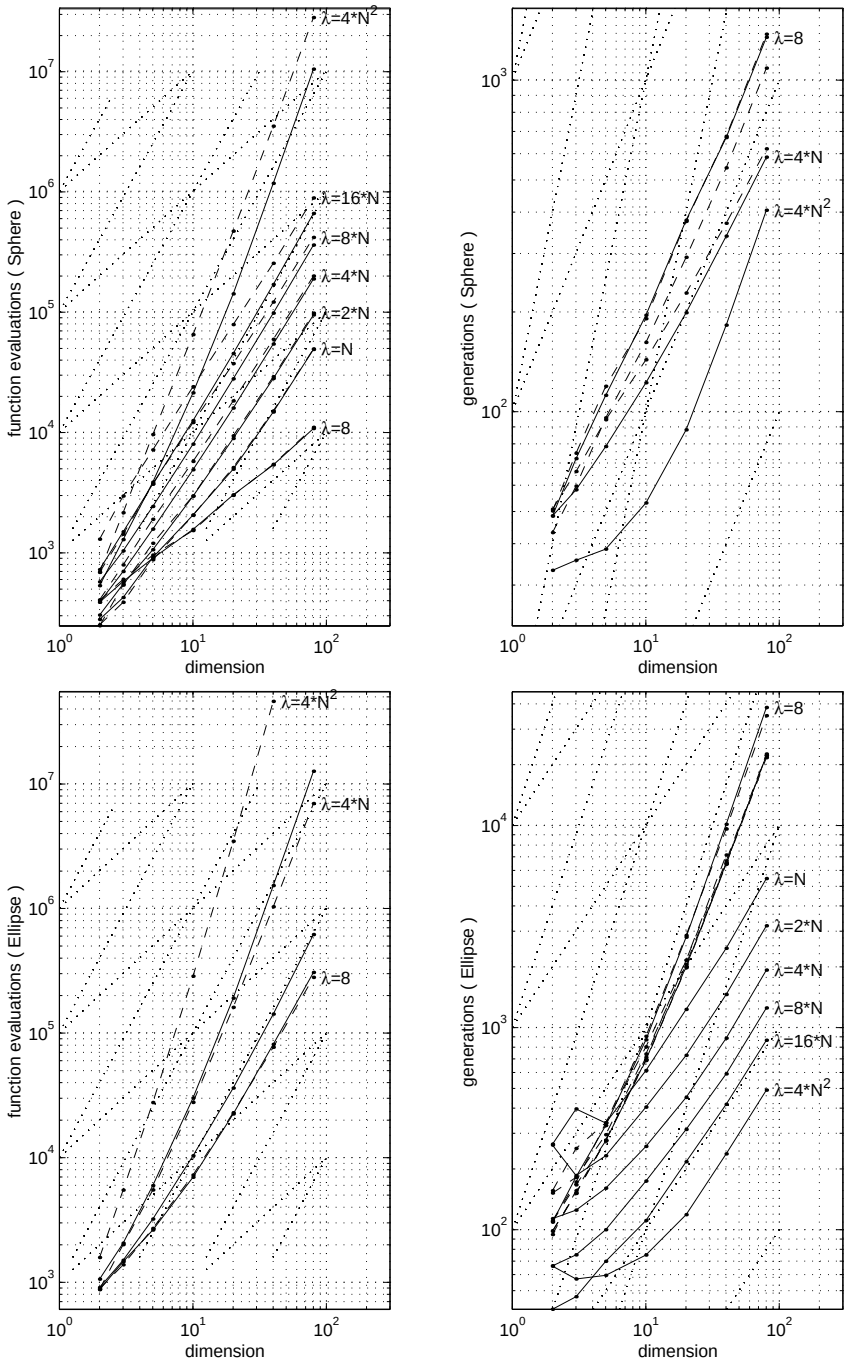

Figure 2: Number of function evaluations (left) and number of generations (right) versus the problem dimension for the sphere function (above) and the ellipse function (below). The Orig-CMA (– – –) and the Hybr-CMA (—) are plotted for $\lambda = 8, n, 2n, 4n, 8n, 16n, 4n^2$ if curves are far apart, or for $\lambda = 8, 4n, 4n^2$.

of the overall step length. While the path length control is shown to adapt nearly optimal step lengths even for $\mu > 1$ (at least for $\lambda < n$, Hansen 1998), the distribution adaptation as described in Equation (11) acquires small step lengths too rapidly, if *the optimal* step length remains constant over time (i.e., in a stationary environment unlike on f_{sphere}). When the optimal step length decreases over time, as on f_{sphere}, this can be an advantage. However, it is not desirable, in general, to use an algorithm that adapts much too small variances. This suggests an upper limit for sensible population sizes for Hybr-CMA, arguably at $\lambda \approx 10n$.

