Online Model Selection for Restricted Covariance Matrix Adaptation

Youhei Akimoto[1(✉)] and Nikolaus Hansen[2]

[1] Faculty of Engineering, Shinshu University, Nagano, Japan
y_akimoto@shinshu-u.ac.jp
[2] Inria, Research Centre Saclay – Île-de-France, Palaiseau, France
nikolaus.hansen@lri.fr

Abstract. We focus on a variant of covariance matrix adaptation evolution strategy (CMA-ES) with a restricted covariance matrix model, namely VkD-CMA, which is aimed at reducing the internal time complexity and the adaptation time in terms of function evaluations. We tackle the shortage of the VkD-CMA—the model of the restricted covariance matrices needs to be selected beforehand. We propose a novel mechanism to adapt the model online in the VkD-CMA. It eliminates the need for advance model selection and leads to a performance competitive with or even better than the algorithm with a nearly optimal but fixed model.

1 Introduction

The covariance matrix adaptation evolution strategy (CMA-ES) [6] is a stochastic search algorithm for continuous optimization. It is considered a state-of-the-art algorithm for black-box scenarios. In the CMA-ES, candidate solutions are generated from a normal (Gaussian) distribution $\mathcal{N}(\mathbf{m}, \sigma^2\mathbf{C})$ with mean vector $\mathbf{m}$, step-size σ, and covariance matrix $\mathbf{C}$. Thanks to the adaptation of positive definite symmetric covariance matrix $\mathbf{C}$, the CMA-ES is known as an efficient optimizer for ill-conditioned and non-separable functions. On a quadratic function, it is empirically known [6] and theoretically supported [1] that the covariance matrix approximates the inverse Hessian, which turns the problem into a spherical function.

In the references [2,3,10,11], variants of CMA-ES with a restricted covariance matrix model are proposed. All of these approaches have common advantages and disadvantages over the standard CMA-ES. The advantages are mainly twofold. One is the internal complexity. As the covariance matrix is represented with a fewer number of parameters, its space complexity is improved. Moreover, computationally efficient update formulas for these restricted covariance matrices are employed, leading to an improvement in the internal time complexity. Therefore, they are promising when solving an optimization problem in a high dimension. The other advantage is the speedup in terms of the number of function evaluations required to adapt the covariance matrix. Since they have fewer parameters to be adapted, the update in one iteration is more reliable, allowing a

J. Handl et al. (Eds.): PPSN XIV 2016, LNCS 9921, pp. 3–13, 2016.
DOI: 10.1007/978-3-319-45823-6_1

greater learning rate. The disadvantage is that the restricted covariance matrix may not be rich enough to approximate the inverse Hessian of the objective function. If this is the case, the convergence rate will be very low. The mean vector will not approach to the optimum within a reasonable run-time.

The VkD-CMA [3] is a variant of the CMA-ES with a restricted covariance matrix mode. It parameterizes the covariance matrix with a diagonal matrix $\mathbf{D}$ and k vectors $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$, i.e., $\mathbf{C} = \mathbf{D}(\mathbf{I} + \mathbf{V}\mathbf{V}^{\mathrm{T}})\mathbf{D}$. It is proved that the algorithm is equivalent to sep-CMA-ES [11] if $k = 0$ and is equivalent to CMA-ES if $k = d - 1$. Therefore, the VkD-CMA is considered as a generalization of these variants of the CMA-ES and allows us to control the model complexity by tuning the number of vectors, i.e., k, between diagonal and positive definite symmetric. However, k needs to be tuned in advance to exploit the structure of a function. Without a strong prior knowledge, it can be prohibitively expensive.

In this paper, we propose an online adaptation of the model complexity for the restricted covariance model used in the VkD-CMA, i.e., online k adaptation. The idea to increase k is to detect the condition that we observe when the covariance matrix model is not rich enough to approximate the inverse Hessian. The idea to decrease k is to check if the current covariance matrix is well approximated with a smaller k. We expect two advantages of the online k adaptation. First, it obviates the need for tuning of k, leading to a speedup in the preprocessing of optimization and turning the algorithm more user-friendly. Second, online adaptation of the model complexity may lead to a faster adaptation of the covariance matrix than the optimal but fixed k.

The rest of the paper is organized as follows. Section 2 is devoted to the introduction to the VkD-CMA. The proposed k adaptation mechanism is presented in Sect. 3. We conduct experiments in Sect. 4 to check how efficiently the proposed mechanism adapts k and to compare with variants of CMA-ES. In Sect. 5 we summarize our contributions and discuss a possible line of future work.

2 VkD-CMA

The VkD-CMA [3] is a variant of the covariance matrix adaptation evolution strategy (CMA-ES) [6–8] with a restricted covariance matrix model. As well as the other variants of CMA-ES, multiple candidate solutions are sampled from the multivariate normal distribution $\mathcal{N}(\mathbf{m}, \sigma^2\mathbf{C})$, they are evaluated on the objective function $f : \mathbb{R}^d \to \mathbb{R}$, and the distribution parameters, $\mathbf{m}$, σ, and $\mathbf{C}$, are updated using the candidate solutions and their fitness ranking. In the VkD-CMA, the covariance matrix $\mathbf{C}$ is parameterized with a d dimensional positive-definite diagonal matrix $\mathbf{D}$ and k orthogonal vectors $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$, the latter of which is decomposed into a k dimensional nonnegative definite diagonal matrix $\mathbf{\Lambda}$ and a $d \times k$ dimensional matrix $\tilde{\mathbf{V}}$ with orthogonal columns of unit length. Then,

$$\mathbf{C} = \mathbf{D}(\mathbf{I} + \mathbf{V}\mathbf{V}^{\mathrm{T}})\mathbf{D}, \quad \text{or equivalently,} \quad \mathbf{C} = \mathbf{D}(\mathbf{I} + \tilde{\mathbf{V}}\mathbf{\Lambda}\tilde{\mathbf{V}}^{\mathrm{T}})\mathbf{D}. \tag{1}$$

