A Generalization of the CMA-ES Algorithm for Functions with Matrix Input

Simon Konzett
Faculty of Mathematics, University of Vienna, Oskar-Morgenstern-Platz 1, Vienna, Austria

Keywords: Evolutionary strategies, Evolutionary algorithms, Adaptations, High-dimensional, covariance matrix, Mutation distribution, Self-adaptation.

Abstract: This paper proposes a novel modification to the covariance matrix adaptation evolution strategy (CMA-ES) introduced by (Hansen and Ostermeier, 1996) under a special problem setting. In this paper the case is considered when the function which has to be optimized takes a matrix as input. Here an approach is presented where without vectorizing directly matrices are sampled and column and row-wise covariance matrices are adapted in each iteration of the proposed evolution strategy to adapt the mutation distribution. The method seems to be able to capture correlations in the entries of the considered matrix and adapt the corresponding covariance matrices accordingly. Numerical tests are performed on the proposed method to show advantages and disadvantages.

1 INTRODUCTION

Evolution strategies are used to minimize non-linear objective functions mapping usually a subspace $X \subseteq \mathcal{R}^n$ to $\mathcal{R}$. The iterative strategy is to select good points and then variate these points in the most promising way to evolve towards a population of points with higher quality. Selection is done by comparing function values of the points in each generation. Variation means suitable recombination of the population found so far and to mutate these points to gain more insight. Mutation can simply mean adding normally distributed random vectors but often more sophisticated methods are necessary. Mostly the mutation distribution is characterized by a covariance matrix which should reflect shape and size of the distribution. Several approaches have been proposed to adapt these covariance matrices in a promising way. One of the first ideas proposed is by (Schwefel, 1977). However in this work a modification of the CMA-ES (Covariance Matrix Adaption Evolution Strategy) by (Hansen and Ostermeier, 1996), (Hansen, 1998) and (Hansen and Ostermeier, 2001) is proposed. The CMA-ES collects information about successful search steps and stores this information in so-called evolution paths. The gained information is used to adapt the covariance and to slowly derandomise the mutation distribution. Until now there have been many modifications of the CMA-ES proposed as in (Jastrebski and Arnold, 2006), (Igeland et al., 2007), (Igel et al., 2006) and many more.

A modification of the CMA-ES is proposed concerning functions $f : \mathcal{R}^{n\times m} \to \mathcal{R}$ with matrices as input. This kind of problem appeared during my research. In particular the task was to determine suitable parameter matrices for a state space model. Obviously the original CMA-ES method can treat this kind of problem by just vectorizing the input matrix although the dimension of such a problem gets quite high soon for reasonably large matrices. In higher dimensions the original CMA-ES method need large covariance matrices to characterize the mutation distribution and as a result the adaptation for these covariance matrices should be done carefully and slowly. Same as in the original CMA-ES the new method will make use of past successful steps and adapts covariance matrices related to the rows respectively columns of the matrix valued mutation distribution. The matrices to adapt in my proposed method are much smaller and so on we hope that the method is more flexible and faster in certain cases.

The paper is from now on organised as in the next section the original CMA-ES is briefly discussed as in (N. Hansen, 2011). Then in section 3 the proposed modification of the CMA-ES is described. Section 4 then shows some computational results and section 5 gives a short review and conclusion.

Konzett S..
A Generalization of the CMA-ES Algorithm for Functions with Matrix Input.
DOI: 10.5220/0005159703370342
In *Proceedings of the International Conference on Evolutionary Computation Theory and Applications* (ECTA-2014), pages 337-342
ISBN: 978-989-758-052-9

2 CMA-ES

In this section a brief outline of the CMA-ES is given. The problem to solve is generally

$$\min_{\mathbf{x}\in\mathcal{R}^n} f(\mathbf{x}) \tag{1}$$

where f is a non-linear real-valued function.

The algorithm needs initially a starting point $\mathbf{m}_0 \in \mathcal{R}^n$ and a characterization of the initial mutation distribution given by a covariance matrix $\mathbf{C}_0 \in \mathcal{R}^{n\times n}$ and an initial global step length $\sigma_0 > 0$. Then the algorithm in each iteration is the following.

