Projection-Based Restricted Covariance Matrix Adaptation for High Dimension

Youhei Akimoto
Faculty of Engineering, Shinshu University
y_akimoto@shinshu-u.ac.jp

Nikolaus Hansen
Inria, Research Centre Saclay
LRI, Univ. Paris-Sud
Université Paris-Saclay, France
lastname@lri.fr

ABSTRACT

We propose a novel variant of the covariance matrix adaptation evolution strategy (CMA-ES) using a covariance matrix parameterized with a smaller number of parameters. The motivation of a restricted covariance matrix is twofold. First, it requires less internal time and space complexity that is desired when optimizing a function on a high dimensional search space. Second, it requires less function evaluations to adapt the covariance matrix if the restricted covariance matrix is rich enough to express the variable dependencies of the problem. In this paper we derive a computationally efficient way to update the restricted covariance matrix where the model richness of the covariance matrix is controlled by an integer and the internal complexity per function evaluation is linear in this integer times the dimension, compared to quadratic in the dimension in the CMA-ES. We prove that the proposed algorithm is equivalent to the sep-CMA-ES if the covariance matrix is restricted to the diagonal matrix, it is equivalent to the original CMA-ES if the matrix is not restricted. Experimental results reveal the class of efficiently solvable functions depending on the model richness of the covariance matrix and the speedup over the CMA-ES.

Keywords

Covariance Matrix Adaptation; Restricted Covariance Matrix; Large Scale Optimization

1. INTRODUCTION

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [6–8] is a state-of-the-art search algorithm for black-box continuous optimization. It is a derivative-free, ranking-based, and stochastic algorithm that maintains the multivariate normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{C})$ from which candidate solutions are generated. It exhibits invariance to several transformations of the optimization problem, which is essential for black-box optimization where a priori knowledge is limited, namely the invariance to any strictly increasing transformation of the objective function, the invariance to any translation and any non-singular linear transformation of the search space.

Generally, the bottleneck of the computational time in black-box optimization is in the evaluation of each candidate solution and we focus on approaching the optimum with a number of function evaluations as small as possible. However, when solving an optimization problem in a high dimensional search space, the internal time and space complexity of the search algorithm can be the bottleneck. If we have a huge number of variables to be optimized but the evaluation of each candidate solution requires a constant time or scales up linearly in the number d of variables, the internal time complexity of the CMA-ES of $\mathcal{O}(d^2)$ per evaluation of each candidate is the bottleneck. The quadratic time complexity of the CMA-ES is due to the covariance matrix $\mathbf{C}$, whose number of elements is $d(d+1)/2$.

To achieve less time and space internal complexity, several variants of the CMA-ES have been proposed [1, 9, 13]. The common strategy is to restrict the covariance matrix so that it can be expressed by a smaller number of parameters. For example, the sep-CMA-ES [13] models the covariance matrix as a diagonal matrix, and the update of the covariance matrix is performed in $\mathcal{O}(d)$, resulting in a linear time and space complexity per function evaluation. Having less elements to parameterize the covariance matrix is advantageous not only in the computational complexity, but also in the number of function evaluations. Since it has less parameters, the adaptation time (in number of function evaluations) is shorter. For the optimization in a high dimensional search space, we do not expect that all the variables are highly correlated, but comparatively fewer dependencies between variables. Therefore, having a restricted covariance matrix is advantageous in these two aspects—we want to have as few parameters to express the covariance matrix as it is required to express the variable dependencies of the problem.

In this paper we propose a novel variant of the CMA-ES with the covariance matrix $\mathbf{C} = \mathbf{D}(\mathbf{I}+\mathbf{V}\mathbf{V}^{\mathrm{T}})\mathbf{D}$, where $\mathbf{D}$ is a diagonal matrix and $\mathbf{V}$ is a $d \times k$ matrix for $k \in [\![0, d-1]\!]$. The model of the covariance matrix used in this paper is a generalization of the models used in VD-CMA [1] and LM-CMA [9]. The update rules of $\mathbf{D}$ and $\mathbf{V}$ are derived based on the projection of the covariance matrix updated by the CMA-ES onto the space of the restricted covariance matrices. The resulting space complexity is $\mathcal{O}(dr)$ and the time complexity per function evaluation is $\mathcal{O}(dr \max(1, r/\lambda))$, where $r = k + \mu + 1$, λ and μ are the number of samples and selected points per iteration, respectively. We prove that the proposed algorithm is equivalent to sep-CMA-ES if $k = 0$,

GECCO '16, July 20 - 24, 2016, Denver, CO, USA

ISBN 978-1-4503-4206-3/16/07…$15.00

DOI: http://dx.doi.org/10.1145/2908812.2908863

and is equivalent to CMA-ES if $k = d - 1$.

The following of this paper is organized as follows. In Section 2 we introduce the CMA-ES and describe its computational complexity. The existing variants of the CMA-ES are reviewed. In Section 3, we propose a novel variant of the CMA-ES with a restricted covariance matrix, named VkD-CMA, and show the connection to the sep-CMA-ES and the CMA-ES. We conduct the experiments on a standard benchmark set and compare the performance with the CMA-ES and its variants in Section 4. We conclude the paper in Section 5 with a summary and a description of future work.

2. CMA-ES AND RELATED WORK

We first review the algorithm of the CMA-ES with Two Point step-size Adaptation (TPA) [5] as the baseline algorithm for the rest of the paper and explain its internal computational time and space complexity that will be a bottleneck when solving a high dimensional optimization problem. Then we introduce several variants of the CMA-ES aiming at reducing the internal cost.

2.1 TPA-CMA-ES

In the CMA-ES, the candidate solutions $x_i \in \mathbb{R}^d$, for $i = 1, \ldots, \lambda$, are drawn independently from the multivariate normal (Gaussian) distribution $\mathcal{N}(\mathbf{m}^{(t)}, (\sigma^{(t)})^2\mathbf{C}^{(t)})$ at each iteration $t \geqslant 0$. The default value for the number of samples (aka population size) is $\lambda = \lfloor 4 + 3\ln(d) \rfloor$. For the sake of notation simplicity, we write $y_i = (x_i - \mathbf{m}^{(t)})/\sigma^{(t)}$ and $z_i = (\mathbf{C}^{(t)})^{-1/2} y_i$ and we drop the iteration counter from x as long as it does not make any confusion. The candidate solutions are evaluated on the given objective $f : \mathbb{R}^d \to \mathbb{R}$ and sorted in the ascending order of their objective values. Let the index of the ith best point among λ current candidate solutions be denoted by $i : \lambda$, i.e., $f(x_{1:\lambda}) \leq \cdots \leq f(x_{\lambda:\lambda})$. We compute the weighted average of the steps $y_{i:\lambda}$,

$$\mathbf{dm}^{(t)} = \sum_{i=1}^{\mu} w_i y_{i:\lambda} \ , \tag{1}$$

where w_i is the weight value assigned to the ith best point $x_{i:\lambda}$ (or equivalently to $y_{i:\lambda}$). Its commonly used value is

$$w_i = \frac{\ln((\lambda+1)/2) - \ln(i)}{\sum_{i=1}^{\mu}(\ln((\lambda+1)/2) - \ln(i))} \tag{2}$$

for $i = 1, \ldots, \mu = \lfloor \lambda/2 \rfloor$ and $w_i = 0$ for $i > \mu$. Let $\mu_{\text{eff}} = 1/(\sum_{i=1}^{\mu} w_i^2)$. With the learning rate c_m, the mean vector is updated as

$$\mathbf{m}^{(t+1)} = \mathbf{m}^{(t)} + c_m \sigma^{(t)} \mathbf{dm}^{(t)} \ . \tag{3}$$

The default value for the learning rate c_m is 1. The evolution path for the rank-one update of the covariance matrix is updated according to

$$\mathbf{p}_c^{(t+1)} = (1 - c_c)\mathbf{p}_c^{(t)} + h_\sigma^{(t)} (c_c(2 - c_c)\mu_{\text{eff}})^{1/2} \mathbf{dm}^{(t)} \ , \tag{4}$$

where c_c is the cumulation factor, i.e., $1/c_c$ is the backward time horizon, and $h_\sigma^{(t)}$ is either 0 or 1 as explained later in the section. The covariance matrix is updated as

$$\begin{aligned} \mathbf{C}^{(t+1)} = {} & (1 - c_\mu - c_1 + (1 - h_\sigma^{(t)}) c_1 c_c (2 - c_c)) \mathbf{C}^{(t)} \\ & + c_\mu \sum_{i=1}^{\mu} w_i y_{i:\lambda} y_{i:\lambda}^{\mathrm{T}} + c_1 (\mathbf{p}_c^{(t+1)})(\mathbf{p}_c^{(t+1)})^{\mathrm{T}} \ , \end{aligned} \tag{5}$$

where c_1 and c_μ are the learning rates for the rank-one update and the rank-μ update, respectively.