6.2 Serial Performance on Other Functions

In this section, we discuss the serial performance of the strategy variants on the remaining functions. Comparing New-CMA with the other strategy variants where $\lambda = 8$ reveals a significant difference on f_{cigar} (Figure 3). While Orig-CMA and Hybr-CMA need about $500n$ function evaluations to reach f_{stop} (compare Figure 4, upper left), New-CMA needs about $120n^2$ function evaluations. Using the evolution path $\boldsymbol{p}_c$ in Equation (11) in addition with an accumulation parameter of $c_c \propto \frac{1}{n}$ in Equation (5) yields this impressive speed-up of Orig-CMA and Hybr-CMA. Although detected in earlier investigations (Hansen and Ostermeier 2001), this speed-up is noteworthy in that a completely adaptable covariance matrix with $\frac{n^2+n}{2}$ free parameters can be adapted to certain topologies in $\mathcal{O}(n)$ function evaluations. Since this observation on f_{cigar} is the only major difference between Hybr-CMA and New-CMA that we have seen, the latter algorithm is excluded from the remaining discussion.

Comparing the serial performance of Orig-CMA and Hybr-CMA, the most prominent effect is the dependency on λ. The smallest λ, $\lambda = 8$, (and even a smaller λ for $n = 5$) performs best in all cases. For example, if $n = 20$ the decline of the serial performance between $\lambda = 8$ and $\lambda = 80$ amounts roughly to a factor between 3 (e.g., on f_{sharpR}) and 10 (e.g., on f_{diffpow}) for Orig-CMA and between 1.2 (e.g., on $f_{\mathrm{two\,ax}}$) and 4 (on f_{cigar}) for Hybr-CMA. These results clearly favor Hybr-CMA and presumably they change in favor of larger values of λ for noisy functions.

Considering the serially optimal $\lambda = 8$, the performance difference between Orig-CMA and Hybr-CMA is small. For more than half of the functions, the difference is less than 15%. Differences of up to 50% can be observed on f_{tablet}, f_{diffpow}, f_{sharpR}, f_{twoax}, and f_{parabR}, but only on the latter two is Orig-CMA faster. This leads to the conclusion that the test suite is not able to reveal significant differences in the overall serial performance for $\mu = 2$ and $\lambda = 8$, but reveals slightly different performance profiles.

While the differences between Orig-CMA and Hybr-CMA appear to be marginal for $\lambda = 8$, the picture changes—in favor of Hybr-CMA—when the population size is increased. For $\lambda = 8$, the scaling of the needed function evaluations with respect to the problem dimension (i.e., the slope of the graphs) is linear on f_{sphere}, f_{parabR} and f_{cigar}, but it is usually subquadratic and at most quadratic as on f_{sharpR}. For $\lambda \propto n$, the scaling of Hybr-CMA becomes nearly quadratic, regardless of whether the scaling is linear or quadratic for $\lambda = 8$ (see in particular f_{sharpR} and f_{diffpow}). This is in contrast to Orig-CMA where the scaling *always* deteriorates when λ is increased to $\lambda \propto n$ and can even become cubic as on f_{diffpow}. As a result, Hybr-CMA never performs worse (and often much better) than Orig-CMA when $\lambda \propto n$. This clear advantage can of course be expected only if $\mu \propto \lambda$, e.g., $\mu \approx \lambda/4$, as chosen in our investigations. Concluding these observations, Hybr-CMA has an undoubtedly better serial performance than Orig-CMA if $\mu \geq 5$.