First a new sample of $\lambda > 0$ candidate points is taken

$$\begin{aligned} \mathbf{z}_k &\sim \mathcal{N}(\mathbf{0},\mathbf{I}) && (2)\\ \mathbf{y}_k &= \mathbf{B}\mathbf{D}\mathbf{z}_k \sim \mathcal{N}(\mathbf{0},\mathbf{C}) && (3)\\ \mathbf{x}_k &= \mathbf{m}+\sigma\mathbf{y}_k \sim \mathcal{N}(\mathbf{m},\sigma^2\mathbf{C}) && (4)\end{aligned}$$

where $\mathbf{m}$ is the best point at this time, the matrices $\mathbf{B}$ and $\mathbf{D}$ are given by an eigendecomposition of the covariance matrix $\mathbf{C} = \mathbf{B}\mathbf{D}\mathbf{D}\mathbf{B}^T$ and the value $\sigma > 0$ is the global step length at this time. So one has a set of sample points $\{\mathbf{x}_k\}_{k=1,\ldots,\lambda}$ and these points are compared to each other. The indices of the points $\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k$ are reordered such that $f(\mathbf{x}_k) \leq f(\mathbf{x}_{k+1})$ and a new best point is determined by

$$\mathbf{m} = \mathbf{m} + \sigma\sum_{i=1}^{\mu}\mathbf{w}_k\mathbf{y}_k = \mathbf{m} + \sigma\langle\mathbf{y}\rangle_w \tag{5}$$

where $\mu \leq \lambda$ is the parental population size and $\mathbf{w}_k$ are weights such that better points have more influence on the adaptation.

Second the global step size control is done

$$\begin{aligned} \mathbf{p}_\sigma &= (1-c_\sigma)\mathbf{p}_\sigma + \\ &+ \sqrt{c_\sigma(2-c_\sigma)\mu_{\text{eff}}}\mathbf{B}\mathbf{D}^{-1}\mathbf{B}^T\langle\mathbf{y}\rangle_w && (6)\\ \sigma &= \sigma\cdot\exp\left(\frac{1}{d}\left(\frac{\|\mathbf{p}_\sigma\|}{\mathbb{E}(\mathcal{N}(\mathbf{0},\mathbf{I}))}-1\right)\right) && (7)\end{aligned}$$

where $\mathbf{p}_\sigma$ is called the conjugate evolution path. The conjugated evolution path is normalized such that $\mathbf{p}_\sigma \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ under purely random selection and then its length is compared with the expected length under purely random selection. For more details about the normalization and the step length adaptation I reference to (Hansen and Ostermeier, 1996), (Hansen, 1998), (Hansen and Ostermeier, 2001) or (N. Hansen, 2011). Basically the idea is that if good steps $\mathbf{y}_k$ point in a similar direction the accumulation of these points leads to a larger path length $\|\mathbf{p}_\sigma\|$ compared with its expectation $\left\|\mathbb{E}(\mathcal{N}(\mathbf{0},\mathbf{I}))\right\|$ under purely random selection and as a consequence the global step length $\sigma > 0$ shall grow. Similarly if good steps $\mathbf{y}_k$ cancel each other out the global step length $\sigma > 0$ gets smaller. The real parameter $d > 0$ controls how fast the global step length can change.

The last part in each iteration of the algorithm is the covariance adaptation,

$$\begin{aligned} \mathbf{p}_c &= (1-c_c)\mathbf{p}_c + \sqrt{c_c(2-c_c)\mu_{\text{eff}}}\langle\mathbf{y}\rangle_w && (8)\\ \mathbf{C} &= (1-c_1-c_\mu)\mathbf{C} + \\ &+ c_1(\mathbf{p}_c\mathbf{p}_c^T) + c_\mu\sum_{k=1}^{\mu}\mathbf{y}_k\mathbf{y}_k^T. && (9)\end{aligned}$$

The evolution path $\mathbf{p}_c$ is accumulated by the non-normalized successful steps and the covariance matrix $\mathbf{C}$ is updated by a rank-one update of the evolution path $\mathbf{p}_c$ and a rank-μ update of the selected steps $\mathbf{y}_k$. If the covariance matrix $\mathbf{C}$ is too large to perform eigendecompositions then equation (9) can be replaced by

$$\begin{aligned} \mathbf{C} &= (1-c_1-c_\mu)\mathbf{C} + \\ &+ c_1\,\text{diag}(\mathbf{p}_c^2) + c_\mu\,\text{diag}\left(\sum_{k=1}^{\mu}\mathbf{y}_k^2\right) && (10)\end{aligned}$$

where $\text{diag}(\mathbf{x})$ denotes the diagonal matrix with the vector $\mathbf{x}$ in its diagonal and the square of a vector is meant component-wise here.