The TPA [5] is an alternative to the cumulative step-size adaptation (CSA) [8, 12]. For $t \geqslant 1$, in the sampling phase it produces two symmetric steps along the previous mean shift $\mathbf{dm}^{(t-1)}$ following

$$y_\pm = \pm \frac{\|\mathcal{N}(0, \mathbf{I})\|}{((\mathbf{dm}^{(t-1)})^{\mathrm{T}} (\mathbf{C}^{(t)})^{-1} \mathbf{dm}^{(t-1)})^{1/2}} \mathbf{dm}^{(t-1)} \ . \tag{6}$$

The denominator is the Mahalanobis distance between $\mathbf{m}^{(t)}$ and $\mathbf{m}^{(t-1)}$ with respect to $\mathbf{C}^{(t)}$. Instead of generating x_1 and x_2 from $\mathcal{N}(\mathbf{m}^{(t)}, (\sigma^{(t)})^2\mathbf{C}^{(t)})$, we set them as $x_1 = \mathbf{m}^{(t)} + \sigma^{(t)} y_+$ and $x_2 = \mathbf{m}^{(t)} + \sigma^{(t)} y_-$. The σ-adaptation is based on the ranking difference between these two symmetric points. Based on the accumulated information of the ranking differences

$$s^{(t+1)} = (1 - c_\sigma) s^{(t)} + c_\sigma \frac{\text{rank}(x_2) - \text{rank}(x_1)}{\lambda - 1} \ , \tag{7}$$

we update the step-size according to

$$\sigma^{(t+1)} = \sigma^{(t)} \exp(s^{(t+1)}/d_\sigma) \ , \tag{8}$$

where $c_\sigma = 0.3$ and $d_\sigma = d^{1/2}$. The $s^{(t+1)}$ tends to be positive if x_1 is better than x_2, meaning that $s^{(t+1)} > 0$ if the previous mean shift still provides a good direction and vice versa. Then, we make the step-size larger since we can still expect the progress by taking longer steps. To prevent $\|\mathbf{p}_c\|$ from increasing rapidly and $\mathbf{C}$ from changing its shape rapidly due to a long $\mathbf{p}_c$, we stall the rank-one update when the step-size is significantly increasing, by setting $h_\sigma^{(t)} = \mathbb{I}\{s^{(t+1)} < 0.5\}$ in (4). See Section 7.2 of [6] for the detailed intuition into h_σ and $\mathbf{p}_c$.

The CMA-ES with TPA repeats the sampling of candidate solutions and the update of the parameters $\mathbf{m}$, σ, and $\mathbf{C}$ according to (3), (8), and (5), respectively, until a termination criterion is satisfied. All the static parameters in the algorithm such as the learning rates have the default values as appeared above. The cumulation factor c_c, the learning rates for rank-one update and for rank-μ update, c_1 and c_μ, respectively, are set by default as follows

$$\begin{aligned} & c_c = \frac{4 + \mu_{\text{eff}}/d}{d + 4 + 2\mu_{\text{eff}}/d}, \qquad c_1 = \frac{2}{(d + 1.3)^2 + \mu_{\text{eff}}}, \\ & c_\mu = \min\left(1 - c_1, \frac{2(\mu_{\text{eff}} - 2 + 1/\mu_{\text{eff}})}{(d+2)^2 + \mu_{\text{eff}}}\right). \end{aligned} \tag{9}$$

2.2 Variants for Large Scale Optimization

One of the bottleneck of the CMA-ES when applied to solve an optimization problem in high dimension ($d \gg 100$) is its time and space complexity. To store the covariance matrix and candidate solutions it requires $\mathcal{O}(d^2 + d\mu)$ floating point storage. To update the covariance matrix in (5), $\mathcal{O}(d^2\mu)$ floating point operations are required. To draw samples from $\mathcal{N}(\mathbf{m}, \sigma^2\mathbf{C})$, we need to produce $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and compute $x = \mathbf{m} + \sigma\mathbf{B}z$, where $\mathbf{B}$ is a real-valued matrix satisfying $\mathbf{C} = \mathbf{B}\mathbf{B}^{\mathrm{T}}$. Additionally, to generate a pair of symmetric samples (6), one needs to solve $x = (\mathbf{C}^{(t)})^{-1}\mathbf{dm}^{(t-1)}$. For this purpose, we compute the eigen decomposition of $\mathbf{C}$ that costs $\mathcal{O}(d^3)$ floating point operations. Performing the eigen decomposition every $\mathcal{O}(d/\lambda)$ iterations reduces the time complexity per f-call to $\mathcal{O}(d^2)$ without deteriorating the performance [4]. In total, the number of floating point operations is $\mathcal{O}(d^2)$ per f-call and $\mathcal{O}(d^2\mu)$ per iteration.

The other bottleneck is the adaptation time for the covariance matrix. It is empirically known that the learning rates c_μ and c_1 need to be proportional to $1/d^2$ or slightly higher to stabilize the adaptation. One may explain that it is because the covariance matrix consists of $d(d+1)/2$ elements. The smaller the learning rates are, the more stable the covariance matrix adaptation is, but the more adaptation time is required.

To tackle these issues, several variants of the CMA-ES have been proposed. The common strategy is to restrict the covariance matrix so that it can be expressed with less parameters. Since the number of parameters is smaller, one can set higher learning rates, resulting in reducing the adaptation time. Moreover, if one can find computationally cheap ways to sample a normal random vector with such a restricted covariance matrix and to update the parameters including the covariance matrix, the internal time complexity will be reduced.

The sep-CMA-ES [13] models the covariance matrix as a diagonal matrix. The covariance matrix adaptation is done by taking the diagonal elements of $\mathbf{C}^{(t+1)}$ in (5) given a diagonal $\mathbf{C}^{(t)}$, which can be done within $\mathcal{O}(d)$ floating point operations. This update is interpreted in two ways. One interpretation is the projection of $\mathbf{C}^{(t+1)}$ from the manifold of positive definite symmetric matrices to its sub-manifold of positive definite diagonal matrices. If the projection is defined by the shortest distance in terms of the Frobenius norm, the diagonal matrix closest to $\mathbf{C}^{(t+1)}$ is the diagonal matrix $\mathrm{diag}(\mathbf{C}^{(t+1)})$ consisting of its diagonal elements. The other interpretation is the natural gradient on the manifold of the Gaussian distributions with a diagonal covariance matrix [2], whereas the original update (5) is based on the natural gradient on a manifold of the Gaussian distributions with a positive definite symmetric covariance matrix.

The VD-CMA-ES [1] maintains a vector $\mathbf{v}$ in addition to a diagonal matrix $\mathbf{D}$ and the covariance matrix is $\mathbf{D}(\mathbf{I} + \mathbf{v}\mathbf{v}^\mathrm{T})\mathbf{D}$. The diagonal matrix learns the scaling of each variable and the vector allows to learn the correlation of variables. Due to the richness of the covariance matrix, the VD-CMA-ES can efficiently solve more functions than the sep-CMA-ES. The update of the covariance matrix is based on the natural gradient, so it is derived from the same design principle as the CMA-ES and the sep-CMA-ES. On the other hand, inspired by L-BFGS [11], the LM-CMA-ES [9] restricts the covariance matrix to $\mathbf{I} + \mathbf{V}\mathbf{V}^\mathrm{T}$, where $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$. Instead of keeping a diagonal matrix, it maintains k vectors to express more variable dependencies.

3. VkD-CMA

We restrict the covariance matrix $\mathbf{C}$ of the sampling distribution to be of the form

$$\mathbf{C} = \mathbf{D}(\mathbf{I} + \mathbf{V}\mathbf{V}^\mathrm{T})\mathbf{D} \ , \tag{10}$$

where $\mathbf{D}$ is a diagonal matrix and $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$ is a $d \times k$ matrix. The diagonal matrix $\mathbf{D}$ is interpreted as the coordinate-wise scaling of the search space. Each column $\mathbf{v}_i$ of the rank-k matrix $\mathbf{V}$ represents a long direction of the sampling distribution. The covariance matrix is then parameterized with $d(k+1)$ elements. The value of k is in $[\![0, d-1]\!]$, where we drop $\mathbf{V}$ from (10) if $k = 0$.

