Functionally Specialized CMA-ES: A Modification of CMA-ES based on the Specialization of the Functions of Covariance Matrix Adaptation and Step Size Adaptation

Youhei Akimoto
akimoto@fe.dis.titech.ac.jp

Jun Sakuma
jun@fe.dis.titech.ac.jp

Isao Ono
isao@dis.titech.ac.jp

Shigenobu Kobayashi
kobayasi@dis.titech.ac.jp

Department of Computational Intelligence and Systems Science
Tokyo Institute of Technology
4259 Nagatsuta-cho, Midori-ku, Yokohama-shi, Kanagawa, Japan

ABSTRACT

This paper aims the design of efficient and effective optimization algorithms for function optimization. This paper presents a new framework of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Recent studies modified the CMA-ES from the viewpoint of covariance matrix adaptation and resulted in drastic reduction of the number of generations. In addition to their modification, this paper modifies the CMA-ES from the viewpoint of step size adaptation. The main idea of modification is semantically specializing functions of covariance matrix adaptation and step size adaptation. This new method is evaluated on 8 classical unimodal and multimodal test functions and the performance is compared with standard CMA-ES. The experimental result demonstrates an improvement of the search performances in particular with large populations. This result is mainly because the proposed Hybrid-SSA instead of the existing CSA can adjust the global step length more appropriately under large populations and function specialization helps appropriate adaptation of the overall variance of the mutation distribution.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*Global optimization, Unconstrained optimization*; G.3 [**Probability and Statistics**]: Probabilistic algorithms

General Terms

Algorithms, Performance, Experimentation

Keywords

Evolution Strategy, Functional Specialization, Step Size Adaptation, Covariance Matrix Adaptation

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.

1. INTRODUCTION

The derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES) [8] is a successful evolutionary algorithm for function optimization problems. The CMA-ES adapts an arbitrary multivariate normal distribution to exhibit several invariances, which are highly desirable for uniform behavior on classes of functions [3].

Originally designed for small population sizes, the CMA-ES was interpreted as a robust local search strategy [7]. The CMA-ES efficiently minimizes unimodal test functions and in particular it is superior on ill-conditioned and non-separable problems to other evolutionary and estimation of distribution algorithms. It was successfully applied to an considerable number of real world problems. In [11, 6] the CMA-ES was expanded by the so-called rank-μ-update. The rank-μ-update exploits the information contained in large populations more effectively without affecting the performance for small population sizes.

Recent studies [10, 5] showed a good performance of the CMA-ES combining large populations and rank-μ-update on the unimodal and multimodal functions without parameter turning except for the population size. In [2, 1] the CMA-ES restart strategies are proposed for the search on multimodal functions. As noted above, the CMA-ES is taken notice as a global optimization algorithm.

Achievement of efficient optimization performance, i.e. reducing the number of generations, is important if a large population size is desired: to improve global search properties on multimodal functions and to implement the algorithm on parallel machines. This is the main objective of this paper.

The remainder is organized as follows. In Sect. 2 we describe the standard CMA-ES and indicate the characteristics. The main point is in Sect. 3. We study the behavior of the CMA-ES and then propose a new framework of CMA-ES which is designed for reduction of the number of generations is referred to as FS-CMA-ES. Section 4 compares FS-CMA-ES with CMA-ES on unimodal and multimodal test functions. Section 5 gives some discussion about FS-CMA-ES. Finally in Sect. 6, this paper is concluded.

2. EXISTING STRATEGY

This section provides a description of the CMA-ES combining weighted recombination and rank-μ-update of the covariance matrix, and describes the characteristics.

Table 1: Default strategy parameters settings

Sampling and Recombination	$\lambda = 4 + \lfloor 3\ln(n) \rfloor$, $\mu = \lfloor \lambda/2 \rfloor$, $\mathrm{w}_i = \frac{\ln(\mu+1)-\ln(i)}{\mu\ln(\mu+1)-\sum_{j=1}^{\mu}\ln(j)}$,
Step Size Adaptation	$c_\sigma = \frac{\mu_{\mathrm{eff}}+2}{n+\mu_{\mathrm{eff}}+3}$, $d_\sigma = 1 + c_\sigma + 2\max\left(0, \sqrt{\frac{\mu_{\mathrm{eff}}-1}{n+1}} - 1\right)$,
Covariance Matrix Adaptation	$c_{\mathrm{c}} = \frac{4}{n+4}$, $\mu_{\mathrm{cov}} = \mu_{\mathrm{eff}}$, $c_{\mathrm{cov}} = \frac{1}{\mu_{\mathrm{cov}}}\frac{2}{(n+\sqrt{2})^2} + \left(1 - \frac{1}{\mu_{\mathrm{cov}}}\right)\min\left(1, \frac{2\mu_{\mathrm{eff}}-1}{(n+2)^2+\mu_{\mathrm{eff}}}\right)$.

2.1 Algorithm

The algorithm outlined here is identical to that described by [5] except for the device that stalls update of $\boldsymbol{p}_{\mathrm{c}}$ if $\boldsymbol{p}_\sigma$ is large. This prevents a too fast increase of axes of $\boldsymbol{C}$ in a linear surrounding, i.e. when the step size is far too small [4]. For easiness we do not use it in this paper.

[Step 0: Parameter Initialization]
The initialization of the mean point $\boldsymbol{m}^{(0)}$, the global step size $\sigma^{(0)}$ and the covariance matrix $\boldsymbol{C}^{(0)}$ are problem dependent. Assign $\mathbf{0}$ as initial values to evolution paths $\boldsymbol{p}_c$ and $\boldsymbol{p}_\sigma$. Set strategy parameters to their default values according to Table 1. Repeat following iteration steps from [Step 1] to [Step 5] until termination criteria are satisfied.

[Step 1: Eigen Decomposition]
Compute an eigen decomposition of the covariance matrix of mutation distribution, $\boldsymbol{C}^{(g)} = \boldsymbol{B}\boldsymbol{D}(\boldsymbol{B}\boldsymbol{D})^{\mathrm{t}}$, where the superscript t denotes matrix transpose operator. The columns of $n \times n$ orthogonal matrix $\boldsymbol{B}$ are the normalized eigenvectors of $\boldsymbol{C}$, and the diagonal elements of $n \times n$ diagonal matrix $\boldsymbol{D}$ are the square roots of the eigenvalues of $\boldsymbol{C}$.