The change rates $c_\sigma, c_c, c_1, c_\mu > 0$ are assumed to be chosen appropriately. In general it can be said that the choice of suitable changing rates is crucial for the success of the algorithm. Larger change rates allow better adaptation and fast change but there are restrictions due to the finite amount of information which is gained through selection. If the change rates are too large the procedure gets random and if they are too small the system is rigid and adapts too slowly which slows the algorithm down. A discussion about suitable choices of these parameters as well as a discussion about a suitable population $\lambda > 0$ size can be found in (Hansen and Ostermeier, 1996), (Hansen, 1998) and (Hansen and Ostermeier, 2001). The parameter $\mu_{\text{eff}} > 0$ is determined such that $\mathbf{p}_\sigma \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ and $\mathbf{p}_c \sim \mathcal{N}(\mathbf{0},\mathbf{C})$ under purely random selection. It is easy to see that the adaptation of the mutation distribution is based on the path of successful steps usually called evolution path.

In the next section the procedure is generalized for matrices instead of vectors as points of the population.

3 A GENERALIZATION FOR MATRICES

As the outline of the algorithm remains the same first a matrix distribution and a sampling procedure need

to be introduced. The random matrix

$$\mathbf{X} \sim \mathcal{MN}(\mathbf{M}, \mathbf{U}, \mathbf{V}) \tag{11}$$

is is said to be matrix normally distributed where $\mathbf{M} \in \mathcal{R}^{m \times n}, \mathbf{U} \in \mathcal{R}^{m \times m}$ and $\mathbf{V} \in \mathcal{R}^{n \times n}$. The distribution is defined by

$$\text{vec}(\mathbf{X}) \sim \mathcal{N}(\text{vec}(\mathbf{M}), \mathbf{V} \otimes \mathbf{U}) \tag{12}$$

and for the matrix normal distribution

$$\begin{aligned} \mathbb{E}(\mathbf{X}) &= \mathbf{M} && (13)\\ \mathbb{E}\left((\mathbf{X}-\mathbf{M})(\mathbf{X}-\mathbf{M})^T\right) &= \text{tr}(\mathbf{V}) \cdot \mathbf{U} && (14)\\ \mathbb{E}\left((\mathbf{X}-\mathbf{M})^T(\mathbf{X}-\mathbf{M})\right) &= \text{tr}(\mathbf{U}) \cdot \mathbf{V} && (15) \end{aligned}$$

is true and where $\text{tr}(\mathbf{X})$ denotes the trace of a quadratic matrix $\mathbf{X}$. The expression $\mathbf{V} \otimes \mathbf{U}$ denotes the Kronecker product of two matrices and $\text{vec}(\mathbf{X})$ denotes the vectorization of a matrix $\mathbf{X}$. For more about matrix distributions I refer to (Dawid, 1981). The properties (14) and (15) and as a consequence the matrices $\mathbf{U} \in \mathcal{R}^{m \times m}$ and $\mathbf{V} \in \mathcal{R}^{n \times n}$ can be interpreted as column-wise and row-wise covariance. The matrices $\mathbf{U} \in \mathcal{R}^{m \times m}$ and $\mathbf{V} \in \mathcal{R}^{n \times n}$ have to be real-valued, symmetric and positive definite. Further the distribution $\mathcal{MN}(\mathbf{M}, \mathbf{U}, \mathbf{V})$ is sampled easily by

$$\mathbf{Y} = \mathbf{B}_u \mathbf{D}_u \mathbf{X} \mathbf{D}_v \mathbf{B}_v^T \tag{16}$$

where each entry of the matrix $\mathbf{X} \in \mathcal{R}^{m \times n}$ is standard normally distributed and $\mathbf{U} = \mathbf{B}_u \mathbf{D}_u \mathbf{D}_u \mathbf{B}_u^T$ and $\mathbf{V} = \mathbf{B}_v \mathbf{D}_v \mathbf{D}_v \mathbf{B}_v^T$ are eigendecompositions.

