Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 525–535

DOI: 10.15439/2015F261
ACSIS, Vol. 5

Small Populations, High-Dimensional Spaces: Sparse Covariance Matrix Adaptation

Silja Meyer-Nieberg
Department of Computer Science,
Universität der Bundeswehr München,
Werner-Heisenberg Weg 37,
85577 Neubiberg, Germany
Email: silja.meyer-nieberg@unibw.de

Erik Kropat
Department of Computer Science,
Universität der Bundeswehr München,
Werner-Heisenberg Weg 37,
85577 Neubiberg, Germany
Email: erik.kropat@unibw.de

***Abstract*—Evolution strategies are powerful evolutionary algorithms for continuous optimization. The main search operator is mutation. Its extend is controlled by the covariance matrix and must be adapted during a run. Modern Evolution Strategies accomplish this with covariance matrix adaptation techniques. However, the quality of the common estimate of the covariance is known to be questionable for high search space dimensions. This paper introduces a new approach by changing the coordinate system and introducing sparse covariance matrix techniques. The results are evaluated in experiments.**

I. Introduction

EVOLUTIONARY COMPUTATION has a long research tradition. The field comprises today the main classes genetic algorithms, genetic programming, evolution strategies, evolutionary programming, and differential evolution. Evolution strategies (ESs), on which the research presented in this paper focuses, are primarily used for optimizing continuous functions. The function is not required to be analytical.

Evolution strategies rely on mutation, i.e., on the random perturbation of candidate solutions to navigate the search space. The process must be controlled in order to achieve good performance. For this, modern ESs apply covariance matrix adaptation in several variants [1]. Nearly all approaches take the sample covariance matrix into account. This estimator is known to be problematic in the case of small sample sizes compared to the search space dimensionality. Since the population size in evolution strategies is typically considerably smaller, this paper argues that the adaptation process may profit from the introduction of different estimators.

So far, evolutionary algorithms or related approaches have only seldom considered statistical estimation methods targeted at high-dimensional spaces. The reason may be twofold: The improved quality of the estimators induces increased computational costs which may lower the convergence velocity of the algorithm. In addition, the estimators are developed and analyzed for samples of independently, identically distributed random variables. Since evolutionary algorithms deploy selection based on rank or fitness, the assumption of the same distribution is not valid. This may be the reason as to why the literature research has resulted in only one previous approach [2]. There, the authors considered Gaussian based estimation of distribution algorithms. The problem they were faced with concerned a non-positive definiteness of the estimated covariance matrix. Therefore, Dong and Yao augmented the algorithm with a shrinkage procedure to guarantee positive definiteness. Shrinkage is one of the common methods to improve the quality of the sample covariance, see e.g. [3]. While the approach in [2] resembles the Ledoit-Wolf estimator [3], it adapted the shrinkage intensity during the run.

This paper extends the work presented in [4], [5], where Ledoit-Wolf shrinkage estimators were analyzed, combined with a maximum entropy approach, and integrated into evolution strategies. While the results were promising, the question remained how to adapt the parameter of the estimator. Therefore, in this paper, another computational simple estimation method is investigated: thresholding.

The paper is structured as follows. First, modern evolution strategies with covariance adaptation are introduced. Afterwards, a short motivation as to why we think that the covariance computation in ESs may profit from estimation theory for high-dimensional spaces is provided. The next section describes the new approach developed and is followed by the experimental section which compares the new approach against the original ES. Conclusions and a discussion of potential future research constitute the last part of the paper.

A. Modern Evolution Strategies

This section provides a short introduction into evolutionary algorithms focussing on evolution strategies and covariance matrix adaptation. Evolutionary algorithms (EAs) [6] in general are population-based stochastic search and optimization algorithms used when only direct function measurements are possible.

Their iterative search process requires the definition of termination criteria and stops if these are fulfilled. In each generation, a series of operations is performed: selection for reproduction, followed by offspring creation, i.e. recombination and mutation processes, and finally survivor selection. The initial population of candidate solutions is either drawn randomly from the permissable search space or is initialized based on information already obtained. First of all, the offspring population has to be created. For this, a subset of the parents is determined during *parent selection*. The creation

978-83-60810-66-8/$25.00

of the offspring is based on recombination and mutation. Recombination combines traits from two or more parents resulting in one or more intermediate offspring. In contrast, mutation is an unary operator changing the components of an individual randomly. After the offspring have been created, survivor selection is performed to determine the next parent population. The different variants of evolutionary algorithms adhere to the same principles in general, but they may differ in the representation of the solutions and how the selection, recombination, and mutation processes are realized.