[Step 2: Sampling and Evaluation]
Generate offspring for $i = 1, \cdots, \lambda$ according to

$$\boldsymbol{x}_i^{(g+1)} = \boldsymbol{m}^{(g)} + \sigma^{(g)}\boldsymbol{B}\boldsymbol{D}\boldsymbol{B}^{\mathrm{t}}\boldsymbol{z}_i^{(g+1)} \quad , \tag{1}$$

where the random vectors $\boldsymbol{z}_i^{(g+1)}$ are independent and n-dimensional normally distributed with expectation zero and the identity covariance matrix, $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$ and we refer to those as normalized points. Then, evaluate the fitness $f(\boldsymbol{x}_i^{(g+1)})$ of the sampled point $\boldsymbol{x}_i^{(g+1)}$ for all i. When $\boldsymbol{x}_i^{(g+1)}$ is infeasible, re-sampling $\boldsymbol{x}_i^{(g+1)}$ until it becomes feasible is a simple way to handle any type of boundaries and constraints. Other methods are usually better such as penalty function method or repair operator if these are available.

[Step 3: Recombination]
Compute the weighted mean of the best μ points according to

$$\boldsymbol{m}^{(g+1)} = \sum_{i=1}^{\mu} \mathrm{w}_i \boldsymbol{x}_{i:\lambda}^{(g+1)} \quad , \tag{2}$$

where $\boldsymbol{x}_{i:\lambda}^{(g+1)}$ denotes the i-th best fitness offspring point. In following steps and Table 1, $\mu_{\mathrm{eff}} = (\sum_{i=1}^{\mu} \mathrm{w}_i^2)^{-1}$ denotes the variance effective selection mass.

[Step 4: Step Size Adaptation (SSA)]
Update the evolution path $\boldsymbol{p}_\sigma$ according to

$$\boldsymbol{p}_\sigma^{(g+1)} = (1 - c_\sigma)\boldsymbol{p}_\sigma^{(g)} + \sqrt{c_\sigma(2 - c_\sigma)}\sqrt{\mu_{\mathrm{eff}}} \sum_{i=1}^{\mu} \mathrm{w}_i \boldsymbol{z}_{i:\lambda}^{(g+1)} \tag{3}$$

and then, update the global step size according to

$$\sigma^{(g+1)} = \sigma^{(g)} \cdot \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\boldsymbol{p}_\sigma^{(g+1)}\|}{\mathrm{E}(\|\mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)\|)} - 1\right)\right) \quad , \tag{4}$$

where $\mathrm{E}(\cdot)$ denotes the average operator, so $\mathrm{E}(\|\mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)\|)$ means the average length of n-dimensional standard normally distributed random vector. We use the approximate value $\sqrt{n}(1 - 1/(4n) + 1/(21n^2))$ instead of the exact value of $\mathrm{E}(\|\mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)\|)$.

[Step 5: Covariance Matrix Adaptation (CMA)]
Update the evolution path $\boldsymbol{p}_{\mathrm{c}}$ according to

$$\boldsymbol{p}_{\mathrm{c}}^{(g+1)} = (1 - c_{\mathrm{c}})\boldsymbol{p}_{\mathrm{c}}^{(g)} + \sqrt{c_{\mathrm{c}}(2 - c_{\mathrm{c}})}\sqrt{\mu_{\mathrm{eff}}}\boldsymbol{B}\boldsymbol{D}\boldsymbol{B}^{\mathrm{t}} \sum_{i=1}^{\mu} \mathrm{w}_i \boldsymbol{z}_{i:\lambda}^{(g+1)} \tag{5}$$

and then, update the covariance matrix according to

$$\boldsymbol{C}^{(g+1)} = (1 - c_{\mathrm{cov}})\boldsymbol{C}^{(g)} + c_{\mathrm{cov}}\left(\frac{1}{\mu_{\mathrm{cov}}}\boldsymbol{p}_{\mathrm{c}}^{(g+1)}\{\boldsymbol{p}_{\mathrm{c}}^{(g+1)}\}^{\mathrm{t}} + \left(1 - \frac{1}{\mu_{\mathrm{cov}}}\right)\boldsymbol{C}_\mu\right) \quad , \tag{6}$$

where

$$\boldsymbol{C}_\mu = \boldsymbol{B}\boldsymbol{D}\boldsymbol{B}^{\mathrm{t}}\left(\sum_{i=1}^{\mu} \mathrm{w}_i \boldsymbol{z}_{i:\lambda}^{(g+1)}\{\boldsymbol{z}_{i:\lambda}^{(g+1)}\}^{\mathrm{t}}\right)\boldsymbol{B}\boldsymbol{D}\boldsymbol{B}^{\mathrm{t}} \quad . \tag{7}$$

In step 3, weighted recombination is treated. This is identical to intermediate recombination if $\mathrm{w}_i = 1/\mu$, and then $\lambda = 4\mu$ is desired. The step size adaptation described in step 4 is the cumulative step size adaptation (CSA, see [12]). The covariance matrix adaptation in step 5 is referred to as the hybrid covariance matrix adaptation (Hybrid-CMA, see [6]). The update rule of covariance matrix using $\boldsymbol{p}_{\mathrm{c}}$ is so-called rank-one-update and that using $\boldsymbol{C}_\mu$ is so-called rank-μ-update.

2.2 Characteristics

2.2.1 *Utilization of Step Size Adaptation*

The adaptation of mutation parameters consists of two parts: adaptation of the covariance matrix $\boldsymbol{C}^{(g)}$ and adaptation of the global step size $\sigma^{(g)}$. Reference [4] finds two reasons to introduce a step size adaptation mechanism in addition to a covariance matrix adaptation mechanism.

Reason 1: Difference of Possible Leaning Rates. The largest reliable learning rate for the covariance matrix update is too slow to achieve competitive change rates for the overall step length. SSA allows the competitive change rates because of the larger possible learning rate for σ update.

Reason 2: Well Estimation of Optimal Overall Step Length. The optimal overall step length cannot be well approximated by existing rules of covariance matrix update e.g. (6), in particular if μ_{eff} is chosen larger than one. SSA supplements the optimal overall step length adaptation.