Let $\mathcal{M}_k$ be the set of positive-definite symmetric matrices expressed in (10). Then, $\mathcal{M}_0$ is the set of all positive-definite diagonal matrices, and $\mathcal{M}_{d-1}$ is the set of all positive-definite symmetric matrices as we will see later. Moreover, $\mathcal{M}_i \subset \mathcal{M}_{i+1}$ for $i = 0, \ldots, d-2$ and $\mathcal{M}_i = \mathcal{M}_{i+1}$ for $i \geqslant d-1$.

We derive a novel variant of CMA-ES with a restricted covariance matrix in $\mathcal{M}_k$ for any given $k \in [\![0, d-1]\!]$. It emulates the original CMA-ES described in Section 2.1 with $\mathbf{C} \in \mathcal{M}_k$. All the steps except the covariance matrix adaptation can be computed without any modification and their computational complexity reduces, as we will see in Section 3.2. For the covariance matrix adaptation, we project $\mathbf{C}^{(t+1)}$ in (5) onto $\mathcal{M}_k$. To do so, we approximate the following optimization problem without computing $\mathbf{C}^{(t+1)}$,

$$\underset{(\mathbf{D},\mathbf{V})}{\mathrm{argmin}} \|\mathbf{D}(\mathbf{I} + \mathbf{V}\mathbf{V}^\mathrm{T})\mathbf{D} - \mathbf{C}^{(t+1)}\|_F \ , \tag{11}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. We derive a computationally cheap method to approximate the solution to (11) in Section 3.1. In the following, we denote the (i, j)th element of a matrix $\mathbf{A}$ by $[\mathbf{A}]_{i,j}$ and the ith diagonal element of a diagonal matrix $\mathbf{A}$ by $[\mathbf{A}]_i$.

3.1 Restricted Covariance Matrix Adaptation

To have a computationally efficient update formula, we approximate the solution to the optimization problem (11) by the following two-step procedure. First, we solve

$$\underset{(\beta,\mathbf{V})}{\mathrm{argmin}} \|(\beta\mathbf{I} + \mathbf{V}\mathbf{V}^\mathrm{T}) - (\mathbf{D}^{(t)})^{-1}\mathbf{C}^{(t+1)}(\mathbf{D}^{(t)})^{-1}\|_F \ . \tag{12}$$

Let β^* and $\mathbf{V}^*$ be the optimal parameters given above. Then, we obtain $\mathbf{D}^*$ by solving

$$\mathrm{diag}(\mathbf{D}(\beta^*\mathbf{I} + \mathbf{V}^*(\mathbf{V}^*)^\mathrm{T})\mathbf{D}) = \mathrm{diag}(\mathbf{C}^{(t+1)}) \ . \tag{13}$$

Finally, we update $\mathbf{V}^{(t+1)}$ and $\mathbf{D}^{(t+1)}$ by cancelling β^* as $\mathbf{V}^{(t+1)} = (\beta^*)^{-1/2}\mathbf{V}^*$ and $\mathbf{D}^{(t+1)} = (\beta^*)^{1/2}\mathbf{D}^*$.

3.1.1 *Derivation*

Given $\mathbf{C}^{(t)} = \mathbf{D}^{(t)}(\mathbf{I} + \mathbf{V}^{(t)}(\mathbf{V}^{(t)})^\mathrm{T})\mathbf{D}^{(t)}$, we can rewrite (5) as

$$\mathbf{C}^{(t+1)} = \mathbf{D}^{(t)}(\alpha_c\mathbf{I} + \alpha_c\mathbf{V}^{(t)}(\mathbf{V}^{(t)})^\mathrm{T} + \mathbf{Y}\mathbf{Y}^\mathrm{T})\mathbf{D}^{(t)} \ , \tag{14}$$

where $\alpha_c = 1 - c_\mu - c_1 + (1 - h_\sigma^{(t)})c_1c_c(2 - c_c)$ and $\mathbf{Y}$ is a $d \times (\mu+1)$ dimensional matrix whose first μ columns are given by $(c_\mu w_i)^{1/2}(\mathbf{D}^{(t)})^{-1}y_{i:\lambda}$ for $i = 1, \ldots, \mu$ and the last column is $c_1^{1/2}(\mathbf{D}^{(t)})^{-1}\mathbf{p}_c^{(t+1)}$. Remember that $y_i = (x_i - \mathbf{m}^{(t)})/\sigma^{(t)}$ and x_i is drawn from $\mathcal{N}(\mathbf{m}^{(t)}, (\sigma^{(t)})^2\mathbf{C}^{(t)})$ independently for $i = 1, \ldots, \lambda$.

Let $\mathbf{W} = [\alpha_c^{1/2}\mathbf{V}^{(t)}, \mathbf{Y}]$, $r = k + \mu + 1$, and the singular value decomposition (SVD) of $\mathbf{W}$ be denoted by $\mathbf{L}\mathbf{S}\mathbf{R}^\mathrm{T}$, where $\mathbf{S}$ is a $d \times r$ dimensional diagonal matrix whose first $\bar{r} = \min(r, d)$ diagonal elements are the singular values of $\mathbf{W}$, $\mathbf{L}$ and $\mathbf{R}$ are orthogonal matrices of dimension d and r, respectively, whose first $\bar{r}$ columns are the left and right unit singular vectors and the other columns of $\mathbf{L}$ are arbitrary unit vectors that results in $\mathbf{L}$ being orthogonal. Let $\mathbf{L}_{:i}$ be the first i columns of $\mathbf{L}$ and $\mathbf{S}_{:i}$ be the upper left $i \times i$ block of $\mathbf{S}$. Note that $\mathbf{W} = \mathbf{L}\mathbf{S}\mathbf{R}^\mathrm{T} = \mathbf{L}_{:\bar{r}}\mathbf{S}_{:\bar{r}}\mathbf{R}_{:\bar{r}}^\mathrm{T}$ and the rightmost side is called the thin SVD. Without loss of generality we assume that the diagonal elements of $\mathbf{S}$ are aligned in descending order, i.e., $[\mathbf{S}]_{1,1} \geqslant \ldots \geqslant [\mathbf{S}]_{\bar{r},\bar{r}}$.

The optimal solution $\mathbf{V}^*$ to (12) is not unique since we have $\mathbf{V}\mathbf{V}^\mathrm{T} = (\mathbf{V}\mathbf{R})(\mathbf{V}\mathbf{R})^\mathrm{T}$ for arbitrary orthogonal $\mathbf{R}$ of dimension k. Theorem 1 provides an optimal solution $\mathbf{V}^*$ whose columns are orthogonal to each other. It requires the

thin SVD of $\mathbf{W}$ and it can be computed with $\mathcal{O}(d\bar{r}^2)$ floating point operations.

THEOREM 1. *One of the optimal solutions to* (12) *is given by* $\mathbf{V}^* = \mathbf{L}_{:k}((\alpha_c - \beta^*)\mathbf{I} + \mathbf{S}^2_{:k})^{1/2}$ *and* $\beta^* = \alpha_c + (d-k)^{-1}\sum_{i=k+1}^{\bar{r}}[\mathbf{S}]^2_{i,i}$.

PROOF. Rewriting (12) we have

$$\begin{aligned}&\|(\beta\mathbf{I} + \mathbf{V}\mathbf{V}^{\mathrm{T}}) - (\mathbf{D}^{(t)})^{-1}\mathbf{C}^{(t+1)}(\mathbf{D}^{(t)})^{-1}\|_F \\ &\quad = \|(\beta\mathbf{I} + \mathbf{V}\mathbf{V}^{\mathrm{T}}) - (\alpha_c\mathbf{I} + \mathbf{W}\mathbf{W}^{\mathrm{T}})\|_F \\ &\quad = \|\mathbf{V}\mathbf{V}^{\mathrm{T}} - ((\alpha_c - \beta)\mathbf{I} + \mathbf{W}\mathbf{W}^{\mathrm{T}})\|_F \ .\end{aligned} \tag{15}$$

If β is fixed, the problem (12) is to find the optimal nonnegative definite rank-k approximation $\mathbf{V}\mathbf{V}^{\mathrm{T}}$ of the symmetric matrix $(\alpha_c-\beta)\mathbf{I}+\mathbf{W}\mathbf{W}^{\mathrm{T}}$, which is given by the SVD of the matrix [10]. Note that $\mathbf{I} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$ and $\mathbf{W}\mathbf{W}^{\mathrm{T}} = \mathbf{L}\mathbf{S}\mathbf{S}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}$. The SVD of the matrix that we want to approximate is then written as $(\alpha_c-\beta)\mathbf{I}+\mathbf{W}\mathbf{W}^{\mathrm{T}} = \mathbf{L}((\alpha_c-\beta)\mathbf{I}+\mathbf{S}\mathbf{S}^{\mathrm{T}})\mathbf{L}^{\mathrm{T}}$. Then the optimal $\mathbf{V}$ is given by $\mathbf{L}_{:k}((\alpha_c-\beta)\mathbf{I}+\mathbf{S}^2_{:k})^{1/2}$. Given this $\mathbf{V}$, since the square of the Frobenius norm of a matrix is the sum of the square of the singular values of it, the optimal β is given by $\alpha_c + (d-k)^{-1}\sum_{i=k+1}^{\bar{r}}[\mathbf{S}]^2_{i,i}$. □

The update of $\mathbf{D}$ can be computed in $\mathcal{O}(dr)$ as follows.