The parameter, k, determines the richness of the covariance matrix mode. Let $\mathcal{M}_k$ be the set of matrices in the form (1). The set $\mathcal{M}_0$ with $k = 0$ is the set of

diagonal matrices and $\mathcal{M}_{d-1}$ with $k = d-1$ is the set of arbitrary positive-definite symmetric matrices. The covariance matrix adaptation, i.e., update of $\mathbf{D}$, $\mathbf{\Lambda}$, and $\tilde{\mathbf{V}}$, is based on the projection of the covariance matrix from $\mathcal{M}_{d-1}$ onto its subset $\mathcal{M}_k$. The algorithm employing the two-point step-size adaptation (TPA) [5] is described below, followed by the description of the parameters appearing in the algorithm and their default values.

Algorithm. We initialize $\mathbf{m}^{(0)}$, $\sigma^{(0)}$ and $\mathbf{D}^{(0)}$ according to the initial search interval of a given problem. Let $\tilde{\mathbf{V}}^{(0)} = \mathbf{0}$, $\mathbf{\Lambda}^{(0)} = \mathrm{diag}(0, \ldots, 0)$, $\mathbf{p}_c^{(0)} = \mathbf{0}$, and $s^{(0)} = 0$. 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

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

where the Mahalanobis norm $\|\mathbf{dm}^{(t-1)}\|^2_{\mathbf{C}^{(t)}} = (\mathbf{dm}^{(t-1)})^{\mathrm{T}}(\mathbf{C}^{(t)})^{-1}\mathbf{dm}^{(t-1)}$ is computed with the following formula: Let $u_1 = \mathbf{D}^{-1}\mathbf{dm}$ and $u_2 = \tilde{\mathbf{V}}^{\mathrm{T}}u_1$, then

$$(\mathbf{dm})^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{dm} = \|u_1\|^2 + u_2^{\mathrm{T}}((\mathbf{I} + \mathbf{\Lambda})^{-1} - \mathbf{I})u_2. \tag{3}$$

Let $y_1 = y_+$, $y_2 = y_-$. If $t = 0$, generate y_1 and y_2 in the same way as the next step.

2. Sample $\lambda - 2$ independent random vectors $z_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, for $i = 3, \ldots, \lambda$, and compute y_i according to

$$y_i \leftarrow \tilde{\mathbf{V}}^{\mathrm{T}}z_i, \quad y_i \leftarrow ((\mathbf{\Lambda} + \mathbf{I})^{1/2} - \mathbf{I})y_i, \text{ and } y_i \leftarrow \mathbf{D}(z_i + \tilde{\mathbf{V}}y_i). \tag{4}$$

Let $x_i = \mathbf{m}^{(t)} + \sigma^{(t)}y_i$ for $i = 1, \ldots, \lambda$.

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 $y_{i:\lambda}$ and update the mean vector $\mathbf{m}^{(t+1)}$ according to

$$\mathbf{dm}^{(t)} = \textstyle\sum_{i=1}^{\mu} w_i y_{i:\lambda} \ , \quad \mathbf{m}^{(t+1)} = \mathbf{m}^{(t)} + c_m\sigma^{(t)}\mathbf{dm}^{(t)}. \tag{5}$$

5. If $t \geqslant 1$, we update $s^{(t+1)}$ and update the step-size according to

$$s^{(t+1)} = (1 - c_\sigma)s^{(t)} + c_\sigma(\mathrm{rank}(x_2) - \mathrm{rank}(x_1))/(\lambda - 1) \ , \tag{6}$$

$$\sigma^{(t+1)} = \sigma^{(t)} \exp(s^{(t+1)}/d_\sigma). \tag{7}$$

Let $h_\sigma = \mathbb{I}\{s^{(t+1)} < 0.5\}$. Otherwise, let $s^{(1)} = s^{(0)}$, $\sigma^{(1)} = \sigma^{(0)}$ and $h_\sigma = 1$.

6. Update the evolution path

$$\mathbf{p}_c^{(t+1)} = (1 - c_c)\mathbf{p}_c^{(t)} + h_\sigma(c_c(2 - c_c)\mu_{\mathrm{eff}})^{1/2}\mathbf{dm}^{(t)}. \tag{8}$$

7. Let $\mathbf{W} = [\alpha_c^{1/2}\tilde{\mathbf{V}}^{(t)}(\mathbf{\Lambda}^{(t)})^{1/2}, \ (\mathbf{D}^{(t)})^{-1}\mathbf{Y}]$, where $\alpha_c = 1 - c_\mu - c_1 + (1 - h_\sigma)c_1c_c(2-c_c)$ and $\mathbf{Y}$ is a $d \times (\mu+1)$ dimensional matrix whose first μ columns are

given by $(c_\mu w_i)^{1/2} y_{i:\lambda}$ for $i = 1, \ldots, \mu$ and the last column is $c_1^{1/2} \mathbf{p}_c^{(t+1)}$. Let $r = \min(k + \mu + 1, d)$. Compute the thin singular value decomposition (SVD) of $\mathbf{W}$, denoted by $\mathbf{LSR}^{\mathrm{T}}$, where $\mathbf{S}$ is a $r \times r$ dimensional diagonal matrix whose diagonal elements are the singular values of $\mathbf{W}$ aligned in descending order, $\mathbf{L}$ and $\mathbf{R}$ are matrices of dimension $d \times r$ and $r \times r$, respectively, whose columns are the left and right singular vectors with unit length. Compute $\beta = \alpha_c + (d-k)^{-1} \sum_{i=k+1}^{r} [\mathbf{S}]_{i,i}^2$ and update $\tilde{\mathbf{V}}^{(t+1)}$, $\mathbf{\Lambda}^{(t+1)}$, and $\mathbf{D}^{(t+1)}$ as