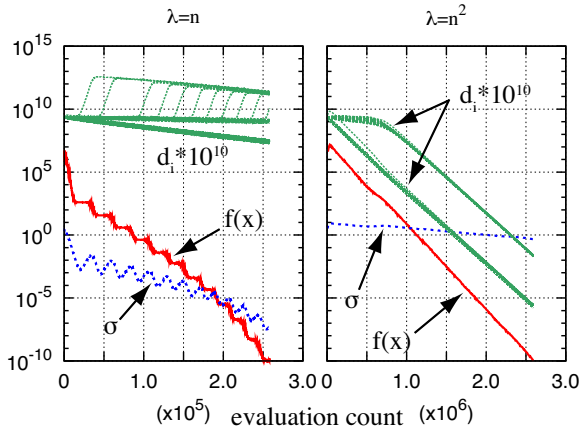

Figure 1: One simulation result for CMA-ES on 80 dimensional k-tablet function. Best function value ($f(x)$), global step size (σ) and eigenvalues of $\boldsymbol{C}$ (d_i), versus function evaluation count are shown for $\lambda = n = 80$ (left) and $\lambda = n^2 = 6400$ (right).

2.2.2 *Relation between Population Sizes and the Behavior of the Existing CMA-ES*

Take a look at Fig. 1. In the case of $\lambda = n$, CMA gradually adjusts the scale of the function while SSA adapts (decreases) the overall step size. On the other hand CMA adjusts the scale and the overall step size, while SSA seems hardly to makes profit in the case of $\lambda = n^2$.

Factor 1: High Change Rate of Overall Variance by CMA. The change rate of the overall variance attributed to the covariance adaptation mechanism can not be ignored by the change rate attributed to the step size adaptation. While large population sizes and large adaptation rates c_{cov} help the CMA mechanism with fast adaptation of $\boldsymbol{C}$, the covariance matrix adaptation remarkably contributes to the change of the overall variance.

Factor 2: The Behavior of CSA under Large Populations. The cumulative step size adaptation ceases to work properly when the population size becomes too large. A larger population size appears to have a destabilizing effect that the momentum term of $\boldsymbol{p}_\sigma$ tends to become unstable for smaller values of d_σ (damping parameter). This would suggest choosing d_σ to be even larger, resulting in an even slower change rate. This is described in [6] and after the report large value is assigned to d_σ in [10, 5] when μ_{eff} is enough large.

3. PROPOSAL STRATEGY

3.1 Motivation

The adaptation mechanisms of covariance matrix $\sigma^2\boldsymbol{C}$ in CMA-ES can be said that the functions are formally allotted to two parts (CMA and SSA). But it is not realized that the function is semantically allotted to two parts in the existing CMA-ES when the population size is large.

We assume there is room for improvement in search performance for large populations. There is an idea for the improvement that the functions of CMA and SSA should be semantically divided not only formal division. The function specialization may improve the search efficiency because (i) the possible learning rate of σ update (overall variance) could be always higher than that of $\boldsymbol{C}$ update (the other information about covariance matrix) and (ii) update of overall variance only by σ update, i.e. step size adaptation, could make adaptation of step length more adequate.

The resulting framework from the realization of the function specialization is referred to as Functionally Specialized CMA-ES (FS-CMA-ES). We realize the FS-CMA-ES by combining normalization of covariance matrix every its update introduced in Sect. 3.2 and a new step size adaptation proposed in Sect. 3.3 as a alternate of CSA.

3.2 CMA with Normalization

First step of the semantical function specialization is to prevent CMA from adapting the overall variance of $\boldsymbol{C}$. There is a trivial idea that the covariance matrix $\boldsymbol{C}$ is normalized after each $\boldsymbol{C}$ update. Reference [6] says that this may be effective even though it is not elegant.

Various scalar amounts can be thought for the measure of the normalization of matrix. In this paper, two variants of normalization mechanism: determinant normalization and trace normalization are presented.

All we have to do is just normalize $\boldsymbol{C}$ according to
determinant normalization:

$$\boldsymbol{C}^{(g+1)} := (\det(\boldsymbol{C}^{(0)})/\det(\boldsymbol{C}^{(g+1)}))^{1/n}\boldsymbol{C}^{(g+1)} \tag{8}$$

or
trace normalization:

$$\boldsymbol{C}^{(g+1)} := (\mathrm{Tr}(\boldsymbol{C}^{(0)})/\mathrm{Tr}(\boldsymbol{C}^{(g+1)}))\boldsymbol{C}^{(g+1)} \tag{9}$$

after each update of covariance matrix by CMA (e.g. Hybrid-CMA or Active-CMA [9]).

3.3 A Hybrid Step Size Adaptation

3.3.1 *The Behavior of CSA with Large Populations*

Let us discuss the unstableness of the cumulative step size adaptation described at Factor 2 in Sect. 2.2.2. Here we consider that intermediate recombination $\mathrm{w}_i = 1/\mu$ is used and $R = \lambda/\mu$ is a constant number. Let $\boldsymbol{y}_i = \boldsymbol{z}_{i:\lambda}$ for $1 \leq i \leq \mu$. Selected normalized points $\{\boldsymbol{y}_i\}$ can be assumed to be independent and identical random vectors with $\mathrm{E}(\boldsymbol{y})$ as its average and $\mathrm{Cov}(\boldsymbol{y})$ as its covariance matrix because intermediate recombination is treated and does not use order information of selected points. Since $\mu_{\mathrm{eff}} = \mu$, an approximation for the second summation term of (3)

$$\sqrt{\mu_{\mathrm{eff}}}\sum_{i=1}^{\mu}\mathrm{w}_i\boldsymbol{y}_i \approx \sqrt{\mu_{\mathrm{eff}}}\cdot\mathrm{E}(\boldsymbol{y}) + \mathcal{N}(\boldsymbol{0}, \mathrm{Cov}(\boldsymbol{y})) \tag{10}$$