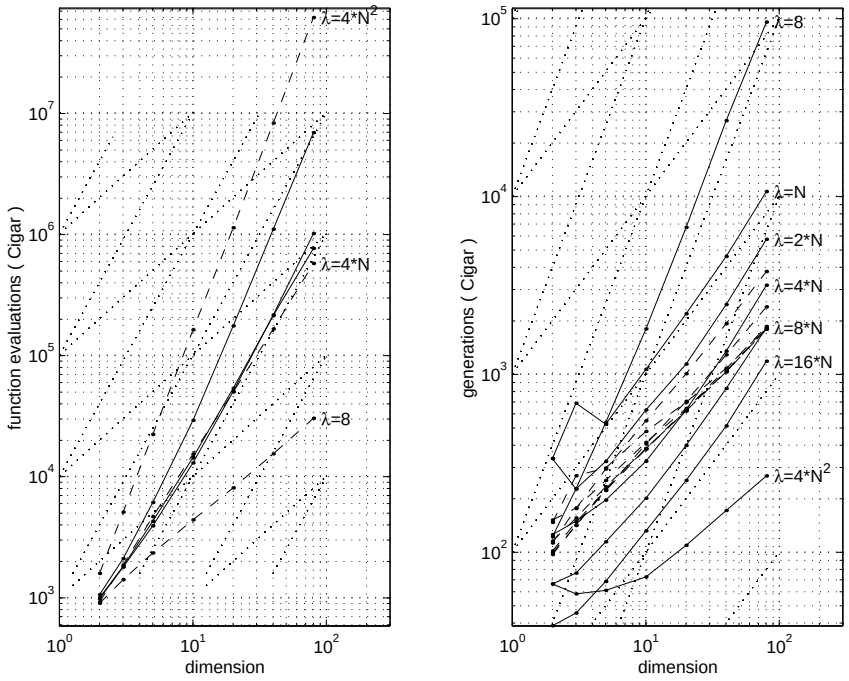

Figure 3: Number of function evaluations (left) and number of generations (right) versus the problem dimension for the cigar function. The Orig-CMA ($- - -$) and the New-CMA (—) are plotted for $\lambda = 8n, 2n, 4n, 8n, 16n, 4n^2$.

6.3 Parallel Performance

Parallel performance comes into play if λ is made considerably large. For this reason, we concentrate the discussion of parallel performance on the cases where $\lambda \propto n$ and $\lambda = 4n^2$. The differences in performance between Orig-CMA and Hybr-CMA discussed above translate to the parallel case. Hybr-CMA outperforms Orig-CMA on every test function, if $\lambda \gg 8$ (assuming $\mu \approx \lambda/4$). The parallel performance of Orig-CMA does not change dramatically when λ is increased. The improvement never exceeds a factor of two in dimensions up to 80, with one exception: On f_{sharpR}, the scaling of the number of generations with the problem dimension improves with increasing λ (Figure 7).[4] Consequently, for $n = 80$ the performance improvement is already far more than one order of magnitude. This is a little surprising because the adaptation rate c_{cov} is constant with respect to the generation number. We ascribe this effect to the larger step sizes that are adapted for larger μ in conjunction with a smaller distance to the ridge peak due to the intermediate recombination. The larger the relation between the mean step length and the mean distance to the ridge peak, i.e., the larger $\sigma^{(g)}/\sqrt{\sum_{i=2}^{n} x_i^2}$, the faster the distribution should be adapted.

In contrast to Orig-CMA, where the parallel performance is less dependent on λ, Hybr-CMA shows a vigorous improvement when λ is increased. For example, increasing λ from n to $8n$ improves the parallel performance by a factor greater than four on f_{elli}, f_{diffpow}, f_{twoax} and f_{sharpR} (increasing in improvement order). Note that we could

[4] The unsteady course of the graphs reveals the large variance of these simulations.