THEOREM 2. *Given an optimal solution* $(\beta^*, \mathbf{V}^*)$ *of* (12)*, the unique solution* $\mathbf{D}^*$ *to* (13) *satisfies, for* $i = 1, \ldots, d$*,*

$$[\mathbf{D}^*]_i = [\mathbf{D}^{(t)}]_i \left(\frac{\alpha_c + \sum_{j=1}^{r}[\mathbf{W}]^2_{i,j}}{\beta^* + \sum_{j=1}^{k}[\mathbf{V}^*]^2_{i,j}}\right)^{1/2} \ .$$

PROOF. Since (13) can be written as

$$\begin{aligned}&\operatorname{diag}(\mathbf{D}(\beta^*\mathbf{I} + \mathbf{V}^*(\mathbf{V}^*)^{\mathrm{T}})\mathbf{D}) \\ &\quad = \mathbf{D}\operatorname{diag}((\beta^*\mathbf{I} + \mathbf{V}^*(\mathbf{V}^*)^{\mathrm{T}}))\mathbf{D} \\ &\quad = \mathbf{D}^{(t)}\operatorname{diag}(\alpha_c\mathbf{I} + \mathbf{W}\mathbf{W}^{\mathrm{T}})\mathbf{D}^{(t)} = \operatorname{diag}(\mathbf{C}^{(t+1)}) \ ,\end{aligned}$$

it is easy to see that $\mathbf{D}^* = \mathbf{D}^{(t)}\big(\operatorname{diag}(\alpha_c\mathbf{I}+\mathbf{W}\mathbf{W}^{\mathrm{T}})\operatorname{diag}(\beta^*\mathbf{I}+\mathbf{V}^*(\mathbf{V}^*)^{\mathrm{T}})^{-1}\big)^{1/2}$. □

3.1.2 Properties

Generally, the solutions to (12) and (13) and the solutions to (11) disagree. However, they agree if either $k = 0$ or $k = d-1$. Moreover, the update of the covariance matrix is consistent with the update in the sep-CMA-ES if $k = 0$, and with the update in the CMA-ES if $k = d-1$.

THEOREM 3. *If* $k = 0$*, the solution to* (11) *and the solution to* (13) *admit the unique solution* $\mathbf{D}^* = \operatorname{diag}(\mathbf{C}^{(t+1)})^{1/2}$.

PROOF. Since the squared Frobenius norm is equal to the sum of the square of all the elements, we can easily verify that $\operatorname{diag}(\mathbf{C}^{(t+1)})^{1/2} = \operatorname{argmin}_{\mathbf{D}}\|\mathbf{D}^2 - \mathbf{C}^{(t+1)}\|_F$. This is also the solution to (13). □

THEOREM 4. *If* $k = d-1$*, any solution* $(\beta^*, \mathbf{V}^*)$ *to* (12) *satisfies* $\beta^*\mathbf{I} + \mathbf{V}^*(\mathbf{V}^*)^{\mathrm{T}} = (\mathbf{D}^{(t)})^{-1}\mathbf{C}^{(t+1)}(\mathbf{D}^{(t)})^{-1}$*. Given any solution* $(\beta^*, \mathbf{V}^*)$ *to* (12)*, the solution to* (13) *is given by* $\mathbf{D}^* = \mathbf{D}^{(t)}$*. Moreover,* $\bar{\mathbf{D}} = (\beta^*)^{1/2}\mathbf{D}^*$ *and* $\bar{\mathbf{V}} = (\beta^*)^{-1/2}\mathbf{V}^*$ *form a solution of* (11) *and* $\mathbf{C}^{(t+1)} = \bar{\mathbf{D}}(\mathbf{I} + \bar{\mathbf{V}}\bar{\mathbf{V}}^{\mathrm{T}})\bar{\mathbf{D}}$.

PROOF. For the first statement, because of (15), it is enough to show that the value of the right-most side of (15) reaches zero for some $(\beta, \mathbf{V})$. Let $\beta = \alpha_c + [\mathbf{S}\mathbf{S}^{\mathrm{T}}]_d$. Then, we have $(\alpha_c - \beta)\mathbf{I} + \mathbf{W}\mathbf{W}^{\mathrm{T}} = \mathbf{L}((\alpha_c - \beta)\mathbf{I} + \mathbf{S}\mathbf{S}^{\mathrm{T}})\mathbf{L}^{\mathrm{T}} = \mathbf{L}_{:d-1}([\mathbf{S}\mathbf{S}^{\mathrm{T}}]_{:d-1} - [\mathbf{S}\mathbf{S}^{\mathrm{T}}]_d\mathbf{I})\mathbf{L}^{\mathrm{T}}_{:d-1}$. It is of rank at most $d-1$, symmetric, and nonnegative definite. Since it can be expressed by $\mathbf{V}\mathbf{V}^{\mathrm{T}}$, the solution to (12) must reach the optimal value of zero. This ends the proof for the first statement.

Substituting $\beta\mathbf{I}+\mathbf{V}\mathbf{V}^{\mathrm{T}} = (\mathbf{D}^{(t)})^{-1}\mathbf{C}^{(t+1)}(\mathbf{D}^{(t)})^{-1}$ in (13), we have $\mathbf{D}^2(\mathbf{D}^{(t)})^{-2}\operatorname{diag}(\mathbf{C}^{(t+1)}) = \operatorname{diag}(\mathbf{C}^{(t+1)})$. The solution to this equation is $\mathbf{D} = \mathbf{D}^{(t)}$. This ends the proof for the second statement.

Since $\beta\mathbf{I} + \mathbf{V}\mathbf{V}^{\mathrm{T}} = (\mathbf{D}^{(t)})^{-1}\mathbf{C}^{(t+1)}(\mathbf{D}^{(t)})^{-1}$ from the first statement and $\mathbf{D} = \mathbf{D}^{(t)}$ from the second statement, we have $(\beta^{1/2}\mathbf{D})(\mathbf{I}+(\beta^{-1/2}\mathbf{V})(\beta^{-1/2}\mathbf{V})^{\mathrm{T}})(\beta^{1/2}\mathbf{D}) = \mathbf{D}^{(t)}(\beta\mathbf{I}+\mathbf{V}\mathbf{V}^{\mathrm{T}})\mathbf{D}^{(t)} = \mathbf{C}^{(t+1)}$. This completes the proof. □

3.1.3 Learning Rate and Cumulation Factor

Since the number of parameters for the covariance matrix is $d(k+1)$, we set the cumulation factor, the learning rate for rank-one update, the learning rate for rank-μ update as follows

$$\begin{aligned}c_c &= \frac{4 + \mu_{\mathrm{eff}}/d}{(d + 2(k+1))/3 + 4 + 2\mu_{\mathrm{eff}}/d} \ , \\ c_1 &= \frac{2}{d(k+1) + 2(k+2) + \mu_{\mathrm{eff}}} \ , \\ c_\mu &= \min\left(1 - c_1, \frac{2(\mu_{\mathrm{eff}} - 2 + 1/\mu_{\mathrm{eff}})}{d(k+1) + 4(k+2) + \mu_{\mathrm{eff}}}\right) \ .\end{aligned} \tag{16}$$

The learning rates, c_1 and c_μ, are proportional to $(d(k+1))^{-1}$. If $k = d-1$, they are very close to the values used in the CMA-ES (9). If $k = 0$, they are proportional to d^{-1} and similar to the values used in the sep-CMA-ES.

3.2 Efficient Sampling and Mahalanobis Distance Computation

We derive computationally efficient ways to generate a sample from $\mathcal{N}(\mathbf{m}, \sigma^2\mathbf{C})$ with $\mathbf{C} \in \mathcal{M}_k$ and to compute the Mahalanobis norm of the mean shift, $(\mathbf{d}\mathbf{m}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{d}\mathbf{m})^{1/2}$, used in (6).

Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$ and assume $\|\mathbf{v}_1\| \geqslant \ldots \geqslant \|\mathbf{v}_k\|$ without loss of generality. Let $\mathbf{\Lambda} = \operatorname{diag}(\|\mathbf{v}_1\|^2, \ldots, \|\mathbf{v}_k\|^2)$ and $\tilde{\mathbf{V}} = \mathbf{V}\mathbf{\Lambda}^{-1/2} = [\mathbf{v}_1/\|\mathbf{v}_1\|, \ldots, \mathbf{v}_k/\|\mathbf{v}_k\|] = [\tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_k]$. By construction, $\mathbf{v}_i$, for $i = 1, \ldots, k$, obtained in Theorem 1 are orthogonal to each other. Then, the ith largest eigenvalue of $\mathbf{I}+\mathbf{V}\mathbf{V}^{\mathrm{T}}$ is $1+\|\mathbf{v}_i\|^2$ for $i = 1, \ldots, k$ and 1 for $i = k+1, \ldots, d$. The normal eigenvector corresponding to the ith eigenvalue is $\tilde{\mathbf{v}}_i$ for $i = 1, \ldots, k$. The eigenspace corresponding to the eigenvalue of 1 is spanned by any set of $d-k$ linearly independent vectors that are orthogonal to $\tilde{\mathbf{v}}_i$ for all $i = 1, \ldots, k$. The following propositions are straightforward.

PROPOSITION 1. *The square root matrix of* $\mathbf{I} + \mathbf{V}\mathbf{V}^{\mathrm{T}}$ *is* $\mathbf{I} + \tilde{\mathbf{V}}[(\mathbf{\Lambda} + \mathbf{I})^{1/2} - \mathbf{I}]\tilde{\mathbf{V}}^{\mathrm{T}}$.

PROPOSITION 2. *The inverse of* $\mathbf{D}(\mathbf{I}+\mathbf{V}\mathbf{V}^{\mathrm{T}})\mathbf{D}$ *is* $\mathbf{D}^{-1}(\mathbf{I}+\tilde{\mathbf{V}}((\mathbf{I}+\mathbf{\Lambda})^{-1} - \mathbf{I})\tilde{\mathbf{V}}^{\mathrm{T}})\mathbf{D}^{-1}$.

With Proposition 1, we can derive an efficient sampling as follows. Given λ independently and d-variate normally distributed random vectors $z_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, compute

$$\begin{aligned}&y_i \leftarrow \tilde{\mathbf{V}}^{\mathrm{T}}z_i, \quad y_i \leftarrow ((\mathbf{\Lambda} + \mathbf{I})^{1/2} - \mathbf{I})y_i, \text{ and} \\ &y_i \leftarrow \mathbf{D}(z_i + \tilde{\mathbf{V}}y_i)\end{aligned} \tag{17}$$

and $x_i = \mathbf{m} + \sigma y_i$. Then, $y_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, $x_i \sim \mathcal{N}(\boldsymbol{m}, \sigma^2\mathbf{C})$, and $\{y_i\}_{i=1,\dots,\lambda}$ and $\{x_i\}_{i=1,\dots,\lambda}$, respectively, are independent. Moreover, the computational cost is $\mathcal{O}(dk)$ for each sample. Note that $y_i = \mathbf{D}(\mathbf{I} + \mathbf{V}\mathbf{V}^{\mathrm{T}})^{1/2} z_i$.

The square Mahalanobis distance of the previous mean shift with respect to $\mathbf{C}$ is computed in $\mathcal{O}(dk)$, thanks to Proposition 2. Let $u_1 = \mathbf{D}^{-1}\mathbf{dm}$ and $u_2 = \tilde{\mathbf{V}}^{\mathrm{T}} u_1$. Then,

$$\begin{aligned}
&(\mathbf{dm})^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{dm} \\
&\quad = (\mathbf{D}^{-1}\mathbf{dm})^{\mathrm{T}}(\mathbf{I} + \tilde{\mathbf{V}}((\mathbf{I}+\mathbf{\Lambda})^{-1} - \mathbf{I})\tilde{\mathbf{V}}^{\mathrm{T}})\mathbf{D}^{-1}\mathbf{dm} \\
&\quad = u_1^{\mathrm{T}}(\mathbf{I} + \tilde{\mathbf{V}}((\mathbf{I}+\mathbf{\Lambda})^{-1} - \mathbf{I})\tilde{\mathbf{V}}^{\mathrm{T}})u_1 \\
&\quad = \|u_1\|^2 + (\tilde{\mathbf{V}}^{\mathrm{T}}u_1)^{\mathrm{T}}((\mathbf{I}+\mathbf{\Lambda})^{-1} - \mathbf{I})(\tilde{\mathbf{V}}^{\mathrm{T}}u_1) \\
&\quad = \|u_1\|^2 + u_2^{\mathrm{T}}((\mathbf{I}+\mathbf{\Lambda})^{-1} - \mathbf{I})u_2 \ . \qquad (18)
\end{aligned}$$

3.3 Algorithm

Initialize $\mathbf{m}^{(0)}$, $\sigma^{(0)}$ and $\mathbf{D}^{(0)}$ according to the initial search interval of a given problem, and let $\tilde{\mathbf{V}}^{(0)} = \mathbf{0}$, $\mathbf{\Lambda}^{(0)} = \mathrm{diag}(0, \dots, 0)$, $\mathbf{p}_c^{(0)} = \mathbf{0}$, and $s^{(0)} = 0$. Note that we keep $\tilde{\mathbf{V}}$ and $\mathbf{\Lambda}$ instead of $\mathbf{V} = \tilde{\mathbf{V}}\mathbf{\Lambda}^{1/2}$. The values for c_c, c_1, and c_μ are set according to (16) and all the other parameters are set to the same default values as the CMA-ES described in Section 2.1. Let $t = 0$ and $r = k + \mu + 1$. Repeat the following steps until a termination criterion is satisfied.

1. If $t \geqslant 1$, generate a pair of symmetric points $y_\pm$ along the previous mean shift $\mathbf{dm}^{(t-1)}$ according to (6), where the Mahalanobis distance $(\mathbf{dm}^{(t-1)})^{\mathrm{T}}(\mathbf{C}^{(t)})^{-1}\mathbf{dm}^{(t-1)}$ is computed with the formula (18). Let $y_1 = y_+$, $y_2 = y_-$. If $t = 0$, generate y_1 and y_2 in the same way as in the next step.

2. Sample $\lambda - 2$ independent random vectors $z_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, for $i = 3, \dots, \lambda$, and compute y_i according to (17). Let $x_i = \mathbf{m}^{(t)} + \sigma^{(t)} y_i$ for $i = 1, \dots, \lambda$.

3. Evaluate x_i on the given objective function f, and let the index of the ith best point among them be denoted by $i : \lambda$.

4. Compute the weighted average $\mathbf{dm}^{(t)}$ of the steps $y_{i:\lambda}$ according to (1), and update the mean vector $\mathbf{m}^{(t+1)}$ according to (3).

5. If $t \geqslant 1$, update the step-size $\sigma^{(t+1)}$ according to (7) and (8), and let $h_\sigma^{(t)} = \mathbb{I}\{s^{(t+1)} < 0.5\}$. Otherwise, let $s^{(1)} = s^{(0)}$, $\sigma^{(1)} = \sigma^{(0)}$ and $h_\sigma^{(t)} = 1$.

6. Update the evolution path $\mathbf{p}_c^{(t+1)}$ according to (4).

7. Compute the thin SVD of $\mathbf{W} = \mathbf{L}_{:\bar{r}}\mathbf{S}_{:\bar{r}}\mathbf{R}_{:\bar{r}}^{\mathrm{T}}$, where $\bar{r} = \min(r, d)$, $\mathbf{W}$ is as given in Section 3.1.1, and the singular values are aligned in descending order. Compute β in Theorem 1. Update $\tilde{\mathbf{V}}^{(t+1)}$, $\mathbf{\Lambda}^{(t+1)}$, and $\mathbf{D}^{(t+1)}$ as
$$\tilde{\mathbf{V}}^{(t+1)} = \mathbf{L}_{:k}, \quad \mathbf{\Lambda}^{(t+1)} = \tfrac{1}{\beta}((\alpha_c - \beta)\mathbf{I} + \mathbf{S}_{:k}^2),$$
$$[\mathbf{D}^{(t+1)}]_i = [\mathbf{D}^{(t)}]_i \Big(\frac{\alpha_c + \sum_{j=1}^{r}[\mathbf{W}]_{i,j}^2}{1 + \sum_{j=1}^{k}[\mathbf{\Lambda}^{(t+1)}]_j [\tilde{\mathbf{V}}^{(t+1)}]_{i,j}^2}\Big)^{1/2} \ .$$

8. Compute the $2d$th root of the determinant of the new covariance matrix as $\gamma = \exp\big(\frac{1}{d}\sum_{i=1}^{d}\ln([\mathbf{D}^{(t+1)}]_i) + \frac{1}{2d}\sum_{j=1}^{k}\ln(1 + [\mathbf{\Lambda}^{(t+1)}]_j)\big)$ and $\mathbf{D}^{(t+1)} \leftarrow \mathbf{D}^{(t+1)}/\gamma$ and $\mathbf{p}_c^{(t+1)} \leftarrow \mathbf{p}_c^{(t+1)}/\gamma$.