is derived from central limit theorem under large μ. At the convergence stage $\sqrt{\mu_{\mathrm{eff}}}\cdot\mathrm{E}(\boldsymbol{y})$ is sufficiently smaller than $\mathcal{N}(\boldsymbol{0}, \mathrm{Cov}(\boldsymbol{y}))$. But at the stage where the mean of mutation distribution moves, e.g. on linear function, $\sqrt{\mu_{\mathrm{eff}}}\sum_{i=1}^{\mu}\mathrm{w}_i\boldsymbol{y}_i$ approaches to $\sqrt{\mu_{\mathrm{eff}}}\cdot\mathrm{E}(\boldsymbol{y})$ with spread of $\mathcal{N}(\boldsymbol{0}, \mathrm{Cov}(\boldsymbol{y}))$. That is, $\|\boldsymbol{p}_\sigma\|$ is enlarged in the order of $\sqrt{\mu_{\mathrm{eff}}}$. To prevent a rapid expansion of σ caused by this, we need to choose d_σ large (proportional to $\sqrt{\mu_{\mathrm{eff}}}$). The suggested choice of d_σ not only makes σ update stable at the stage of mean movement but also disturbs σ convergence.

Table 2: Test functions to be minimized and initialization regions

Function Name	Local Search Performance	Init. Region
Sphere	$f_{\mathrm{Sphere}} = \sum_{i=1}^{n} x_i^2$	$[1,5]^n$
k-tablet($k = n/4$)	$f_{k-\mathrm{tablet}} = \sum_{i=1}^{k} x_i^2 + \sum_{i=k+1}^{n} (100x_i)^2$	$[1,5]^n$
Ellipsoid	$f_{\mathrm{Ellipsoid}} = \sum_{i=1}^{n} (1000^{\frac{i-1}{n-1}} x_i)^2$	$[1,5]^n$
Rosenbrock	$f_{\mathrm{Rosenbrock}} = \sum_{i=1}^{n-1} \left(100\left(x_i^2 - x_{i+1}\right)^2 + (x_i - 1)^2\right)$	$[-2,2]^n$
Function Name	**Global Search Performance**	**Init. Region**
Ackley	$f_{\mathrm{Ackley}} = 20 - 20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) + e - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)$	$[1,30]^n$
Bohachevsky	$f_{\mathrm{Bohachevsky}} = \sum_{i=1}^{n-1} \left(x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7\right)$	$[1,15]^n$
Schaffer	$f_{\mathrm{Schaffer}} = \sum_{i=1}^{n-1} \left(x_i^2 + x_{i+1}^2\right)^{0.25} \left(\sin^2\left(50\left(x_i^2 + x_{i+1}^2\right)^{0.1}\right) + 1.0\right)$	$[10,100]^n$
Rastrigin	$f_{\mathrm{Rastrigin}} = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right)$	$[1,5]^n$

3.3.2 Proposal of Hybrid Step Size Adaptation

Section 3.3.1 indicates that it is difficult for CSA with large populations to efficiently and effectively adapt the global step size even if the damping parameter is carefully chosen. This section proposes a new step size adaptation to adapt adequate step size not only with small populations but also with large populations.

We introduce a new parameter like maximum likelihood estimator of variance

$$\nu_\sigma^{(g+1)} = \sum_{i=1}^{\mu} \mathrm{w}_i \|\boldsymbol{z}_{i:\lambda}^{(g+1)}\|^2 \ . \tag{11}$$

in order to achieve the ability to adapt an adequate step size under large populations. This is identical to the sample mean of squared norm of selected μ normalized points $\boldsymbol{z}$ if intermediate recombination is used. The parameter divided by n, ν_σ/n, means the sample variance of selected normalized points when its covariance matrix can be written as $\sigma^2\boldsymbol{I}$.

It can be easy to expect that a global step size adaptation using this ν_σ gets stable as the population size becomes large. But my preliminary experimental result shows that such step size adaptation adapts smaller global step size than a step size adaptation using an evolution path like the cumulative step size adaptation, and then it has lower progress rate of fitness, and also it is easier to be caught by local minima. This is simply because the latter is based on the idea that the global step size may as well enlarge when the mean vector of mutation distribution repeatedly moves to almost same direction, but the former is not. Hence, the new step size adaptation also utilizes an evolution path which is identical to one used by the cumulative step size adaptation and obeys (3).

We propose a new step size adaptation to achieve the ability to adapt an adequate step size whether the population size is small or large, and refer to the resulting algorithm as the hybrid step size adaptation (Hybrid-SSA). The global step size σ obeys

$$\begin{aligned}\sigma^{(g+1)} = \sigma^{(g)} \cdot [&(1 - c_{\mathrm{ssa}}) \\ &+ c_{\mathrm{ssa}}\{(1-\alpha_\sigma)\nu_\sigma^{(g+1)} + \alpha_\sigma\|\boldsymbol{p}_\sigma^{(g+1)}\|^2\}/n]^{1/2} \ ,\end{aligned} \tag{12}$$

where α_σ is a parameter for hybrid rate and c_{ssa} is a parameter of learning rate for the step size adaptation. Next section discuss these parameters and the learning rate of evolution path c_σ.

3.3.3 Discussion about the Parameters

It remains to determine appropriate values for the parameters of the hybrid step size adaptation. It is not an exaggeration to say that whether the hybrid step size adaptation works well or not depends on the parameters' setting.

The destabilization discussed in Sect 3.3.1 also appears in this case if the the parameters are inadequate. The value of the hybrid rate α_σ should be inversely proportional to μ_{eff} to prevent the destabilization and make best use of ν_σ for efficient adaptation under large population size. If α_σ is not so, c_{ssa} need to be instead. But this incurs too slow convergence as seen in CSA.

We focus on σ^2 increasing rate on like linear function where the mean of mutation distribution moves to almost same direction repeatedly and the global step size gets larger. The average squared length of the summation term in (3) can be assumed to be independent on the number of dimension of search space, and then the average squared length of evolution path $\|\boldsymbol{p}_\sigma\|^2$ becomes proportional to $\mu_{\mathrm{eff}}(2 - c_\sigma)/c_\sigma$. And the variance can be assumed to be almost proportional to n. These are because the selection add pressure on the gradient direction of mutation distribution to move and add no pressure on the perpendicular directions to it. Since the Hybrid-SSA utilizes an evolution path to achieve the ability to make σ large in cases like this, the σ^2 increasing effect by the average of evolution path should be independent on n. For this, the coefficient of $\|\boldsymbol{p}_\sigma\|$ in (12), $\alpha_\sigma\mu_{\mathrm{eff}}(2-c_\sigma)/(c_\sigma n)$, should be independent on n, and also μ, of course. Therefore we choose c_σ so.