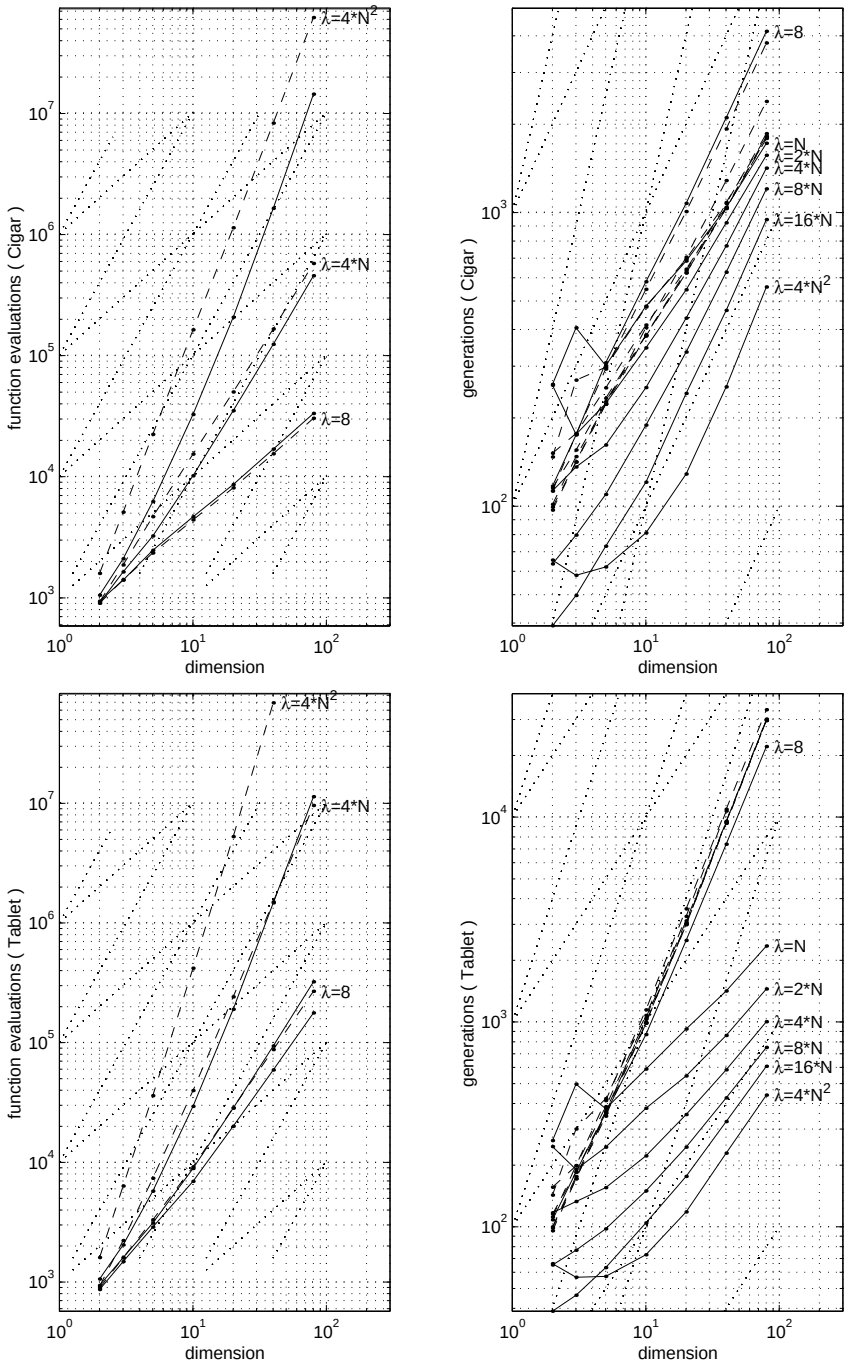

Figure 4: Number of function evaluations (left) and number of generations (right) versus the problem dimension for the cigar function (above) and the tablet function (below). The Orig-CMA (– – –) and the Hybr-CMA (—) are plotted for $\lambda = 8, n, 2n, 4n, 8n, 16n, 4n^2$ if curves are far apart, or for $\lambda = 8, 4n, 4n^2$.