Thus the first step of the algorithm is

$$\begin{aligned} \mathbf{Z}_k &\sim \mathcal{MN}(\mathbf{0}, \mathbf{I}_m, \mathbf{I}_n) && (17)\\ \mathbf{Y}_k &= \mathbf{B}_u \mathbf{D}_u \mathbf{Z}_k \mathbf{D}_v \mathbf{B}_v^T \sim \\ &\sim \mathcal{MN}(\mathbf{0}, \mathbf{U}, \mathbf{V}) && (18)\\ \mathbf{X}_k &= \mathbf{M} + \sigma \mathbf{Y}_k \sim \\ &\sim \mathcal{MN}(\mathbf{M}, \sigma^2 \mathbf{U}, \mathbf{V}). && (19) \end{aligned}$$

There is to note that

$$\mathcal{MN}(\mathbf{M}, \sigma^2 \mathbf{U}, \mathbf{V}) = \mathcal{MN}(\mathbf{M}, \mathbf{U}, \sigma^2 \mathbf{V})$$

is true.

Next as in the outline of the original CMA-ES proposed the population points indices are reordered by their corresponding function values such that $f(\mathbf{X}_k) \leq f(\mathbf{X}_{k+1})$. So next the best point $\mathbf{M} \in \mathcal{R}^{n \times n}$ is updated by

$$\mathbf{M} = \mathbf{M} + \sigma \sum_{i=1}^{\mu} \mathbf{w}_k \mathbf{Y}_k = \mathbf{M} + \sigma \langle \mathbf{Y} \rangle_w . \tag{20}$$

Next the global step length $\sigma > 0$ has to be adapted. First the conjugate evolution path is updated

$$\mathbf{p}_\sigma = (1 - c_\sigma)\mathbf{p}_\sigma + \sqrt{c_\sigma(2 - c_\sigma)\mu_{\text{eff}}} \mathbf{B}_u \langle \mathbf{Z} \rangle_w \mathbf{B}_v^T \tag{21}$$

where

$$\begin{aligned} \langle \mathbf{Z} \rangle_w &= \sum_{i=1}^{\mu} \mathbf{w}_k \mathbf{Z}_k && (22)\\ &= \mathbf{B}_u \mathbf{D}_u^{-1} \mathbf{B}_u^T \langle \mathbf{Y} \rangle_w \mathbf{B}_v \mathbf{D}_v^{-1} \mathbf{B}_v^T && (23) \end{aligned}$$

By using the same arguments as in the original CMA-ES the conjugate evolution path $\mathbf{p}_\sigma$ is standard normally distributed in each row and each column under purely random selection and as a consequence the vectorized evolution path $\text{vec}(\mathbf{p}_\sigma)$ is also standard normally distributed under purely random selection. So the global step length $\sigma > 0$ is adapted in the same manner as in the previous section

$$\sigma = \sigma \cdot \exp\left(\frac{1}{d}\left(\frac{\|\text{vec}(\mathbf{p}_\sigma)\|}{\mathbb{E}(\mathcal{N}(\mathbf{0}, \mathbf{I}))} - 1\right)\right). \tag{24}$$

The last step in each iteration is again the covariance adaptation. Here both the column-wise and row-wise covariance matrices have to be adapted. The equations are

$$\begin{aligned} \mathbf{p}_c &= (1 - c_c)\mathbf{p}_c + \\ &+ \sqrt{c_c(2 - c_c)\mu_{\text{eff}}} \langle \mathbf{Y} \rangle_w && (25)\\ \text{tr}(\mathbf{V}) \cdot \mathbf{U} &= (1 - c_1 - c_\mu)\,\text{tr}(\mathbf{V}) \cdot \mathbf{U} + \\ &+ c_1(\mathbf{p}_c \mathbf{p}_c^T) + c_\mu \sum_{k=1}^{\mu} \mathbf{Y}_k \mathbf{Y}_k^T && (26)\\ \text{tr}(\mathbf{U}) \cdot \mathbf{V} &= (1 - c_1 - c_\mu)\,\text{tr}(\mathbf{U}) \cdot \mathbf{V} + \\ &+ c_1(\mathbf{p}_c^T \mathbf{p}_c) + c_\mu \sum_{k=1}^{\mu} \mathbf{Y}_k^T \mathbf{Y}_k . && (27) \end{aligned}$$

Again the accumulation of the evolution path leads to $p_c \sim \mathcal{MN}(\mathbf{M}, \mathbf{U}, \mathbf{V})$ under purely random selection by the same arguments as in the original CMA-ES. As easily seen the outline of the algorithm is basically the same as the original CMA-ES but in the last covariance adaptation step one has to be careful when updating the left and right covariance matrices. For reasons of simplicity the system of equations (26) and (27) is not solved but the values $\text{tr}(\mathbf{U})$ and $\text{tr}(\mathbf{V})$ are taken from the previous iteration.