Finally we consider the value of c_{ssa}. Needless to say, it is desired that c_{ssa} becomes high when the population size becomes large. But the higher learning rate causes the unstableness of step size. Therefore it should be chosen carefully. We determine the value of c_{ssa} to prevent the destabilization of evolution path even though the hybrid rate $\alpha_\sigma = 1$.

From above-mentioned discussion and careful thought of dependency relation between these parameters, we find the values of new parameters for Hybrid-SSA according to

$$c_\sigma = \frac{2\rho}{1+\rho}, \quad \alpha_\sigma = \frac{n}{\mu_{\mathrm{eff}}} \cdot \rho \ , \tag{13}$$

$$c_{\mathrm{ssa}} = \left(\frac{n}{\mu}\frac{c_\sigma}{2-c_\sigma}\alpha_\sigma + (1-\alpha_\sigma)\right) \cdot \rho \ , \tag{14}$$

where $\rho = 1 - \exp(-\mu/n)$ s.t. $\rho \leq 1$ and $\rho \leq \mu_{\mathrm{eff}}/n$. There may be room for improvement.

Table 3: Averaged generation numbers to reach f_{stop} over 50 trials with CMA-ES (CMA) and FS-CMA-ES (FS) for population size $\lambda = 4+\lfloor 3\cdot\ln(n)\rfloor$ (def.), n, n^2 on Sphere, Ellipsoid, k-tablet and Rosenbrock functions for dimension $n = 10, 20, 40, 80$. Notice that the number of function evaluation equals to the number of generation multiplied by λ.

Func. Name		Sphere			Ellipsoid			k-tablet			Rosenbrock		
λ		def.	n	n^2	def.	n	n^2	def.	n	n^2	def.	n	n^2
$n = 10$	CMA	180.4	180.4	94.5	339.8	339.8	114.8	481.7	481.7	135.3	686.5	686.5	216.4
	FS	134.0	134.0	55.0	302.5	302.5	75.3	405.6	405.6	97.5	642.2	642.2	172.8
$n = 20$	CMA	276.5	224.4	136.9	738.2	499.8	161.2	1350.1	909.7	184.3	1850.0	1306.4	406.3
	FS	217.9	164.7	73.4	698.6	455.8	95.3	1217.9	792.6	123.1	1826.3	1271.6	309.0
$n = 40$	CMA	412.8	285.9	205.1	1725.2	830.7	238.5	3732.3	1826.2	268.0	5552.5	2918.1	965.7
	FS	333.9	209.7	101.7	1650.1	785.2	128.8	3408.0	1617.9	162.4	5498.3	2872.2	676.8
$n = 80$	CMA	676.2	378.6	315.3	4079.0	1492.1	365.6	8620.6	3299.4	405.6	18899.2	7442.1	2707.0
	FS	558.1	281.8	145.3	3883.8	1402.6	180.7	7636.0	2840.1	223.0	19515.2	7573.8	1700.7

4. EXPERIMENTAL EVALUATION

4.1 Experimental Procedure

In this section, the local search performance and global search performance of FS-CMA-ES using determinant normalization proposed in Sect. 3 are compared with those of CMA-ES explained in Sect. 2.

Unconstrained unimodal and multimodal test functions are summarized in Table 2. All of the functions have a minimal function value of 0, located at $\boldsymbol{x} = \mathbf{0}$, except for the Rosenbrock function, where the global optimum is located at $x_i = 1$ for all i. Besides the Rosenbrock function, the functions are point symmetry around the global optimum. To avoid an easy exploitation of the symmetry, asymmetrical initialization intervals are used.

The performances are compared for $n = 10$, 20, 40, 80. All of the initialization regions are n-dimensional super cubic regions described as $[a, b]^n$. Therefore $\boldsymbol{C}^{(0)}$ is set to a n-dimensional unit matrix $\boldsymbol{I}_n$, the initial step size is set to half of the initialization intervals, $\sigma^{(0)} = (b - a)/2$. The starting point is set to the center point of the initialization regions, $m_i^{(0)} = (a + b)/2$ for all i. All runs are performed with the default strategy parameter settings given in Sect. 2 for CMA-ES and in Sect. 3 for FS-CMA-ES[1], except for the population size λ. 50 runs are conducted for each setting. Each run is stopped and regarded as success, when the function value smaller than $f_{\text{stop}} = 10^{-10}$ is attained. Additionally, the run is stopped after $n \times \lambda \times 10^3$ function evaluations, or when $\mathrm{d}_{\min} \times \sigma^{(g)} < 10^{-15}$, where $\mathrm{d}_{\min}$ is the minimum eigenvalue of covariance matrix (for f_{Schaffer} 10^{-30}).

4.2 Local Search Performance

Local search methods are desired to attain local minima efficiently. We evaluate the local search performance by the averaged number of generations to reach f_{stop} on unimodal test functions. The Rosenbrock function is not a unimodal function, but the number of failure that a CMA-ES (including FS-CMA-ES) attains not the global optimum but a local minimum is less than on the other multimodal functions in Table 2. Hence, we evaluate the local search performance on the Rosenbrock function.

[1]In this experiment, $c_{\text{ssa}} = 1 - \alpha_\sigma(1 - c_\sigma)$ instead of (14) because we found the adequate c_{ssa} in (14) after the experiment. I think that the setting of c_{ssa} in (14) is more appropriate.

We compare the local search performances of FS-CMA-ES with standard CMA-ES for the default population size $\lambda = 4 + \lfloor 3 \cdot \ln(n)\rfloor$, n and n^2. Table 3 shows the averaged generation numbers to reach f_{stop} over 50 trials on each function for each dimension. Serial performance is shown in function evaluations under small λ and parallel one is evaluated the number of generations under large λ.