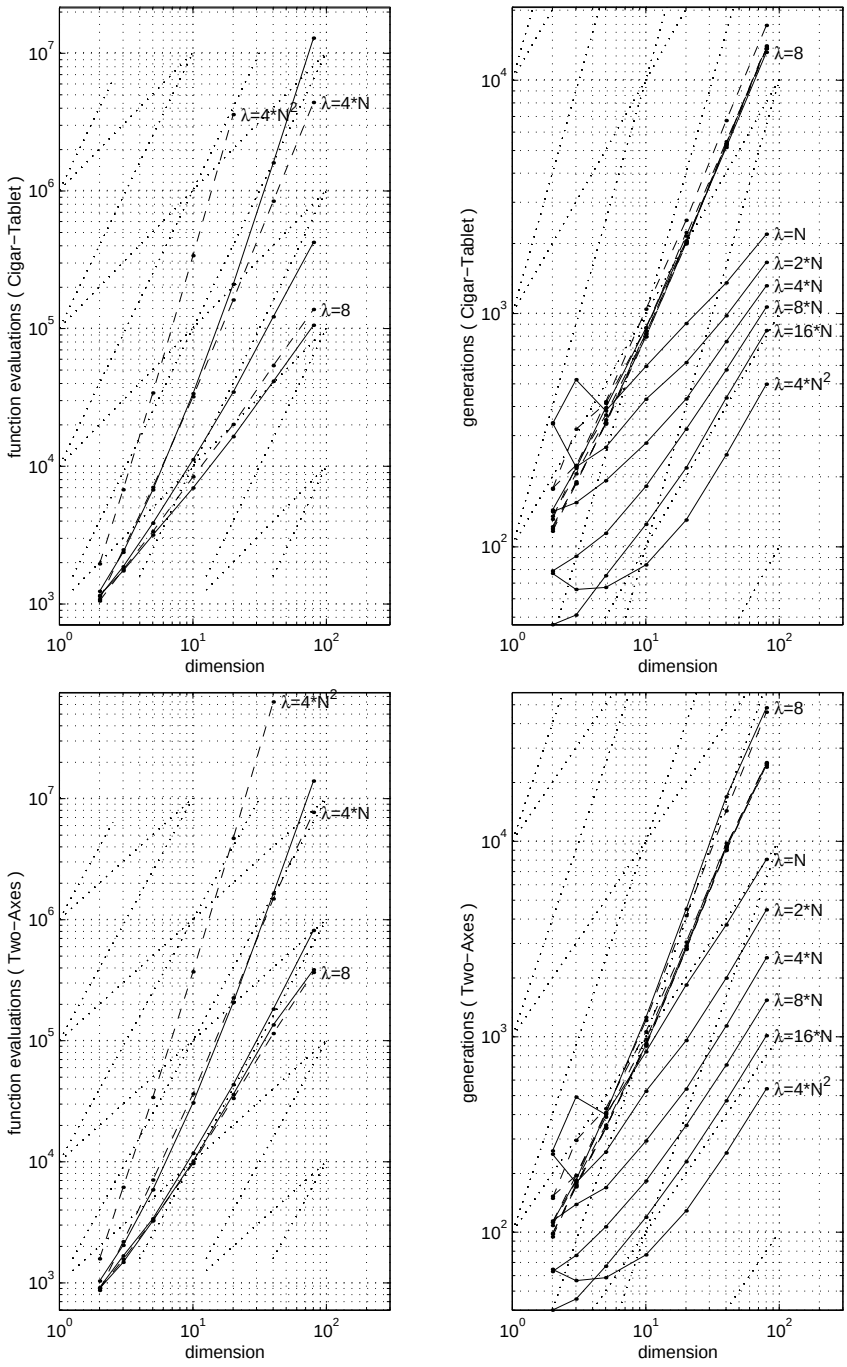

Figure 5: Number of function evaluations (left) and number of generations (right) versus the problem dimension for the cigar-tablet function (above) and the two-axes function (below). The Orig-CMA ($- \; - \; -$) and the Hybr-CMA (—) are plotted for $\lambda = 8, n, 2n, 4n, 8n, 16n, 4n^2$ if curves are far apart, or for $\lambda = 8{,}4\ n, 4n^2$.