$$\tilde{\mathbf{V}}^{(t+1)} = \mathbf{L}_{:,:k} \quad , \quad \mathbf{\Lambda}^{(t+1)} = \frac{(\alpha_c - \beta)\mathbf{I} + \mathbf{S}^2_{:k,:k}}{\beta} \quad , \quad [\mathbf{D}^{(t+1)}]_{i,i} = \frac{[\mathbf{D}^{(t)}]_{i,i} (\alpha_c + \sum_{j=1}^{r} [\mathbf{W}]_{i,j}^2)^{1/2}}{(1 + \sum_{j=1}^{k} [\mathbf{\Lambda}^{(t+1)}]_{j,j} [\tilde{\mathbf{V}}^{(t+1)}]_{i,j}^2)^{1/2}}. \tag{9}$$

Here, for any matrix $\mathbf{A}$, $[\mathbf{A}]_{i,j}$ denotes the (i,j)th element of $\mathbf{A}$ and $\mathbf{A}_{:i,:i}$ denotes the $i \times i$ upper left block of $\mathbf{A}$, $\mathbf{A}_{:,:i}$ denotes the first i columns of $\mathbf{A}$.

8. Compute the $2d$th root of the determinant of the new covariance matrix as $\gamma = \exp\left(\frac{1}{d} \sum_{i=1}^{d} \ln([\mathbf{D}^{(t+1)}]_{i,i}) + \frac{1}{2d} \sum_{j=1}^{k} \ln(1 + [\mathbf{\Lambda}^{(t+1)}]_{j,j})\right)$ and $\mathbf{D}^{(t+1)} \leftarrow \mathbf{D}^{(t+1)}/\gamma$ and $\mathbf{p}_c^{(t+1)} \leftarrow \mathbf{p}_c^{(t+1)}/\gamma$. Then, we have $\det(\mathbf{C}^{(t+1)}) = 1$.

Default Parameter Values. The default parameter values are summarized as follows. The population size $\lambda = \lfloor 4 + 3\ln(d) \rfloor$, the number of parents $\mu = \lfloor \lambda/2 \rfloor$, and the weights

$$w_i = \frac{\ln((\lambda+1)/2) - \ln(i)}{\sum_{i=1}^{\mu} (\ln((\lambda+1)/2) - \ln(i))} \quad (i = 1, \ldots, \mu) \ , \qquad w_i = 0 \quad (i > \mu). \tag{10}$$

Let $\mu_{\mathrm{eff}} = 1/(\sum_{i=1}^{\mu} w_i^2)$. The learning rate for $\mathbf{m}$-update $c_m = 1$, the learning rate for s in TPA $c_\sigma = 0.3$, the damping parameter for TPA $d_\sigma = d^{1/2}$. The cumulation factor c_c, the learning rate for the rank-one update c_1 and the learning rate for the rank-μ update c_μ are as follows[1] (letting $a \wedge b = \min(a, b)$ for any real a and b)

$$c_c = \frac{4 + \mu_{\mathrm{eff}}/d}{(d+2(k+1))/3 + 4 + 2\mu_{\mathrm{eff}}/d}, \ c_1 = \frac{2}{d(k+2) + 2(k+2) + \mu_{\mathrm{eff}}}, \ c_\mu = (1 - c_1) \wedge \frac{2(\mu_{\mathrm{eff}} - 2 + 1/\mu_{\mathrm{eff}})}{d(k+1) + 4(k+2) + \mu_{\mathrm{eff}}}. \tag{11}$$

Properties. With the restricted covariance matrix model, we achieve cheaper computational time and space complexity and faster adaptation of the covariance matrix than the CMA-ES. Its space complexity is $\mathcal{O}(dr)$, where $r = \min(d, k + \mu + 1)$, and its time complexity is $\mathcal{O}(dr^2 + dk\lambda)$ per iteration. If $r \ll d$, it is cheaper than the CMA-ES, which requires $\Theta(d^2 + d\mu)$ space and $\Theta(d^2\lambda)$ time complexity per iteration. Moreover, since there are fewer parameters to be adapted if k is smaller, it accepts relatively higher learning rates (11) than the default values used in the CMA-ES, resulting in faster adaptation of the covariance matrix. Therefore, we want to keep k as small as possible.

[1] The default c_1 is slightly different from the original setting in [3]. The value presented in the paper is slightly more stable for k close to zero.

On the other hand, if k is too small to approximate the inverse Hessian of the objective function, the VkD-CMA is not able to solve the problem efficiently. Empirically, we know that the convergence rate, defined as the slope of the step-size in log-scale, is proportional to $\mathrm{Cond}(\mathbf{AC})$ on a quadratic objective function $f(x) = x^{\mathrm{T}}\mathbf{A}x$. Therefore, we need to set k large enough to approximate the inverse Hessian of the objective function. However, since a prior knowledge on the problem is limited in the black-box scenario, it is hard to choose a reasonable k in advance. In the next section, we propose a mechanism to adapt k, i.e., the model richness of the covariance matrix, during the optimization process.