Also standard CMA-ES and proposed one cost smaller numbers of generations to reach f_{stop} when the population size becomes large. The effect of the reduction of the number of generations appears significantly in particular on Ellipsoid function and k-tablet function, which are ill-conditioned function, and Rosenbrock function, which has a curved ridge structure made of strong variable dependency, because large populations help adaptation of covariance matrix. A greater effect of generations under large populations is on FS-CMA-ES than on CMA-ES. The number of generations (or evaluations) with FS-CMA-ES is zero to about 25 percent smaller than that with standard CMA-ES in the case of $\lambda = n$ and about 20 to 50 percent smaller in the case of $\lambda = n^2$. Higher dimensionality has a greater tendency to show the performance improvement. This is mainly because Hybrid-SSA makes better use of the information in large populations than CSA.

The standard CMA-ES with small populations ($\lambda \leq n$) narrows down mutation distribution by reduction of σ, i.e. by SSA, as well as the proposed one. Therefore the better performance of the proposed one seen in small λ is attributed to the effect of replacement CSA with Hybrid-SSA. On the other hand, standard CMA-ES with large populations does it by the effect of reduction of eigenvalues of $\boldsymbol{C}$ and CSA does not work well (described in Sect. 2.2.2). FS-CMA-ES update the overall variance of mutation distribution only by Hybrid-SSA regardless of population sizes. Therefore the difference of the performance on $\lambda = n^2$ shows the difference between the convergence performance of CMA (for standard CMA-ES) and that of Hybrid-SSA (for FS-CMA-ES). These result are summarized as follows: Hybrid-SSA has an ability to more efficiently search optimum than CSA; updating overall step length only by SSA is more efficient than by CMA and a little bit by CSA.

Above-mentioned difference of performance significantly appears prominently in the result on well-conditioned function e.g. Sphere function. At an early stage of optimization on ill-conditioned functions, e.g. Ellipsoid function and k-tablet function, CMA-ESs including standard and proposal

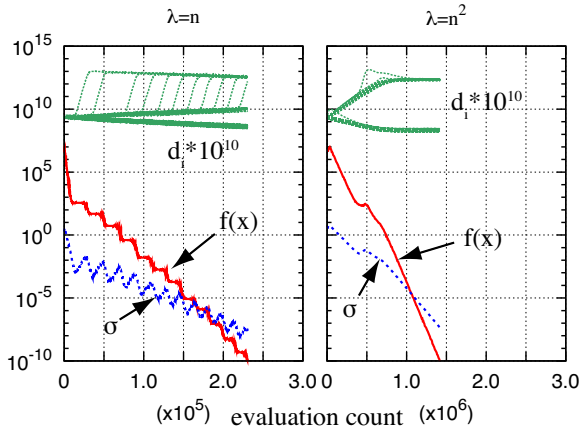

Figure 2: One simulation result for FS-CMA-ES on 80 dimensional k-tablet function. Best function value ($f(x)$), global step size (σ) and eigenvalues of $\boldsymbol{C}$ (d_i), versus function evaluation count are shown for $\lambda = n = 80$ (left) and $\lambda = n^2 = 6400$ (right).

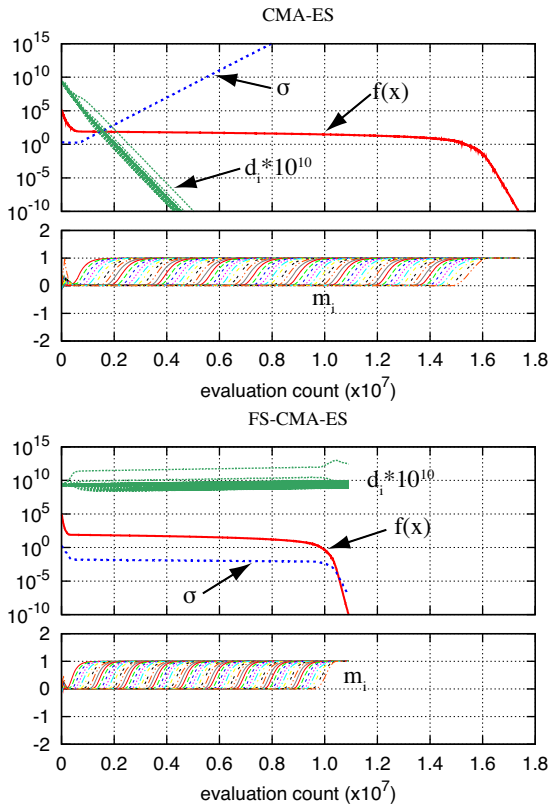

Figure 3: One simulation results on 80 dimensional Rosenbrock function. Best function value ($f(x)$), global step size (σ) , eigenvalues of $\boldsymbol{C}$ (d_i) and mean coordinate values m_i, versus function evaluation count are shown for standard CMA-ES (top) and FS-CMA-ES (bottom) with $\lambda = n^2$.

adjust the covariance matrix to landscape of function. In particular, most generations to reach the optimum are spent on the adjustment when the population size is small (see also Fig. 1 for standard CMA-ES and Fig. 2 for FS-CMA-ES). Therefore performance improvement by FS-CMA-ES on ill-conditioned function is shown less than on well-conditioned Sphere function under small populations while the improvement appears as well as on Sphere function under large populations. Rosenbrock function has a curved ridge structure and so CMA-ESs gradually moves the mutation distribution along the ridge to optimum. Figure 3 shows that the mean vector of mutation distribution is gradually moving from $\mathbf{0}$ to $\mathbf{1}$ and most generations to reach optimum are spent on the movement. At the stage of the movement, SSA enlarges σ and CMA makes $\boldsymbol{C}$ small and so it is needed to keep an appropriate balance between σ and $\boldsymbol{C}$ in standard CMA-ES, while SSA keeps σ and the size of $\boldsymbol{C}$ is kept by normalization in FS-CMA-ES. The comparison of Fig. 3 indicate that it makes available to adapt step size effectively that the adaptation of the overall variance is done only by SSA.

4.3 Global Search Performance

Global search methods are needed to attain the global optimum on many number of local minima. The global search performance is evaluated by average function evaluations for successful runs, divided by success rate. This measurement is introduced in [5]. Because the optimal population size takes a wide range of values [5], population size is increased repeatedly in the sequence a, $\sqrt{2}a$, $2a$, $\cdots$, $8\sqrt{2}a$. Start population size a is selected as $a_{\mathrm{Ackley}} = 2\ln(n)$, $a_{\mathrm{Bohachevsky}} = n$, $a_{\mathrm{Schaffer}} = 2n$, $a_{\mathrm{Rastrigin}} = 10n$.