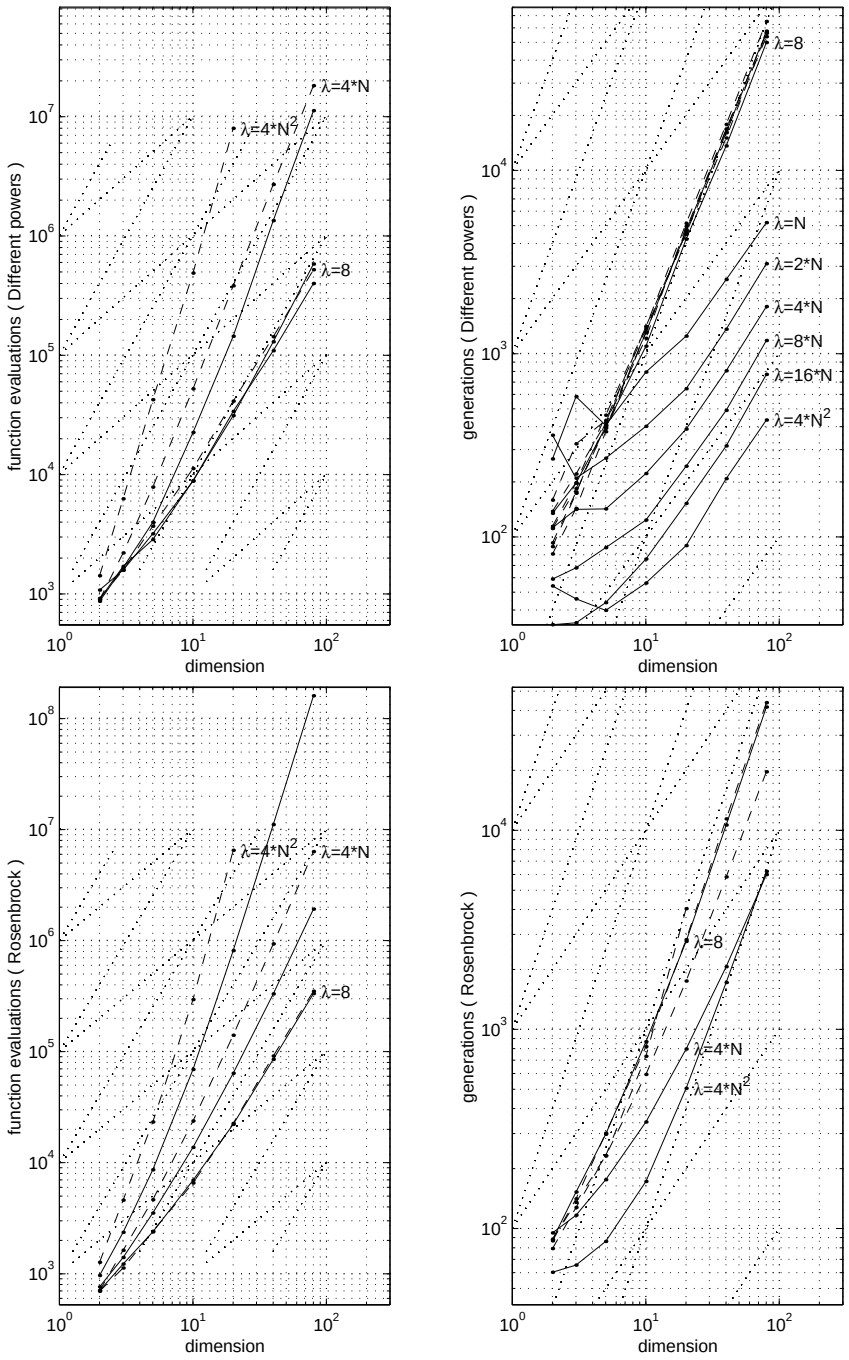

Figure 6: Number of function evaluations (left) and number of generations (right) versus the problem dimension for the different powers function (above) and the Rosenbrock's function (below). The Orig-CMA ($- - -$) and the Hybr-CMA (—) are plotted for $\lambda = 8$,n , $2n, 4n, 8n, 16n, 4n^2$ if curves are far apart, or for $\lambda = 8, 4n, 4n^2$.