Note that the above algorithm with $k = d - 1$ is equivalent to the original CMA-ES with TPA except the last step where the diagonal matrix $\mathbf{D}$ and the evolution path $\mathbf{p}_c$ are scaled by the same factor so that the determinant of the covariance matrix is fixed to be 1. If $k = 0$, it is equivalent to the sep-CMA-ES with TPA. The proposed algorithm, called *VkD-CMA*, bridges the gap between the sep-CMA-ES and the CMA-ES and controls the richness of the covariance model by $k \in [\![0, d-1]\!]$. The space complexity of VkD-CMA is $\mathcal{O}(dr)$ and the time complexity per f-call is $\mathcal{O}(dr\max(1, r/\lambda))$. For example, if we set $k = \mu = \lfloor \lambda/2 \rfloor$, the time complexity per f-call is $\mathcal{O}(d(k + \mu + 1))$, compared to $\mathcal{O}(d^2)$ in the CMA-ES.

Implementation Remark. Since $\mathbf{\Lambda}^{(0)}$ is the zero matrix, only the first $t(\mu + 1)$ diagonal elements of $\mathbf{\Lambda}^{(t)}$ can be nonzero. Then, in the first few iterations, $k - t(\mu+1)$ columns of $\mathbf{W}$ are zero vectors. To avoid unnecessary numerical errors, it is advised to remove such columns from $\mathbf{W}$ before the thin SVD of $\mathbf{W}$ is computed. Moreover, if some of the diagonal elements of $\mathbf{\Lambda}$ are $\ll 1$ (e.g., $< 10^{-14}$), we drop the corresponding columns of $\tilde{\mathbf{V}}$ and $\mathbf{\Lambda}$ since the effect of such singular values are negligible.

4. EXPERIMENTS

Table 1 summarizes the definitions of the test functions. All the functions except the Rosenbrock function f_{ros} and the rotated Rosenbrock function f_{rosrot} are quadratic and their inverse Hessian matrices are in $\mathcal{M}_1$. The global minimum point is located at $\mathbf{1}$, $\mathbf{Q}^{\mathrm{T}}\mathbf{1}$ and $\mathbf{0}$ for f_{ros}, f_{rosrot} and all the other functions, respectively, and the minimum f-value is zero for all the functions. For each setting, we run the experiments ten times. We produce ten instances of $\mathbf{Q}$ and $\mathbf{u}$ that are used in common for different settings so that we can compare the performance on the same instances.

The initial step-size $\sigma^{(0)} = 2$ and the initial scaling matrix $\mathbf{D}^{(0)} = \mathrm{diag}(1, \dots, 1)$ for all the functions. The initial values for $\mathbf{m}^{(0)}$ is generated from $2\mathcal{N}(\mathbf{0}, \mathbf{I})$ for f_{ros} and f_{rosrot}, from $3\cdot\mathbf{1} + 2\mathcal{N}(\mathbf{0}, \mathbf{I})$ for all the other functions. The default values are used for all the strategy parameters such as the number of samples and the learning rates described in Section 3.3. The value of k is mentioned in each experiment.

We measure the performance of algorithms by the number of function evaluations spent before reaching the target function value $f_{\mathrm{target}} = 10^{-8}$ for all the functions. Each run is considered successfull if the target value is achieved before $5 \times 10^4 d$ function evaluations.

4.1 Effect of k

Any quadratic function and its strictly increasing transformation, i.e., $f(x) = g(\frac{1}{2}x^{\mathrm{T}}\mathbf{H}x)$ with a positive definite symmetric Hessian $\mathbf{H}$ and a strictly increasing function $g : \mathbb{R} \to \mathbb{R}$, can be identified with the Sphere function $f(x) = \|x\|^2$ in the CMA-ES if the covariance matrix is fixed to $\mathbf{C} \propto \mathbf{H}^{-1}$. Therefore, if the inverse Hessian $\mathbf{H}^{-1} \in \mathcal{M}_k$ and is well approximated, i.e. $\mathrm{Cond}(\mathbf{CH}) \lessapprox 10$, by the restricted covariance matrix adaptation in VkD-CMA, we expect that the VkD-CMA can also solve the function efficiently.

To check if the proposed adaptation of the restricted covariance matrix can learn the inverse Hessian in $\mathcal{M}_k$, we use the following quadratic test function

$$f(x) = x^{\mathrm{T}}\mathbf{D}_{\mathrm{ell}}(10^6\mathbf{I} - (10^6 - 1)\mathbf{U}\mathbf{U}^{\mathrm{T}})\mathbf{D}_{\mathrm{ell}}x \ , \qquad (19)$$

Table 1: Benchmark function suite. The orthogonal matrix $\mathbf{Q}$ is constructed as follows. First all the elements are generated from the standard normal distribution and apply Gram-Schmidt procedure to orthonormalize its columns. The unit vector $\mathbf{u}$ is generated from the standard normal distribution and divided by its norm. The ith diagonal element of the diagonal matrix $\mathbf{D}_{\mathrm{ell}}$ is $10^{3\frac{i-1}{d-1}}$.

Definitions	
$f_{\mathrm{sph}}(x)$	$\sum_{i=1}^{d} x_i^2$
$f_{\mathrm{cig}}(x)$	$x_1^2 + 10^6 \sum_{i=2}^{d} x_i^2$
$f_{\mathrm{cigrot}}(x)$	$10^6 f_{\mathrm{sph}}(x) + (1 - 10^6)\langle x, \mathbf{u}\rangle^2$
$f_{\mathrm{dis}}(x)$	$10^6 x_1^2 + \sum_{i=2}^{d} x_i^2$
$f_{\mathrm{ell}}(x)$	$f_{\mathrm{sph}}(\mathbf{D}_{\mathrm{ell}} x)$
$f_{\mathrm{ellcig}}(x)$	$f_{\mathrm{cigrot}}(\mathbf{D}_{\mathrm{ell}} x)$
$f_{\mathrm{ros}}(x)$	$\sum_{i=1}^{d-1} 10^2 (x_i^2 - x_{i+1})^2 + (x_i - 1)^2$
$f_{\mathrm{rosrot}}(x)$	$f_{\mathrm{ros}}(\mathbf{Q} x)$
$f_{\mathrm{twoax}}(x)$	$\sum_{i=1}^{\lfloor d/2 \rfloor} x_i^2 + 10^6 \sum_{i=\lfloor d/2 \rfloor}^{d} x_i^2$

where $\mathbf{U}$ is a $d \times k_{\mathrm{cig}}$ matrix whose columns are orthogonal to each other and of length one. If $k_{\mathrm{cig}} = 0$ or $k_{\mathrm{cig}} = 1$, it is equivalent to $10^6 \times f_{\mathrm{ell}}$ and f_{ellcig}, respectively. Its inverse Hessian is written in the form $\mathbf{D}(\mathbf{I} + \mathbf{V}\mathbf{V}^{\mathrm{T}})\mathbf{D}$, where $\mathbf{D} = (2 \cdot 10^6)^{-1/2} \mathbf{D}_{\mathrm{ell}}^{-1}$ and $\mathbf{V} = (10^6 - 1)^{1/2}\mathbf{U}$, and is in $\mathcal{M}_{k_{\mathrm{cig}}}$. If k for the VkD-CMA is no less than k_{cig}, the inverse Hessian is included in $\mathcal{M}_k$ and we expect that the VkD-CMA efficiently solves the function (19). In this experiment, the dimension is $d = 100$ and $\sigma^{(0)}$, $\mathbf{D}^{(0)}$, and $\mathbf{m}^{(0)}$ are initialized in the same way as for f_{ellcig}.

Figure 1 shows the average and standard deviation of the number of function evaluations till the target function value $f_{\mathrm{target}} = 10^{-8}$ is reached over ten independent runs, where $\mathbf{U}$ is randomly generated with the same procedure as $\mathbf{Q}$ in Table 1 and is generated independently for each run. It shows that the function is efficiently optimized if $k_{\mathrm{cig}} \leqslant k$ and the target value is not reached within the given budget of f-calls if $k_{\mathrm{cig}} > k$. The result indicates that the inverse Hessian $\mathbf{H}^{-1} \in \mathcal{M}_{k_{\mathrm{cig}}}$ can be well approximated by the proposed procedure with $k \geqslant k_{\mathrm{cig}}$ and solves the function efficiently, whereas if $k < k_{\mathrm{cig}}$, since $\min_{\mathbf{C} \in \mathcal{M}_k} \mathrm{Cond}(\mathbf{C}\mathbf{H}) \gg 1$, the VkD-CMA is not capable of approximating the inverse Hessian and the convergence is very slow. On the other hand, the greater the value of k is, the more function evaluations are required to reach the same target value, which is due to the smaller learning rates designed in (16).