Figure 6 shows averaged number of evaluations divided by success rate, versus population size. Each line indicates the performance on $n = 10, 20, 40, 80$ with standard CMA-ES and FS-CMA-ES. Note that the scale of x-axis and y-axis are logarithmic.

It is important for global search to figure out the landscape of the function in order to locate a relatively favorable local optimum (global optimum in this case). The larger population sizes helps to locate the global optimum with higher probability in many cases however the difficulty of locating the global optimum differs according to multimodal property of objective function. Figure 4 shows the success rate versus population size for Schaffer function, which represents the typical picture. The dependency between success rate and population size of FS-CMA-ES is similar to that of standard CMA-ES (see [5] for detail), except that FS-CMA-ES require about $\sqrt{2}$ times larger population sizes than CMA-ES. In Fig. 6 the best performance for FS-CMA-ES is located about $\sqrt{2}$ times larger population size than standard CMA-ES, but the best performance for FS-CMA-ES is as well as CMA-ES. This is that the serial performance (function evaluations) of FS-CMA-ES is as well as standard CMA-ES on the other hand the parallel performance (the number of generations) is better (less) than standard one, because the number of function evaluations is equal to the number of generations times population sizes (the number of sampled point for each generation). Also with larger populations than the best population size the performance significantly improves. FS-CMA-ES can locate the global optimum efficiently in the case that larger population size is needed because the best population size is generally unknown.

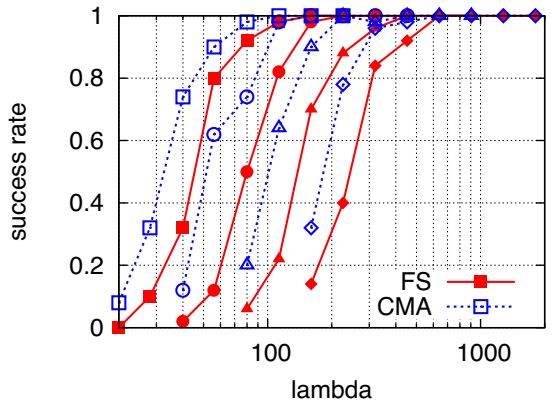

Figure 4: Success rate to reach f_{stop} versus population size on Schaffer function for dimensions $n = 10(-\square-), 20(-\circ-), 40(-\triangle-), 80(-\diamond-)$ for CMA-ES (dotted lines with open symbols) and FS-CMA-ES (solid lines with filled symbols). This figure represents the dependency on success rate for all of 4 multimodal test function as a typical picture.

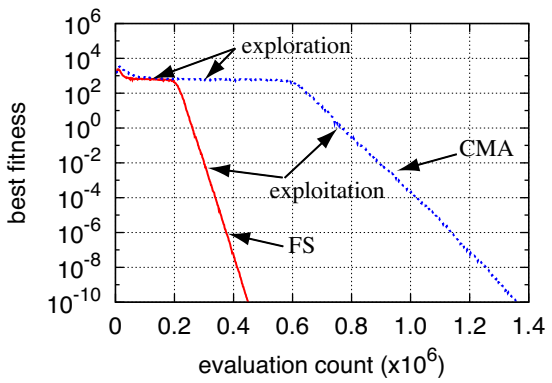

Figure 5: Convergence graph for typical one simulation result on 80 dimensional Rastrigin function for FS-CMA-ES (FS) and standard CMA-ES (CMA) with $\lambda = 1600$.

Figure 5 shows a typical convergence graph on Rastrigin function. The first stage of search is spent on the exploration of the peak where the optimum is located, and after it, CMA-ES (including standard and proposal) does the function evaluation on exploitation of the optimum as well as on single peak functions. The performance improvement appears on exploration stage as well as exploitation stage.

5. DISCUSSION

This section discusses something to be validated but does not yet.

Which normalization is better, trace or determinant? In Sect. 3.2 we propose determinant normalization and trace normalization of $\boldsymbol{C}$, but Section 4 shown only the result using determinant normalization. We conducted the same experiments in Sect. 4 with trace normalization and gave the same result with determinant normalization except for one feature that one eigenvalue of $\boldsymbol{C}$ is enlarged by rank-one update and causes instability of eigenvalues. We think this is because trace normalization has a tendency to make smaller covariance matrix than determinant normalization and so is easier to be affected by rank-one update. Fortunately the effect is prevented by only use of rank-μ update or by adjustment of hybrid rate between rank-one update and rank-μ update (choose smaller $1/\mu_{\text{cov}}$). Trace normalization is better than determinant one from the view point of computational complexity, but above-mentioned fact should be considered.

Is the normalization needed? We conducted the same experiments in Sect. 4 with and without normalization. CMA without normalization can locate the global optimum more efficiently than with normalization on unimodal test functions except for Rosenbrock function, in particular larger populations. On multimodal test functions CMA without normalization needs larger populations and the serial performance is as well as with normalization. But in terms of parallel performances, the number of generations to locate the global optimum without normalization is maximum 50 percent less than with normalization. It is the reason for good performance of the CMA-ES without normalization that a CMA without normalization works like a step size adaptation using ν_σ after adaptation of relation between variables. On the other hand, CMA with normalization can locate the global optimum on Rosenbrock function with smaller function evaluations than without normalization. This is thought because of the same reason discussed in Sect. 4.2 (see also Fig. 3). These results imply that a small change of σ update rule in the Hybrid-SSA could improve the FS-CMA-ES (with normalization) in search efficiency on many functions, without affecting good performance on Rosenbrock function. Further research of this is needed.

6. CONCLUSION

This paper presented a new framework of the derandomized evolution strategy with covariance matrix adaptation. This paper aimed reducing the number of generations and modified the CMA-ES from the viewpoint of making better use of step size adaptation. The main idea of modification was semantically specializing the function of covariance matrix adaptation and step size adaptation. The proposed CMA-ES was evaluated on 8 classical unimodal and multimodal test functions and the performance was compared with standard CMA-ES. The experimental result demonstrated the improvement of search performances, in particular under large populations.