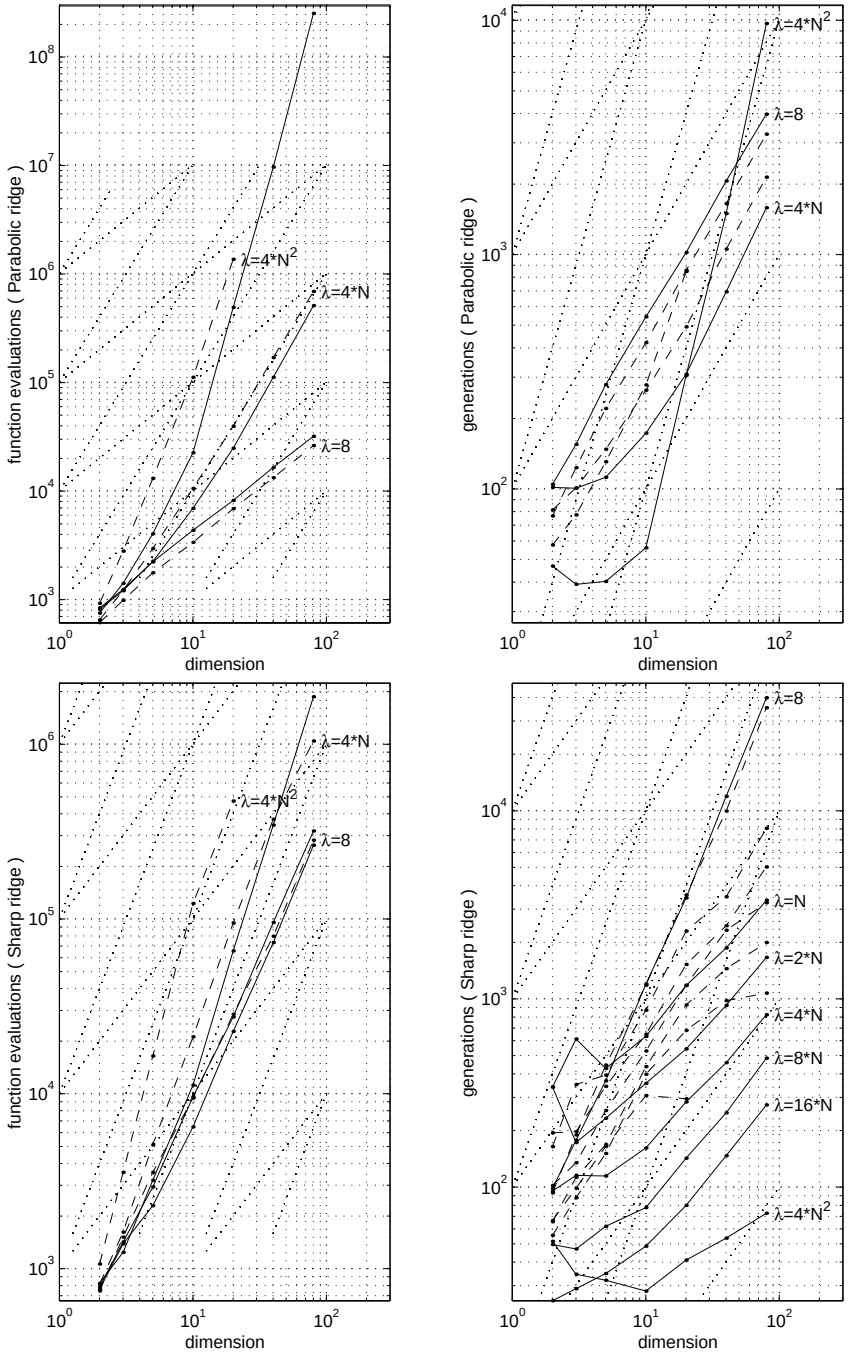

Figure 7: Number of function evaluations (left) and number of generations (right) versus the problem dimension for the parabolic ridge function (above) and the sharp ridge function (below). The Orig-CMA ($- - -$) and the Hybr-CMA (—) are plotted for $\lambda = 8, n, 2n, 4n, 8n, 16n, 4n^2$ if curves are far apart, or for $\lambda = 8{,}4\ n, 4n^2$.

never reasonably expect an improvement by a factor greater than eight in this example.

However, the most impressive result concerns the parallel performance of Hybr-CMA where $\lambda \propto n$. When $\lambda \leq 10n$, in Orig-CMA the number of generations scales (nearly) quadratically with n on all functions except on f_{sphere}. The same holds for Hybr-CMA for $\lambda = 8$. However, when λ is increased to be proportional to n in Hybr-CMA, the number of generations scales *linearly* with the problem dimension for convex quadratic test functions. The scaling is even slightly sublinear for f_{sphere}, f_{tablet}, and f_{sharpR} and it is (slightly) worse than linear on f_{parabR} and f_{Rosen} (and arguable on f_{diffpow}), but we have no good explanation for these differences. Nevertheless, the roughly linear scaling for $\lambda \propto n$ on most test functions is the main improvement of the new Hybr-CMA compared to Orig-CMA.

7 Conclusions

We have presented a highly parallel evolutionary algorithm derived from the derandomized evolution strategy with covariance matrix adaptation. In order to exploit the often emphasized feature of evolution strategies being easily parallelizable, we devised a technique that can optimize in fewer number of generations than the original strategy.

Reviewing the results from Section 6, we have achieved our goal for population sizes up to $\lambda = 10n$. Choosing $\alpha_{\mathrm{cov}} = \frac{1}{\mu}$ and $\mu \approx \lambda/4$, the proposed algorithm seems to efficiently exploit the information prevalent in the population and it reveals mainly linear time complexity for population sizes proportional to n and up to $10n$, if fully parallelized. This implies that we were able to reduce the time complexity of the adaptation from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$.