3 Adaptive Covariance Model Selection

Ideas. Let us consider solving a quadratic function with a positive definite symmetric Hessian $\mathbf{A}$. If $\mathbf{C} \propto \mathbf{A}^{-1}$, the quadratic function is identified with Sphere function $f(x) = \|x\|^2$. In this case, we can deduce that the optimal convergence rate (i.e., the slope of $\ln(\sigma)$) is approximately $0.5\lambda/d$ using the result given in [4][2]. We also empirically know that the convergence rate is approximately proportional to $1/\mathrm{Cond}(\mathbf{CA})$ if $\mathrm{Cond}(\mathbf{CA}) \gg 1$.

Consider the cases where the covariance matrix is richer than sufficient. It means that the inverse Hessian of the objective function can be approximated by the covariance matrix in $\mathcal{M}_k$ with a smaller k. Let us consider that we drop the ith vector $\mathbf{v}_i$, i.e., drop the ith column of $\tilde{\mathbf{V}}$ and ith column and row of $\mathbf{\Lambda}$, and let $\tilde{\mathbf{C}}$ be the resulting covariance matrix. Then, we have that $\mathrm{Cond}(\mathbf{A}\tilde{\mathbf{C}}) \leqslant \mathrm{Cond}(\tilde{\mathbf{C}}\mathbf{C}^{-1})\mathrm{Cond}(\mathbf{AC}) = (1 + [\mathbf{\Lambda}]_{i,i})\mathrm{Cond}(\mathbf{AC})$. It means that by dropping the ith component from the covariance matrix, we may increase the condition number at most by the factor of $1 + [\mathbf{\Lambda}]_{i,i}$. If $1 + [\mathbf{\Lambda}]_{i,i}$ is smaller than a given threshold β_{dec}, it is safe to remove $\mathbf{v}_i$. However, if the covariance matrix is in the middle of adaptation and $[\mathbf{\Lambda}]_{i,i}$ is still increasing, there is a chance that $1 + [\mathbf{\Lambda}]_{i,i}$ will grow up above the threshold. Therefore, we want to decrease k and drop some components from the covariance matrix only when $1 + [\mathbf{\Lambda}]_{i,i}$ is small enough and is regarded as not increasing.

Consider the cases where the covariance matrix is not rich enough. In this case we observe that $\mathbf{C}$ is kept roughly constant and σ converges very slowly. If we observe $\mathbf{C}$ not significantly changing (except the scaling factor), it implies that $\mathbf{C}$ is close to the optimal approximation of the inverse Hessian in the current covariance model. If the covariance model is rich enough, $\mathrm{Cond}(\mathbf{CA})$ will be close to 1 and the convergence rate will not be very small compared with the optimal convergence rate. If the covariance model is not rich enough, $\mathrm{Cond}(\mathbf{CA}) \gg 1$ and we will observe a slow convergence of σ with the convergence rate proportional to $\mathrm{Cond}(\mathbf{CA})$. Based on this observation, we detect insignificant change of $\mathbf{C}$ and slow convergence of σ. If all of $1 + [\mathbf{\Lambda}]_{i,i}$ are greater than a given threshold β_{inc} and both of the conditions are satisfied, we increase k.

[2] If we use only nonnegative weights as we do in this paper, the possible convergence rate halves.

Algorithm. Let $t_{\mathrm{ada}} = 0$ be the number of iterations after the last k increase or the initialization. Initialize the exponential moving average M_* of the slopes of $\ln(\sigma)$, $\ln([\mathbf{C}]_{i,i})$ and $\ln(1 + [\mathbf{\Lambda}]_{i,i})$ by zero, where $*$ is either $\ln(\sigma)$, $\ln([\mathbf{C}]_{i,i})$ or $\ln(1 + [\mathbf{\Lambda}]_{i,i})$. The following steps are performed after Step 8 of the VkD-CMA algorithm.

1. Update the exponential moving averages according to

$$M_{\ln(\sigma)}^{(t+1)} = (1 - \alpha_{\mathrm{exp}}^{(\sigma)}) M_{\ln(\sigma)}^{(t)} + \alpha_{\mathrm{exp}}^{(\sigma)} \big(\ln(\sigma^{(t+1)}) - \ln(\sigma^{(t)}) \big). \tag{12}$$

The exponential moving averages for $\ln([\mathbf{C}]_{i,i})$ and $\ln(1 + [\mathbf{\Lambda}]_{i,i})$ are updated analogously with the decay factor $\alpha_{\mathrm{exp}}^{(\mathbf{C})}$. Note that $\ln([\mathbf{C}]_{i,i}) = 2\ln([\mathbf{D}]_{i,i}) + \ln(1 + \sum_{j=1}^{k} [\mathbf{\Lambda}]_{j,j} [\tilde{\mathbf{V}}]_{i,j}^2)$.

2. If $t_{\mathrm{ada}} > \mathrm{T}_{\mathrm{exp}}$ and $1 + [\mathbf{\Lambda}]_{i,i} > \beta_{\mathrm{inc}}$ for all $i \in [\![1, k]\!]$, we additionally check if (1) $\big|M_{\ln(\sigma)}^{(t+1)}\big| < \gamma_\sigma \min(0.5, 0.5\lambda/d) / \max(1, \beta_{\mathrm{inc}}/10)$ and if (2) $\max_i \big|M_{\ln([\mathbf{C}]_{i,i})}^{(t+1)}\big| < \gamma_{\mathbf{C}}(c_1 + c_\mu)$. If all the above conditions are satisfied, update k as

$$k = \min(\max(\lfloor \kappa_{\mathrm{inc}} k \rfloor, k + 1), k_{\max}) \tag{13}$$

and set $[\tilde{\mathbf{V}}]_{:,i} = (0, \ldots, 0)$ and $[\mathbf{\Lambda}]_{i,i} = 0$ for all $i \in [\![k_{\mathrm{old}} + 1, k]\!]$, where k_{old} is the value for k before updated. Let $t_{\mathrm{ada}} = 0$.