4.2 Comparison with Other Variants

Figure 2 shows the function evaluations to reach the target value averaged over the successful runs divided by the success probability and the dimension. A horizontal line implies that the number of function evaluations spent by an algorithm grows up linearly in dimension. Six methods are performed: VkD-CMA with $k = 1$ and $k = \mu$, the CMA-ES with TPA (Section 2.1), the CMA-ES with CSA [6], the sep-CMA-ES [13], the VD-CMA-ES [1]. The parameters for the last three algorithms are taken from the references. Due to the time complexity, we conduct the simulation up to 200 dimension for the CMA-ES, whose time complexity is quadratic in dimension, and we do the experiments up to 1000 dimension for the other algorithms, whose time complexities are linear in dimension.

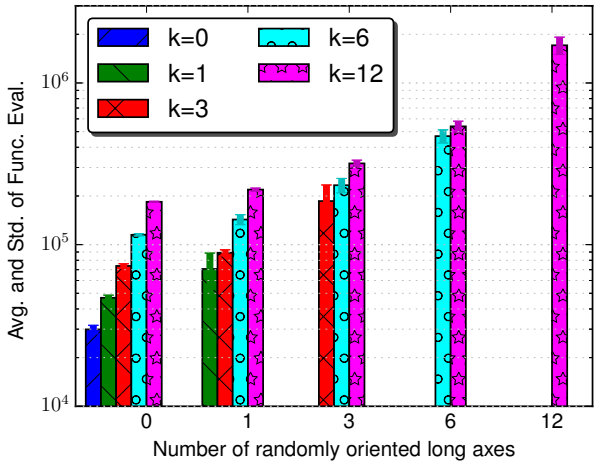


Figure 1: Average and standard deviation of the number of function evaluations spent before the target value $f_{\mathrm{target}} = 10^{-8}$ is reached. Each bar represents the average number of function evaluations over 10 runs for each $k = 0, 1, 3, 6, 12$ in the VkD-CMA and the error bar represents the standard deviation. The number of randomly oriented vectors is $k_{\mathrm{cig}} = 0, 1, 3, 6, 12$. Missing data implies f_{target} is not reached within $5 \times 10^4 d$ function evaluations.

On f_{sph}, we do not observe much difference in performance between the variants of the CMA-ES. This is because the initial covariance matrix $\mathbf{C}^{(0)} = \mathbf{I}$ is proportional to the inverse Hessian and the adaptation is not required. The main search component is the σ adaptation. Since TPA tends to lead to slightly faster convergence [3], VkD-CMA and the CMA-ES with TPA are slightly more efficient than the other variants using CSA.

On f_{cig} and f_{cigrot}, we do not observe much speed up of variants with restricted covariance matrix models over the CMA-ES. These functions inverse Hessian matrices are of the form $\mathbf{I} + \mathbf{v}\mathbf{v}^{\mathrm{T}}$, where $\mathbf{v} \in \mathbb{R}^d$. The rank-one update of the covariance matrix is known to excel at learning such a covariance matrix (the same reported in [1]). Note that the sep-CMA-ES can not solve f_{cigrot} efficiently since its covariance matrix model is $\mathcal{M}_0$ whereas the inverse Hessian of f_{cigrot} is in $\mathcal{M}_1 \setminus \mathcal{M}_0$.

On f_{discus}, f_{ell}, and f_{ellcig}, we observe relatively great difference between the variants with restricted covariance matrix models and the CMA-ES. That is because the variants with restricted covariance matrix models use greater values for the learning rates c_μ and c_1, which makes the adaptation time for the covariance matrix shorter. We also observe the difference between VkD-CMA with $k = 1$ and $k = \mu$ due to the different learning rates.

On f_{ros} and f_{rosrot}, we observe less differences between the variants except that the CMA-ES scales up slightly worse than the others. That is because the most of the function evaluations are spent to move along a long curved ridge while the covariance matrix adaptation is less critical since the condition number of the Hessian during the search is not very high (≈ 100). On a curved ridge structure such as f_{ros}

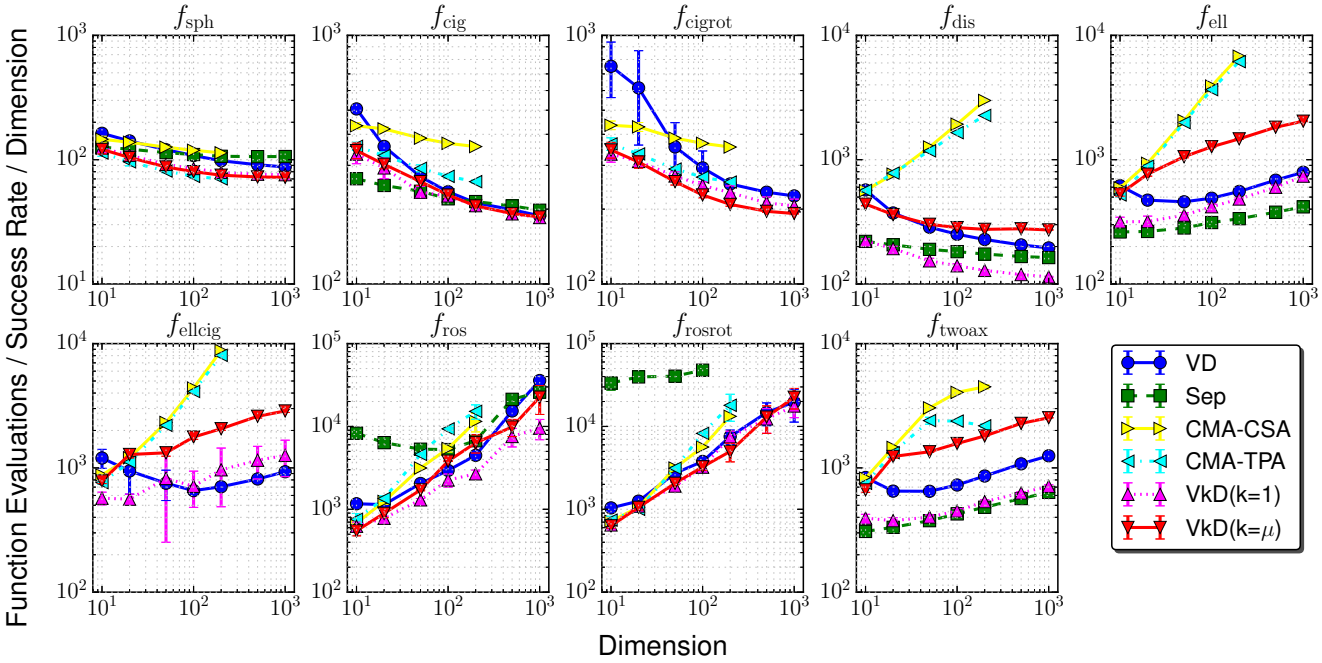


Figure 2: Function evaluations to reach the target value averaged over the successful runs divided by the success probability and the dimension. Each error bar shows the standard deviation.

and f_{rosrot}, $\mathcal{M}_1$ seems to be sufficient to approximate the inverse Hessian, while it is not exactly in $\mathcal{M}_1$.

On f_{twoax}, we observe a greater difference between VkD-CMA with $k = 1$ and $k = \mu$ than the difference on the other functions. Moreover, the difference between the CMA-ES with TPA and the VkD-CMA with $k = \mu$ is getting smaller for $d \geqslant 50$ and they spend more or less the same function evaluations on $d = 200$ in spite of a big difference in learning rates. This will be discussed in the next section.

4.3 On TwoAxes Function

Figure 3 shows single run results for VkD-CMA with $k = 0, 7, 24, 25, 49$ on 50 dimensional f_{TwoAx}. The Hessian $\mathbf{H}^{-1}$ of the function is diagonal; the first $d/2$ diagonal elements (i.e., eigenvalues) are 2 and the others are $2 \cdot 10^6$. Whatever the value of k is, the inverse Hessian can be expressed as $\mathbf{D} = \mathbf{H}^{-1/2}$ and $\mathbf{V} = \mathbf{0}$. However, if $k \geqslant d/2$, the following non-trivial expression is possible: $\mathbf{D} = \mathrm{diag}(d_1 \mathbf{I}_{d/2}, d_2 \mathbf{I}_{d/2})$, $\mathbf{\Lambda} = \mathrm{diag}((10^6 (d_2/d_1)^2 - 1)\mathbf{I}_{d/2}, \mathbf{0})$, and the first $d/2$ columns of $\tilde{\mathbf{V}}$ span the subspace of the first $d/2$ coordinates of the search space.