a) Evolution Strategies: Evolution strategies (ESs) [7], [8] are used for continuous optimization $f : \mathbb{R}^N \to \mathbb{R}$. Several variants have been introduced see e.g. [9], [1]. In many cases, a population of μ parents is used to create a set of λ offspring, with $\mu \leq \lambda$. For recombination, ρ parents are chosen uniformly at random without replacement and are then recombined. Recombination usually consists of determining the (weighted) mean or centroid of the parents [9]. The result is then mutated by adding a normally distributed random variable with zero mean and covariance matrix $\sigma^2\mathbf{C}$. While there are ESs that operate without recombination, the mutation process is seen as the essential process. It is often interpreted as the main search operator. After the offspring have been created, the individuals are evaluated using the function to be optimized or a derived function which allows an easy ranking of the population. Only the rank of an individual is important for the selection. In the case of continuous optimization, the old parent population is typically discarded with the selection considering only the λ offspring of which the μ best are chosen.

The covariance matrix which is central to the mutation must be adapted during the run: Evolution strategies with ill-adapted parameters converge only slowly or may even fail in the optimization. Therefore, research on methods for adapting the scale factor σ or the full covariance matrix has a long research tradition in ESs dating back to their origins [7]. The next section describes one of the current approaches.

b) Updating the Covariance Matrix: To our knowledge, covariance matrix adaptation comprises two main classes: one applied in the *covariance matrix adaptation evolution strategy* (CMA-ES) [10] and an alternative used in the *covariance matrix self-adaptation evolution strategy* (CMSA-ES) [11]. Both consider information from the present population combining it with information from the search process so far. The CMA-ES is one of the most powerful evolution strategies and often referred to as the standard in ESs. However, as pointed out in [11], its scaling behavior with the population size may not be good. Beyer and Sendhoff [11] showed that the CMSA-ES performs comparably to the CMA-ES for smaller populations but that is less computational expensive for larger population sizes.

Therefore, the present paper focuses on the CMSA-ES leaving the CMA-ES for future research. The CMSA-ES uses weighted intermediate recombination, in other words, the weighted centroid $\mathbf{m}^{(g)}$ of the μ best individuals of the population is computed. To create the offspring, random vectors are drawn from the multivariate normal distribution $\tilde{\mathcal{N}}(\mathbf{m}^{(g)}, (\sigma^{(g)})^2\mathbf{C}^{(g)})$. The notation of covariance matrix as $(\sigma^{(g)})^2\mathbf{C}^{(g)}$ illustrates that the actual covariance matrix is interpreted as the combination of a general scaling factor (or step-size or mutation strength) with a rotation matrix. Following the usual practice in literature on evolution strategies the latter matrix $\mathbf{C}^{(g)}$ is referred to as *covariance matrix* in the remainder of the paper.

The covariance matrix update is based upon the common estimate of the covariance using the newly created population. Instead of considering all offspring for deriving the estimates, though, it introduces a bias towards good search regions by taking only the μ best individuals into account. Furthermore, it does not estimate the mean anew but uses the weighted mean $\mathbf{m}^{(g)}$. Following [10],

$$\mathbf{z}_{m:\lambda}^{(g+1)} := \frac{1}{\sigma^{(g)}}\left(\mathbf{x}_{m:\lambda}^{(g+1)} - \mathbf{m}^{(g)}\right) \tag{1}$$

are determined with $\mathbf{x}_{m:\lambda}$ denoting the mth best of the λ particle according to the fitness ranking. The rank-μ update then obtains the covariance matrix as

$$\mathbf{C}_{\mu}^{(g+1)} := \sum_{m=1}^{\mu} w_m \mathbf{z}_{m:\lambda}^{(g+1)} (\mathbf{z}_{m:\lambda}^{(g+1)})^{\mathrm{T}} \tag{2}$$

which is usually a positive semi-definite matrix since $\mu \ll N$. The weights w_m should fulfill $w_1 \geq w_2 \geq \ldots \geq w_\mu$ with $\sum_{m=1}^{\mu} w_i = 1$. To derive reliable estimates larger population sizes are required which would lower the algorithm's speed. Therefore, past covariance matrices are taken into account via the convex combination of (2) with the sample covariance being shrunk towards the old covariance

$$\mathbf{C}^{(g+1)} := (1 - \frac{1}{c_\tau})\mathbf{C}^{(g)} + \frac{1}{c_\tau}\mathbf{C}_{\mu}^{(g+1)} \tag{3}$$

with the weights usually set to $w_m = 1/\mu$ and

$$c_\tau = 1 + \frac{N(N+1)}{2\mu}, \tag{4}$$

see [11]. As long as $\mathbf{C}^{(g)}$ is positive semi-definite, (3) will result in a positive definite matrix.