3. If $t_{\mathrm{ada}} > k\mathrm{T}_{\mathrm{exp}}$ and there is at least one index $i \in [\![1, k]\!]$ such that $1 + [\mathbf{\Lambda}]_{i,i} < \beta_{\mathrm{dec}}$, let J be the set of such indices. If there exist indices in J satisfying $M_{\ln(1+[\mathbf{\Lambda}]_{i,i})}^{(t+1)} < 0$, drop ith column from $\tilde{\mathbf{V}}$ and drop ith column and row from $\mathbf{\Lambda}$ for all such indices i and update k accordingly. Then, re-normalize $\mathbf{D}$ and $\mathbf{p}_c$ (perform Step 8 in the VkD-CMA algorithm again).

4. Let $t_{\mathrm{ada}} = t_{\mathrm{ada}} + 1$. If k is updated, update c_1, c_μ, c_c according to (11).

Default Parameter Values. The parameters appearing in the k-adaptation algorithm are described below, together with their default values.

- $\alpha_{\mathrm{exp}}^{(\sigma)} = 0.5 \min(1, \lambda/d) / \max(1, \beta_{\mathrm{inc}}/10)$, $\alpha_{\mathrm{exp}}^{(\mathbf{C})} = 1/d$; The discount rate for the exponential moving average, $0 < \alpha_{\mathrm{exp}} < 1$.
- $\gamma_\sigma = 0.1$, $\gamma_{\mathbf{C}} = 0.3$; The threshold to detect insignificant change of the step-size and the covariance matrix, $\gamma_\sigma > 0$ and $\gamma_{\mathbf{C}} > 0$.
- $\mathrm{T}_{\mathrm{exp}} = \frac{2}{\min(\alpha_{\mathrm{exp}}^{(\sigma)}, \alpha_{\mathrm{exp}}^{(\mathbf{C})})} - 1$; The number of iterations to wait for k adaptation after the last k increase, $\mathrm{T}_{\mathrm{exp}} \geqslant 0$. It is introduced to prevent k from oscillating. The sum of the weights for last $\mathrm{T}_{\mathrm{exp}} + 1$ iterations, i.e., $\alpha_{\mathrm{exp}}, \alpha_{\mathrm{exp}}(1 - \alpha_{\mathrm{exp}}), \ldots, \alpha_{\mathrm{exp}}(1 - \alpha_{\mathrm{exp}})^{\mathrm{T}_{\mathrm{exp}}}$, is $1 - (1 - \alpha_{\mathrm{exp}})^{\mathrm{T}_{\mathrm{exp}}+1} \approx 1 - \exp(-2)$. It implies that the last $\frac{2}{\alpha_{\mathrm{exp}}} - 1$ iterations contain about 86 % of the information in the exponential moving average and the information in the exponential moving average is considered refreshed.
- $k_{\min} = 0$, $k_{\max} = d - 1$, $k_{\mathrm{init}} = k_{\min}$; The minimum, maximum and initial number of vectors, $0 \leqslant k_{\min} \leqslant k_{\mathrm{init}} \leqslant k_{\max} \leqslant d - 1$. The maximum value should be set smaller if the available memory and cpu time are limited.
- $\kappa_{\mathrm{inc}} = 1.414$; The factor for increment of k.

– $\beta_{\mathrm{inc}} = \beta_{\mathrm{dec}} = 30$; This corresponds to the condition number $\mathrm{Cond}(\mathbf{C}^{-1/2}\tilde{\mathbf{C}}\mathbf{C}^{-1/2})$ we accept, where $\mathbf{C}$ and $\tilde{\mathbf{C}}$ are the covariance matrix before and after k-decrease. The greater β_{inc} is, k tends to be smaller.

4 Experiments

We first test the performance of the proposed k-adaptation mechanism on the quadratic function $f(x) = \frac{1}{2}x^{\mathrm{T}}\mathbf{A}x$ with the inverse Hessian

$$\mathbf{A}^{-1} = (10^{-6}/2)\mathbf{D}_{\mathrm{ell}}^{-1}(\mathbf{I} + (10^6 - 1)\mathbf{U}\mathbf{U}^{\mathrm{T}})\mathbf{D}_{\mathrm{ell}}^{-1}, \tag{14}$$

where $\mathbf{D}_{\mathrm{ell}}$ is a diagonal matrix whose ith diagonal component is $10^{3\frac{i-1}{d-1}}$, and $\mathbf{U}$ is a $d \times k_{\mathrm{cig}}$ matrix whose columns are orthogonal to each other and of length one. Ten instances of $\mathbf{U}$ are generated by applying the same procedure to create $\mathbf{R}$ with $m = k_{\mathrm{cig}}$ in Table 1. The inverse Hessian is then in $\mathcal{M}_{k_{\mathrm{cig}}}$. In this experiment, the dimension is $d = 100$ and $\sigma^{(0)} = 2$, $\mathbf{D}^{(0)} = \mathbf{I}$, and $\mathbf{m}^{(0)}$ is generated from $\mathcal{N}(3 \cdot \mathbf{1}, 2^2\mathbf{I})$ for each run. All the other parameters for the VkD-CMA and for the k adaptation mechanism are set to the default values described in this paper.