For $\mu = 1$, the expanded algorithm is identical with the original one. With increasing μ, the performance improves remarkably compared to the original algorithm. In our tests, the expanded algorithm, with $\alpha_{\mathrm{cov}} = \frac{1}{\mu}$, reveals no serious disadvantage compared to the original one. For population sizes larger than $10n$ the adjustment of the overall step length becomes problematic. Two problems can be identified.

- For large population sizes and large adaptation rates c_{cov} the covariance matrix adaptation contributes remarkably to the change of the overall step length. Unfortunately, the covariance adaptation mechanism tends to adapt a far too small overall variance, if $\mu \gg 1$. This effect could be avoided by choosing smaller c_{cov} values, at the cost of a slower adaptation of the distribution shape. Alternatively, the covariance matrix $\boldsymbol{C}$ could be renormalized after each update to prevent the matrix elements from diminishing continuously. Even though it is not elegant, this method may be effective.

- The cumulative path length control ceases to work properly when the population size becomes too large. As the population size increases, the possible rate of change of the overall step length should also increase in order to realize the optimal progress rates, e.g., on the sphere function. Within small populations change rates can be achieved as fast as necessary on the sphere function, if $d_\sigma \approx \frac{1}{c_\sigma}$ and $\sqrt{\frac{n}{2}} < \frac{1}{c_\sigma} < n$. Faster changes rates do not seem to be possible because the momentum term $\boldsymbol{p}_\sigma$ tends to become unstable for smaller values of d_σ. Even worse, a larger population size itself appears to have a similar destabilizing effect that would suggest choosing d_σ to be even larger, resulting in an even slower change rate. At the moment, we have no explanation for this behavior and no solution at hand.

This suggests replacing the cumulative path length control for population sizes larger than $10n$. However, as it turns out, it is difficult to find a reasonable adaptation mechanism even for smaller population sizes if $\mu > 1$ and if intermediate multi-recombination is applied.[5] To our knowledge, the only solution is a hierarchical approach, in which a few populations with different step sizes run in parallel. However, the question remains open as to the size of the change rates that can actually be achieved with this approach.

Future work could focus on the principle limitations of any implementable global step size control in dependency of μ. It would be interesting to exploit theoretical results that reveal the limits of what we *can* achieve when searching for better adaptation mechanisms for large μ.

Acknowledgments

This work was supported by the Swiss National Science Foundation under grant 21-54093.98. SM wishes to thank Sebastian Müller, ETH Zürich, for inspiring discussions.

References

Bäck, T., Hammel, U., and Schwefel, H.-P. (1997). Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1(1):3–17.

Beyer, H.-G. (1996). On the asymptotic behavior of multirecombinant evolution strategies. In Voigt, H.-M., Ebeling, W., Rechenberg, I., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature—PPSN IV, Proceedings*, pages 122–133, Berlin. Springer Verlag.

Beyer, H.-G. (2001). *The Theory of Evolution Strategies*. Natural Computing Series. Springer Verlag, Heidelberg. ISBN 3-540-67297-4.

Grimmett, G. and Stirzaker, D. (1998). *Probability and Random Processes*. Clarendon Press, Oxford. Second Edition.

Hansen, N. (1998). *Verallgemeinerte individuelle Schrittweitenregelung in der Evolutionsstrategie. Eine Untersuchung zur entstochastisierten, koordinatensystemunabhängigen Adaptation der Mutationsverteilung*. Mensch und Buch Verlag, Berlin. ISBN 3-933346-29-0.

Hansen, N. and Ostermeier, A. (1997). Convergence properties of evolution strategies with the derandomized covariance matrix adaptation: The $(\mu/\mu_{\mathrm{I}}, \lambda)$-CMA-ES. In Zimmermann, H.-J., editor, *EUFIT'97, 5th Europ. Congr. on Intelligent Techniques and Soft Computing, Proceedings*, pages 650–654, Aachen, Germany. Verlag Mainz.

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

Ostermeier, A. (1997). *Schrittweitenadaptation in der Evolutionsstrategie mit einem entstochastisierten Ansatz*. PhD thesis, Technische Universität Berlin.

Schwefel, H.-P. (1995). *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology Series. John Wiley & Sons Inc., New York.

[5] We prefer *intermediate* (multi-)recombination, because it is independent of the coordinate system.