As already mentioned in the beginning in this proposed algorithm the covariance matrices $\mathbf{U}$ and $\mathbf{V}$ which characterize the distribution of the mutation operator have fewer degree of freedom than the covariance matrix $\mathbf{C}$ in the original CMA-ES. By using a matrix normal distribution for the mutation distribution we have $\frac{m \cdot (m+1) + n \cdot (n+1)}{2}$ degrees of freedom in the covariance matrices $\mathbf{U}$ and $\mathbf{V}$ as in the original CMA-ES the covariance matrix $\mathbf{C}$ has $\frac{m \cdot n \cdot (m \cdot n + 1)}{2}$ degrees of freedom. So the vectorized problem has a lot more degrees of freedom

for adaptation. In high dimensions it is suggested to use the CMA-ES only with diagonal covariance matrices because computing an eigendecomposition in high dimensions is costly. Then the degree of freedom is only $m \cdot n$ any more which is comparable to $\frac{m \cdot (m+1) + n \cdot (n+1)}{2}$. However using a diagonal covariance matrix for characterizing the distribution of the mutation means just considering the variance of each coefficient. Thus no correlation between different coefficients is considered any more and as a consequence it is also not reflected in the mutation distribution. So the benefit of the proposed method should be that even in high dimensions correlation between different variables can be identified.

Next the proposed method is tested on some test functions.

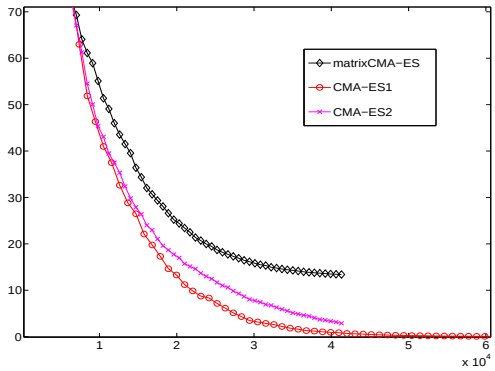

Figure 1: Performance of *matrix-CMA-ES* compared with *CMA-ES1* and *CMA-ES2* on the function f_1. Shown is on the *y*-axis the current best function value and on the *x*-axis the number of function evaluations at this time.

4 NUMERICAL RESULTS

In this section the original CMA-ES method is compared with the here proposed modification. The CMA-ES method is executed with the change rates proposed in (N. Hansen, 2011). Here no eigendecomposition of the covariance matrix in the CMA-ES are computed because the dimension of the considered problems is to high. So only diagonal covariance matrices are considered in the original CMA-ES. The changing rates for the here proposed methods are adapted from the original method. Numerical tests suggested that the here proposed method benefits from larger change rates and fewer sampling points in each iteration.

I compare here two versions of the original the CMA-ES algorithm with the here proposed modification. The algorithm denoted by *CMA-ES1* uses the initially suggested population size $\lambda > 0$ in (Hansen and Ostermeier, 2001) or (N. Hansen, 2011) and the algorithm denoted by *CMA-ES2* uses smaller population size $\lambda > 0$ in each iteration same as the method proposed in this paper. The method which is introduced in this paper is denoted by *matrix-CMA-ES* in the figures below.

The first test problem which is considered here is just the spectral norm for matrices,

$$f_1(\mathbf{A}) = \|\mathbf{A} - \mathbf{B}\|_2 \tag{28}$$

where $\mathbf{B} \in \mathcal{R}^{15\times 20}$ is a randomly chosen matrix. Obviously the solution for minimizing this function is $\mathbf{B}$ and $f(\mathbf{B}) = 0$. The dimension of the problem by vectorizing the matrix $\mathbf{A}$ for the original CMA-ES algorithm then is 300. In figure 1 one can easily see that the here proposed method does not do well solving this problem. As the algorithm continues the progress gets smaller and smaller. In contrary the original CMA-ES method works quite good and is able to solve the problem in a reasonable time. The reason for the bad behaviour is that there is no linear correlation between different variables to capture and so the original CMA-ES algorithm where only the variances of each entry are adapted does a lot better. The behaviour for other kind of norm operators is very similar.