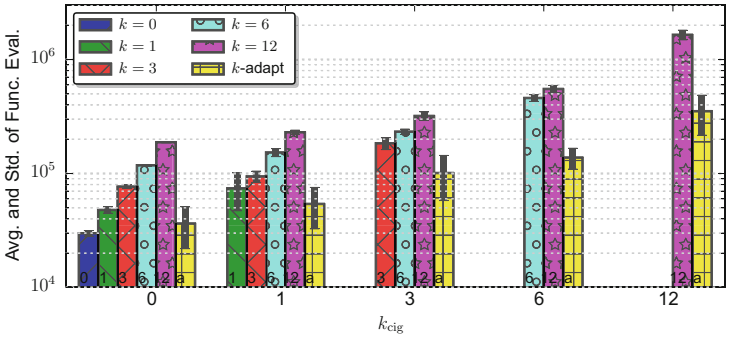

Fig. 1. Average and standard deviation of the number of function evaluations to reach f_{target}. VkD-CMA with fixed $k = 0, 1, 3, 6, 12$ and with the adaptive k are compared on the quadratic with the inverse Hessian (14) with $k_{\mathrm{cig}} = 0, 1, 3, 6, 12$.

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 10 independent runs. If $k < k_{\mathrm{cig}}$, the target value was not reached within 10^7 function evaluations. If $k \geqslant k_{\mathrm{cig}}$, the smaller the value of k is, the smaller the number of function evaluations to reach the target value is. Comparing to the fixed optimal $k = k_{\mathrm{cig}}$, the adaptive strategy requires even fewer function evaluations except for the case $k_{\mathrm{cig}} = 0$. Figure 2 reveals an advantage of the adaptive strategy. The VkD-CMA first adapts $\mathbf{D}$ on this quadratic function, then learns $\tilde{\mathbf{V}}$ and $\mathbf{\Lambda}$.

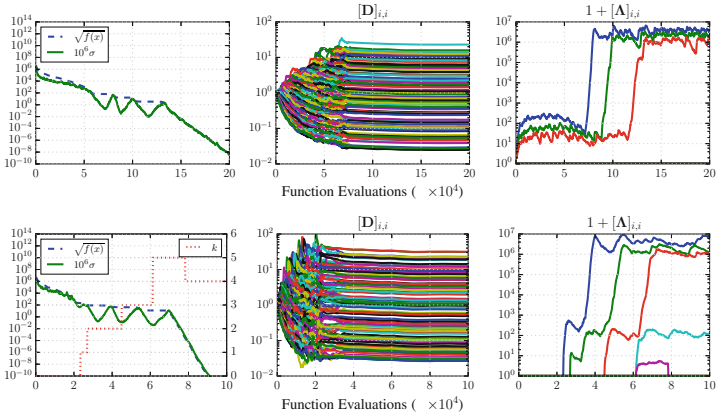

Fig. 2. A typical behavior of the VkD-CMA with fixed optimal $k = k_{\mathrm{cig}}$ (top) and with adaptive k (bottom) on a 100-D quadratic with the inverse Hessian (14) with $k_{\mathrm{cig}} = 3$.

At the beginning, the proposed k adaptation strategy keeps k to zero, resulting in the faster adaptation of $\mathbf{D}$ than the algorithm with the fixed k. Then, it increases k and finally learns the nearly optimal k.

Next, we compare the different variants of CMA-ES, namely, the VkD-CMA with the proposed k adaptation, the VkD-CMA with fixed $k = 1, \mu$, the sep-CMA-ES (SEP) [11], the CMA-ES with cumulative step-size adaptation (CMA-CSA) and the CMA-ES with TPA (CMA-TPA). The parameter values described in Sect. 2 are used for the VkD-CMA. The parameter values for the CMA-CSA and CMA-TPA are taken from the reference [3], and the parameter values for the SEP are taken from [11]. The comparison is done on the test functions summarized in Table 1. For all the functions, the target function value is $f_{\mathrm{target}} = 10^{-8}$. Each run is considered successful if and only if the algorithm evaluates a candidate solution having a better function value than f_{target} before spending $10^5 d$ function evaluations. We conduct ten independent runs for each setting. The initial mean vector and step-size are $\mathbf{m}^{(0)} = 3 \cdot \mathbf{1} + \mathcal{N}(\mathbf{0}, 2^2\mathbf{I})$ and $\sigma^{(0)} = 2$ for all but f_{ros} and f_{rosrot}, where they are initialized as $\mathbf{m}^{(0)} = \mathbf{0} + \mathcal{N}(\mathbf{0}, 0.1^2\mathbf{I})$ and $\sigma^{(0)} = 0.1$.

Figure 3 shows the average number of f-calls. As reported in the references [2,3,11], the sep-CMA-ES and the VkD-CMA with $k = 1$ and $k = \mu$ can solve the functions with inverse Hessian in $\mathcal{M}_0$, $\mathcal{M}_1$, $\mathcal{M}_\mu$, respectively, and can not efficiently solve the functions with highly ill-conditioned inverse Hessian outside them. On f_{sph}, f_{cig}, f_{cigrot}, we do not observe a significant difference between variants compared to the other cases. See [2] for the detail. On the others, a variant with smaller covariance matrix model tends to solve the functions with inverse Hessian inside the model more efficiently.

The VkD-CMA with the proposed k adaptation mechanism succeeded to find the target value within the given budget for all scenarios. Moreover, its efficiency is competitive with or better than the other variants including

Table 1. Benchmark function suite. The d dimensional 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 block diagonal orthogonal matrix $\mathbf{B} = \mathrm{diag}(\mathbf{Q}_1, \mathbf{Q}_2)$ is constructed from two orthogonal matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ of dimension $d/2$ that are generated analogously to $\mathbf{Q}$. The $d \times m$ dimensional matrix $\mathbf{R}$, where $m = \lfloor 2 \ln(d) \rfloor$, is the first m columns of $\mathbf{Q}$. For $f_{\mathrm{ellrotsub}}$, the inverse Hessian does not live in $\mathcal{M}_{\lfloor 2 \ln(d) \rfloor}$, but in its closure.