Future work could focus on the theoretical and experimental analysis of Hybrid-SSA and the confirmation of the normalization of covariance matrix discussed in Sect. 5.

7. REFERENCES

[1] A. Auger and N. Hansen. Performance evaluation of an advanced local search evolutionary algorithm. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005*, pages 1777–1784, 2005.

[2] A. Auger and N. Hansen. A restart cma evolution strategy with increasing population size. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005*, pages 1768–1776, 2005.

[3] N. Hansen. Invariance, self-adaptation and correlated mutations in evolution strategies. In *Sixth*

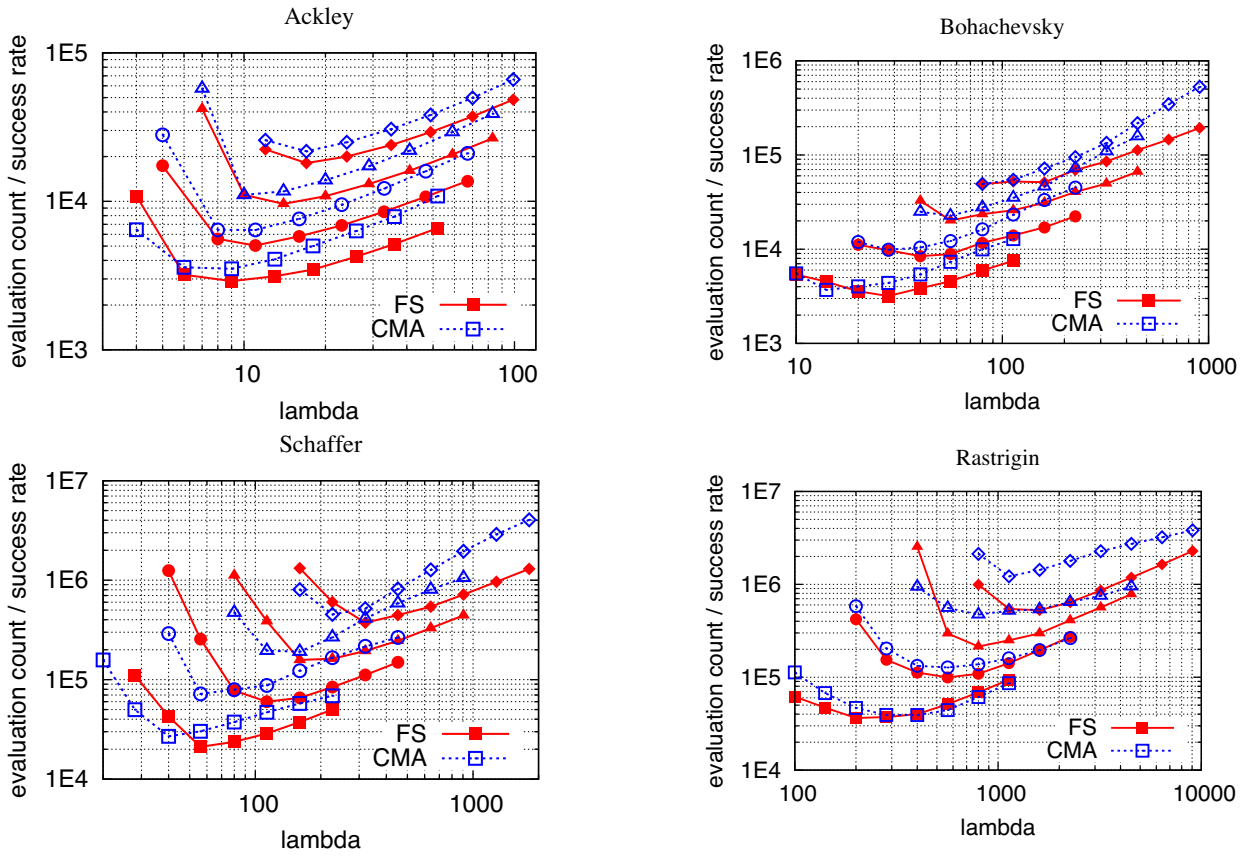

Figure 6: Averaged number of function evaluations to reach $f_{stop} = 10^{-10}$ of success trials over 50 trials divided by the success rate, versus population size for CMA-ES (dotted lines with open symbols), FS-CMA-ES (solid lines with filled symbols), on Ackley, Bohachevsky, Schaffer and Rastrigin function for dimensions $n = 10(-\square-), 20(-\circ-), 40(-\triangle-), 80(-\diamond-)$.

International Conference on Parallel Problem Solving from Nature PPSN VI, Proceedings, pages 355–364, Berlin, 2000. Springer.

[4] N. Hansen. The CMA evolution strategy: a comparing review. In J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, editors, *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pages 75–102. Springer, 2006.

[5] N. Hansen and S. Kern. Evaluating the cma evolution strategy on multimodal test functions. In *Eighth International Conference on Parallel Problem Solving from Nature PPSN VIII, Proceedings*, pages 282–291, Berlin, 2004. Springer.

[6] N. Hansen, S. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.

[7] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 1996*, pages 312–317, 1996.

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

[9] G. A. Jastrebski and D. V. Arnold. Improving evolution strategies through active covariance matrix adaptation. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2006*, pages 9719–9726, 2006.

[10] S. Kern, S. Müller, N. Hansen, D. Büche, J. Ocenasek, and P. Koumoutsakos. Learning probability distributions in continuous evolutionary algorithms–a comparative review. *Natural Computing*, 3(1):77–112, 2004.

[11] S. D. Müller, N. Hansen, and P. Koumoutsakos. Increasing the serial and the parallel performance of the cma-evolution strategy with large populations. In *Seventh International Conference on Parallel Problem Solving from Nature PPSN VII, Proceedings*, pages 422–431, Berlin, 2002. Springer.

[12] A. Ostermeier, A. Gawelczyk, and N. Hansen. Step-size adaptation based on non-local use of selection information. In Springer, editor, *Eighth International Conference on Parallel Problem Solving from Nature PPSN VIII, Proceedings*, pages 189–198, Jerusalem, 1994.