The second problem that is considered is

$$f_2(\mathbf{A}) = \|\mathbf{C} \cdot (\mathbf{A} - \mathbf{B}) \cdot \mathbf{D}\|_\infty \tag{29}$$

where $\mathbf{B} \in \mathcal{R}^{15\times 20}, \mathbf{C} \in \mathcal{R}^{6\times 15}$ and $\mathbf{D} \in \mathcal{R}^{20\times 5}$ are randomly chosen matrices. The optimal solution value is again $f = 0$ but the problem has no unique solution. The dimension of the problem by vectorizing the matrix $\mathbf{A}$ for the original CMA-ES algorithm then is again 300. This problem obviously has some linear correlation added by the matrices $\mathbf{C}$ and $\mathbf{D}$. In figure 2 one can see that the here proposed method performs a lot better than the original CMA-ES method. The reasoning for the better performance is that the modified procedure is capable of covering the linear structure of the problem as the original CMA-ES struggles to do that. The original CMA-ES method even gets random after some time where no real progress with respect to the function value can be seen.

The third function numerical tests were performed on is

$$f_3(\mathbf{A}) = \|(\mathbf{A} - \mathbf{B})\|_2 + \|(\mathbf{A} - \mathbf{C})\|_2 \tag{30}$$

where $\mathbf{B} \in \mathcal{R}^{15\times 20}$ and $\mathbf{C} \in \mathcal{R}^{15\times 20}$ are randomly chosen matrices. The dimension of the problem by vectorizing the matrix $\mathbf{A}$ for the original CMA-ES algorithm then is again 300. Although the third prob-

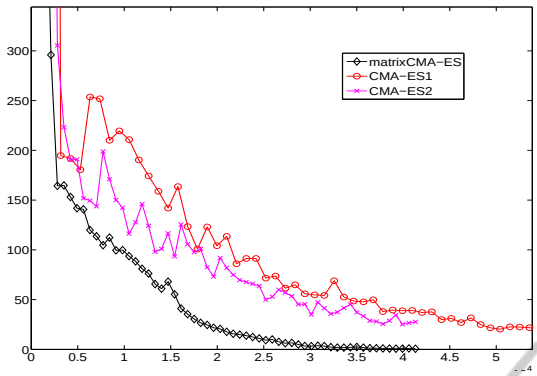

Figure 2: Performance of *matrix-CMA-ES* compared with *CMA-ES1* and *CMA-ES2* on the function f_2. Shown is on the *y*-axis the current best function value and on the *x*-axis the number of function evaluations at this time

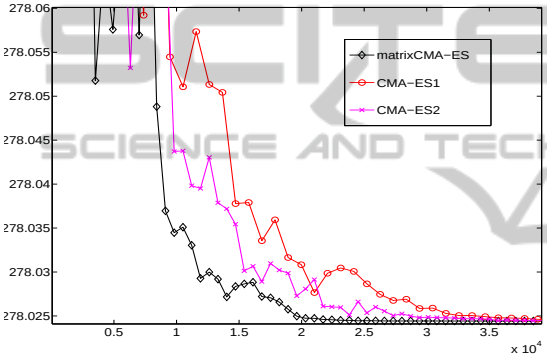

Figure 3: Performance of *matrix-CMA-ES* compared with *CMA-ES1* and *CMA-ES2* on the function f_3. Shown is on the *y*-axis the current best function value and on the *x*-axis the number of function evaluations at this time.

lem looks similar than the first problem the performance of the modified algorithm is much better. The proposed algorithm does not outperform the original CMA-ES but it is a little better than the original versions. The reason for the better performance is that here correlation between the coefficients has to be captured.

The last function I consider is

$$f_4(\mathbf{A}) = \log\left(|\det(\mathbf{C}\cdot(\mathbf{A}-\mathbf{B})\cdot\mathbf{D})|+1\right) \qquad (31)$$

where $\mathbf{B}\in\mathcal{R}^{10\times 12}$, $\mathbf{C}\in\mathcal{R}^{6\times 12}$ and $\mathbf{D}\in\mathcal{R}^{10\times 6}$ are randomly chosen matrices. This problem has no unique solution but the minimal function value is $f=0$. In figure 4 one can see that the here proposed method drastically outperforms the conventional methods for this problem. The newly proposed method adapts better to inner structure of the problem whereas the conventional algorithms can not adapt to this structure.