Name	Definition	(pseudo) inverse Hessian
Sphere	$f_{\mathrm{sph}}(x) = \sum_{i=1}^{d} x_i^2$	$\mathcal{M}_0$
Cigar	$f_{\mathrm{cig}}(x) = x_1^2 + 10^6 \sum_{i=2}^{d} x_i^2$	$\mathcal{M}_0$
Ellipsoid	$f_{\mathrm{ell}}(x) = f_{\mathrm{sph}}(\mathbf{D}_{\mathrm{ell}} x)$	$\mathcal{M}_0$
Discus	$f_{\mathrm{dis}}(x) = 10^6 x_1^2 + \sum_{i=2}^{d} x_i^2$	$\mathcal{M}_0$
TwoAxes	$f_{\mathrm{twoax}}(x) = \sum_{i=1}^{\lfloor d/2 \rfloor} x_i^2 + 10^6 \sum_{i=\lfloor d/2 \rfloor}^{d} x_i^2$	$\mathcal{M}_0$
Ellipsoid-Cigar($k_{\mathrm{cig}} = 1$)	$f_{\mathrm{ellcig}}(x) = f_{\mathrm{cigrot}}(\mathbf{D}_{\mathrm{ell}} x)$	$\mathcal{M}_1$
rotated Cigar	$f_{\mathrm{cigrot}}(x) = f_{\mathrm{cig}}(\mathbf{Q}x)$	$\mathcal{M}_1$
Ellipsoid-Cigar($k_{\mathrm{cig}} = \lfloor \ln(d) \rfloor$)	$f_{\mathrm{ellciglog}}(x) = \frac{1}{2} x^{\mathrm{T}} \mathbf{A} x$ with $\mathbf{A}$ in (14)	$\mathcal{M}_{\lfloor \ln(d) \rfloor}$
subspace rotated Ellipsoid	$f_{\mathrm{ellrotsub}}(x) = f_{\mathrm{ell}}(\mathbf{R}^{\mathrm{T}} x)$	$\mathcal{M}_{\lfloor 2 \ln(d) \rfloor}$ (semi-positive)
rotated TwoAxes	$f_{\mathrm{twoaxrot}}(x) = f_{\mathrm{twoax}}(\mathbf{Q}x)$	$\mathcal{M}_{\lfloor d/2 \rfloor}$
2-blocks rotated Ellipsoid	$f_{\mathrm{ellrot(2\text{-}blocks)}}(x) = f_{\mathrm{ell}}(\mathbf{B}x)$	$\mathcal{M}_{d-1}$
rotated Ellipsoid	$f_{\mathrm{ellrot}}(x) = f_{\mathrm{ell}}(\mathbf{Q}x)$	$\mathcal{M}_{d-1}$
rotated Discus	$f_{\mathrm{disrot}}(x) = f_{\mathrm{disrot}}(\mathbf{Q}x)$	$\mathcal{M}_{d-1}$
Rosenbrock	$f_{\mathrm{ros}}(x) = \sum_{i=1}^{d-1} 10^2 (x_i^2 - x_{i+1})^2 + (x_i - 1)^2$	$\mathcal{M}_{d-1}$ (non-quadratic)
rotated Rosenbrock	$f_{\mathrm{rosrot}}(x) = f_{\mathrm{ros}}(\mathbf{Q}x)$	$\mathcal{M}_{d-1}$ (non-quadratic)

CMA-ES on all but f_{ellrot}, f_{disrot}, and $f_{\mathrm{ellrot(2\text{-}blocks)}}$ functions. To approximate the inverse Hessian of these functions, k needs to be increased nearly to $d - 1$. Then the proposed algorithm spends more function evaluations to adapt the covariance matrix than the CMA-ES does. The function $f_{\mathrm{ellrot(2\text{-}blocks)}}$ has the inverse Hessian in $\mathcal{M}_{d-1} \setminus \mathcal{M}_{d-2}$ but we have $\min_{\mathbf{C} \in \mathcal{M}_0} \mathrm{Cond}(\mathbf{AC}) \leqslant 10^3$, which is relatively small. Even though the convergence speed of the sep-CMA-ES is slower than the CMA-ES due to this condition number, the adaptation time for the covariance matrix for the sep-CMA-ES is shorter and it requires fewer function evaluations to reach the finite target value of $f_{\mathrm{target}} = 10^{-8}$. The proposed algorithm, however, tends to increase k as well as on the fully rotated Ellipsoid function.