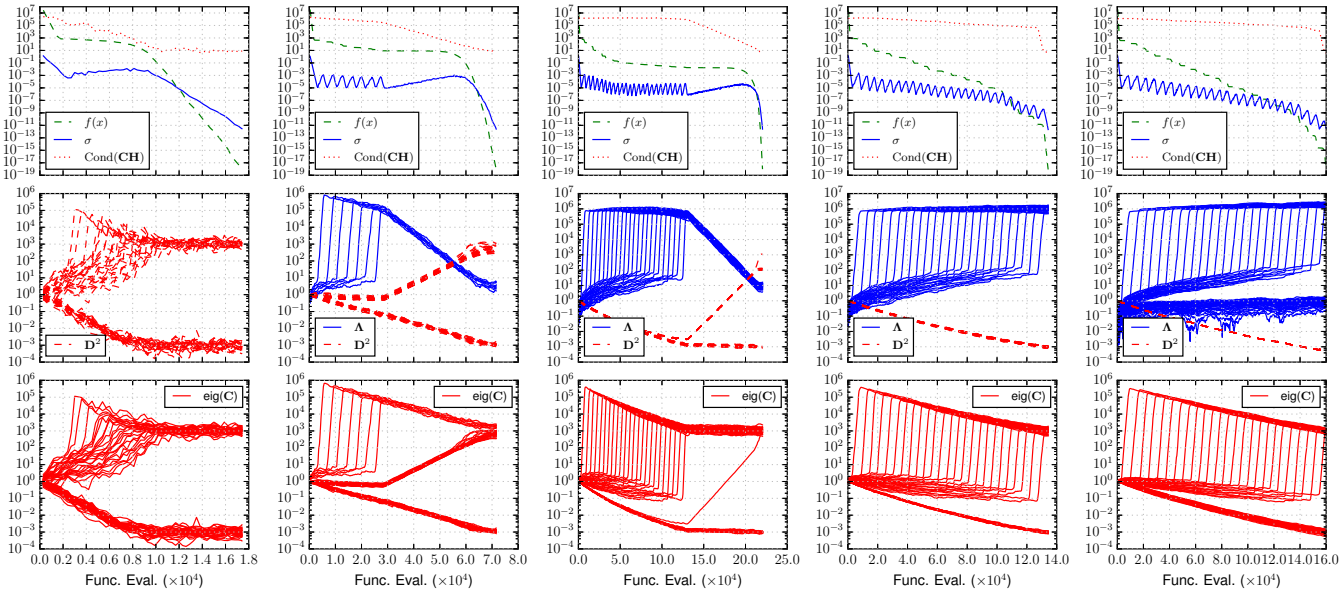
With $k = 0$, VkD-CMA attains the trivial expression of the inverse Hessian. With $k \geqslant 25$, it tends to learn non-trivial expressions of the inverse Hessian while the diagonal matrix is kept proportional to the identity. If $k = 7$ and $k = 24$, it first tries to approximate the inverse Hessian by $\mathbf{V}$. However, to attain a non-trivial expression of the inverse Hessian, $\mathbf{V}$ needs to be either such that $\tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathrm{T}}$ is approximately diagonal or that $\mathbf{\Lambda}$ is small enough. Once all the vectors become long, i.e., all the elements of $\mathbf{\Lambda}$ are around 10^6, VkD-CMA shortens all the vectors and makes the diagonal elements proportional to the inverse Hessian. Finally, the inverse Hessian is approximated by the trivial expression. Therefore, if k is smaller than the dimension of the subspace corresponding to the greater eigenvalues of the inverse Hessian ($\lfloor d/2 \rfloor$ in this case), it looses the function evaluations to learn $\mathbf{V}$ uselessly.

The same problem can essentially happen when the inverse Hessian of a function has an eigenvalue with the geometric multiplicity (i.e., the dimension of the eigenspace associated with the eigenvalue) greater than one.

5. CONCLUSION

In this paper we have proposed a computationally efficient variant of the CMA-ES with a restricted covariance matrix model $\mathbf{C} = \mathbf{D}(\mathbf{I} + \mathbf{V}\mathbf{V}^{\mathrm{T}})\mathbf{D} \in \mathcal{M}_k$, namely VkD-CMA. We have shown that the VkD-CMA is equivalent to the sep-CMA-ES or the original CMA-ES when $k = 0$ or $k = d - 1$, respectively, and that the internal complexity per f-call is $\mathcal{O}(dr \max(1, r/\lambda))$—where $r = k + \mu + 1$, λ and μ are the number of samples and selected samples per iteration, and d is the dimension—compared to $\mathcal{O}(d^2)$ for the original CMA-ES. Experimental results have shown that the VkD-CMA have an advantage in terms of the number of function evaluations additionally to the internal computational complexity, mainly because of larger learning rates than the ones used in the CMA-ES. The richer the covariance matrix model (the larger the value of k) is, the more functions can be efficiently solvable while the more function evaluations are required to adapt the covariance matrix. Meanwhile, the results also reveal the shortcomings of the VkD-CMA on a function such as f_{TwoAx} described in Section 4.3.

There are two main future work to be done. One is to adapt the model complexity, k, online. Currently, we need to set k in advance relying on a limited priori knowledge of the problem. If we can adapt k online, we do not need to tune k in advance. Moreover, we may be able to keep k as small as possible, resulting in a fast adaptation of the



(a) $k = 0$ (b) $k = \mu = 7$ (c) $k = 24$ (d) $k = 25$ (e) $k = 49$

Figure 3: Single run results of VkD-CMA with $k = 0, 7, 24, 25, 49$ on f_{TwoAx} in 50 dimension. The best function value at each iteration, the step-size σ (on the top), the condition number Cond($\mathbf{CH}$), each element of $\mathbf{D}$ and $\mathbf{\Lambda}$ (on the middle), each eigenvalue of $\mathbf{C} = \mathbf{D}(\mathbf{I} + \mathbf{V}\mathbf{V}^{\mathrm{T}})\mathbf{D}$ (on the bottom) are shown.

covariance matrix. This may also help to prevent the problem on f_{TwoAx}. The other line is to extend the algorithm so that it can adapt a short direction of the distribution, i.e., a small eigenvalue of the inverse Hessian. The eigenvalues of $\mathbf{I} + \mathbf{V}\mathbf{V}^{\mathrm{T}}$ are not smaller than one, which prevents the algorithm from efficiently working on a rotated f_{dis} function.

Acknowledgements. This work was initiated by a discussion during the Dagstuhl Seminar 15211, and is partially supported by JSPS KAKENHI Grant Number 15K16063.

6. REFERENCES

[1] Y. Akimoto, A. Auger, and N. Hansen. Comparison-based natural gradient optimization in high dimension. In *Proceedings of Genetic and Evolutionary Computation Conference*, pages 373–380, 2014.

[2] Y. Akimoto, Y. Nagata, I. Ono, and S. Kobayashi. Theoretical Foundation for CMA-ES from Information Geometry Perspective. *Algorithmica*, 64:698–716, 2012.

[3] A. Atamna. Benchmarking IPOP-CMA-ES-TPA and IPOP-CMA-ES-MSR on the BBOB noiseless testbed. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 1135–1142, 2015.

[4] N. Hansen. The CMA Evolution Strategy: A Comparing Review. In J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, editors, *Towards a new evolutionary computation*, pages 75–102. Springer, 2006.

[5] N. Hansen, A. Atamna, and A. Auger. How to assess step-size adaptation mechanisms in randomised search. In *Parallel Problem Solving from Nature–PPSN XIII*, pages 60–69. 2014.

[6] N. Hansen and A. Auger. Principled Design of Continuous Stochastic Search: From Theory to Practice. In Y. Borenstein and A. Moraglio, editors, *Theory and Principled Methods for the Design of Metaheuristics*. Springer, 2014.

[7] N. Hansen, S. D. Muller, 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.

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

[9] I. Loshchilov. A computationally efficient limited memory cma-es for large scale optimization. In *Proceedings of Genetic and Evolutionary Computation Conference*, pages 397–404, 2014.

[10] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11(1):50–59, 1960.

[11] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, second edition, 2006.

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

[13] R. Ros and N. Hansen. A simple modification in CMA-ES achieving linear time and space complexity. *Parallel Problem Solving from Nature–PPSN X*, pages 296–305, 2008.