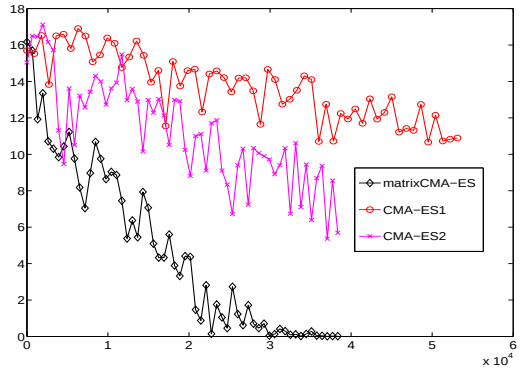

Figure 4: Performance of *matrix-CMA-ES* compared with *CMA-ES1* and *CMA-ES2* on the function f_4. Shown is on the *y*-axis the current best function value and on the *x*-axis the number of function evaluations at this time.

5 CONCLUSION

As a conclusion this paper has introduced a modification to the many existing CMA-ES algorithms. The goal was to be able to apply a CMA-ES algorithm to a function where the input is not a vector but a matrix without vectorizing this matrix. The method proposed here gives the possibility to do this and the paper also investigates the advantages and disadvantages of vectorizing the matrix input when a CMA-ES algorithm is applied. The problem with vectorizing matrices for the purpose of optimization with CMA-ES is that vectorizing even fairly small matrices leads quite fast to a high dimensional problem. In higher dimensions it is very costly and therefore unreasonable to perform eigendecompositions all the time during the algorithm. Therefore linear correlations between the variables can not be captured any more by the conventional CMA-ES algorithm. The here proposed method considers covariance matrices of smaller size and it is therefore no problem to do this eigendecompositions in each iteration. This results into a better way of capturing the inner structure of the considered problem. The disadvantage is the loss of some degrees of freedom in the characterization of the distribution of the mutation operator. If the characterization of the mutation distribution has too few degrees of freedom like in the first example in the previous section the behaviour of the optimization procedure gets random.

In section 4 where some numerical tests have been performed one can see that the more complicated the function to minimize is the better the here proposed method does work as for the easiest kind of problem where the inner structure is very smooth the conventional method performs a lot better. The modified al-

gorithm also seems to allow larger change rates and is therefore more flexible in capturing the distribution of the mutation operator.

The method seems applicable for problems which are not too smooth and where strong linear correlations have to be captured. It is also imaginable for a high dimensional problems where the covariance matrix in the conventional CMA-ES is too large to perform eigendecompositions to rewrite the vector in the form of an suitably sized matrix such that linear correlations between the variables can be captured by the applied method.

In further research methods to overcome the difficulties when the problems structure is very smooth have to be found. Here I think it is possible to look at some other modifications of the CMA-ES algorithm. Moreover the change rates are crucial to the success of a CMA-ES algorithm. So finding solid and good change rates for the algorithm will further improve the proposed method. In addition more numerical tests have to be performed.

REFERENCES

Dawid, A. P. (1981). Some matrix-variate distribution theory: notational considerations and a Bayesian application. *Biometrika*, 68(1):265–274.

Hansen, N. (1998). Verallgemeinerte individuelle Schrittweitenregelung in der Evolutionsstrategie. *Mensch & Buch Verlag, Berlin*.

Hansen, N. and Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 312–317. IEEE.

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

Igel, C., Suttorp, T., and Hansen, N. (2006). A computational efficient covariance matrix update and a (1+ 1)-CMA for evolution strategies. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 453–460. ACM.

Igeland, C., Hansen, N., and Roth, S. (2007). Covariance matrix adaptation for multi-objective optimization. *Evolutionary computation*, 15(1):1–28.

Jastrebski, G. and Arnold, D. (2006). Improving evolution strategies through active covariance matrix adaptation. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 2814–2821. IEEE.

N. Hansen (2011). The CMA Evolution Strategy: A Tutorial.

Schwefel, H. (1977). *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*. Birkhäuser Verlag, Basel.