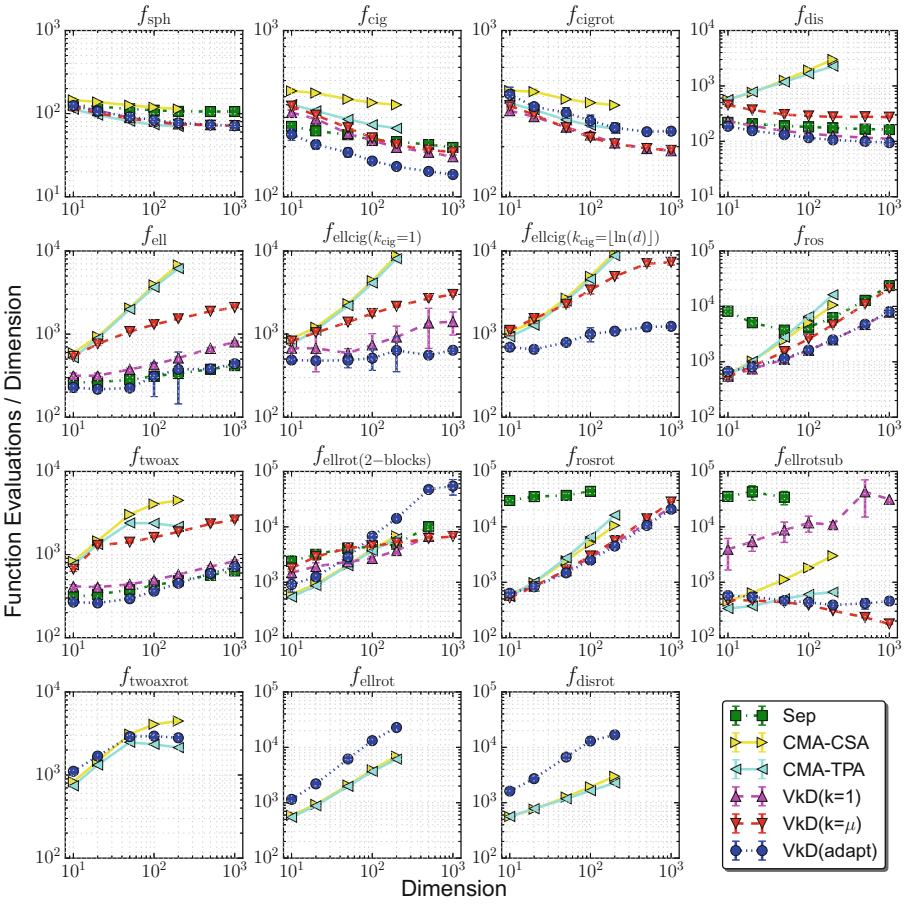

Fig. 3. Average and standard deviation of the number of function evaluations over ten independent runs v.s. dimension d. The data points displayed in the figures correspond to the setting for which the target function values are reached within the given budged $(10^5 d)$ in all the ten runs. Note that the experiments have been done up to $d = 200$ for the CMA-CSA and CMA-TPA, whereas the maximum d is 1000 for the other algorithms except for f_{twoaxrot}, f_{ellrot}, f_{disrot} where the maximum d is 200.

5 Discussion

The proposed mechanism adapts the number k of vectors, i.e., the model complexity of the restricted covariance matrix, online for VkD-CMA. The proposed approach removes the need for pre-selection of the restricted covariance matrix model, which is the main shortage of VkD-CMA. The experimental results reveal that the proposed algorithm is competitive with a variant of the CMA-ES with a nearly optimal model of the restricted covariance matrices on most of the problems with limited variable dependencies, importantly without any tuning of the

model in advance. On the fully non-separable functions we observe slowdown compared to the CMA-ES; it takes about 7.5 times more function evaluations at the worst case (f_{disrot}, $d = 200$). On f_{disrot}, the inverse Hessian has one smaller eigenvalue than the others, meaning that there is one direction in which the function value is sensitive. For such a function, the active covariance matrix update [9], i.e., assigning negative weights for unsuccessful candidates, is known to accelerate the covariance matrix adaptation in the standard CMA-ES. We expect that it improves the performance of the proposed algorithm. This is one of the main future work.

Acknowledgments. This work is partially supported by JSPS KAKENHI Grant Number 15K16063.

References

1. Akimoto, Y.: Analysis of a natural gradient algorithm on monotonic convex-quadratic-composite functions. In: Proceedings of Genetic and Evolutionary Computation Conference, pp. 1293–1300. ACM (2012)
2. Akimoto, Y., Auger, A., Hansen, N.: Comparison-based natural gradient optimization in high dimension. In: Proceedings of Genetic and Evolutionary Computation Conference, pp. 373–380. ACM (2014)
3. Akimoto, Y., Hansen, N.: Projection-based restricted covariance matrix adaptation for high dimension. In: Proceedings of Genetic and Evolutionary Computation Conference. ACM (2016, to appear)
4. Arnold, D.V.: Optimal weighted recombination. In: Wright, A.H., Vose, M.D., De Jong, K.A., Schmitt, L.M. (eds.) FOGA 2005. LNCS, vol. 3469, pp. 215–237. Springer, Heidelberg (2005)
5. Hansen, N., Atamna, A., Auger, A.: How to assess step-size adaptation mechanisms in randomised search. In: Bartz-Beielstein, T., Branke, J., Filipič, B., Smith, J. (eds.) PPSN 2014. LNCS, vol. 8672, pp. 60–69. Springer, Heidelberg (2014)
6. Hansen, N., Auger, A.: Principled design of continuous stochastic search: from theory to practice. In: Borenstein, Y., Moraglio, A. (eds.) Theory and Principled Methods for the Design of Metaheuristics. NCS. Springer, Berlin (2014)
7. Hansen, N., Muller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol. Comput. **11**(1), 1–18 (2003)
8. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evol. Comput. **9**(2), 159–195 (2001)
9. Jastrebski, G., Arnold, D.V.: Improving evolution strategies through active covariance matrix adaptation. In: 2006 IEEE Congress on Evolutionary Computation, pp. 9719–9726. IEEE (2006)
10. Loshchilov, I.: A computationally efficient limited memory CMA-ES for large scale optimization. In: Proceedings of Genetic and Evolutionary Computation Conference, pp. 397–404 (2014)
11. Ros, R., Hansen, N.: A simple modification in CMA-ES achieving linear time and space complexity. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 296–305. Springer, Heidelberg (2008)