c) Step-Size Adaptation: The CMSA implements the step-size using *self-adaptation* first introduced in [7] and developed further in [8]. Here, evolution is used for fitting the strategy parameters of the mutation process. In other words, the scaling parameter or in its full form, the complete covariance matrix, undergoes recombination, mutation, and indirect selection processes. The working principle is based on an indirect stochastic linkage between good individuals and appropriate parameters: Well adapted parameters should result more often in better offspring than too large or too small values or misleading directions. Although self-adaptation has been developed to adapt the whole covariance matrix, it is applied today mainly to adapt the step-size or a diagonal covariance matrix. In the case of the mutation strength, usually a log-normal distribution

$$\sigma_l^{(g)} = \sigma_{\mathrm{base}} \exp(\tau \mathcal{N}(0,1)) \tag{5}$$

is used for mutation. The parameter τ is called the *learning rate* and is usually chosen to scale with $1/\sqrt{2N}$. The baseline σ_{base} is either the mutation strength of the parent or if recombination is used the recombination result. For the step-size, it is possible to apply the same type of recombination as for the positions although different forms – for instance a multiplicative combination – could be used instead. The self-adaptation of the step-size is referred to as *σ-self-adaptation* (σSA) in the remainder of this paper.

The newly created mutation strength is then directly used in the mutation of the offspring. If the resulting offspring is sufficiently good, the scale factor is passed to the next generation.

Self-adaptation with recombination has been shown to be "robust" against noise [12] and is used in the CMSA-ES as the update rule for the scaling factor.

B. Concerning the Covariance Matrix Adaptation ...

In the case of $\lambda > 1$, the sample covariance (2) appears in nearly any adaptation process. Disregarding the distortion due to selection, the sample covariance as the maximum likelihood estimator of the true covariance matrix is known as a good and reliable estimate if $\mu \gg N$. Evolution strategies typically operate with $\mu < N$, however. For example, following [13] the sizes of the parent and offspring populations in the standard CMA-ES should be chosen as $\lambda = \lfloor \log(3N) \rfloor + 4$ and $\mu = \lfloor \lambda/2 \rfloor$.

Unfortunately, $\mu < N$ leads to problems with respect to the covariance estimation. This is a well-known problem in statistics [14], [15], giving raise to a broad range on literature on alternative estimators e.g. [15], [16], [17], [18], [19], [20], [21], [22], [23]. The quality of a maximum likelihood estimate may be insufficient – especially for high-dimensional spaces, see e.g. [16]. For example, Marčenko and Pastur showed that if $N/\mu \not\to 0$ but $N/\mu \in (0,1)$, instead, the eigenvalues of the covariance matrix are distributed in the interval $((1 - \sqrt{N/\mu})^2, (1 + \sqrt{N/\mu})^2)$ in the case of the standard normal distribution [17].

Equation (3) actually attempts to counteract the singularity of the population covariance matrix by using the well-known concept of shrinking. However, some distinctive differences are present. First of all, the target is a full covariance matrix whereas shrinkage typically considers simpler regulation forms as e.g. a diagonal matrix. Secondly, the parameter is usually determined via optimizing a performance measure.

Seeing that evolution strategies already apply some kind of shrinkage, some questions arise: Can we improve the estimator further by not only "shrinking" the population or sample covariance matrix but by applying further concepts stemming from the estimation of high-dimensional covariance matrices? And considering that (3) is one regulation technique among several, is it possible to find another well-performing substitute? Or did research in evolution strategies already happen upon the best technique possible?

II. A Sparse Covariance Matrix Adaptation

This section introduces the new covariance adaptation technique which uses thresholding to transform the population covariance matrix. The decision for thresholding is based upon the comparatively computational efficiency of the approach.

A. Space Transformation

The ideal covariance matrix for the search depends on the function landscape which is unknown in practical applications. Considering the smooth test functions of typical black-box optimization suites, shows that the Hessians of several functions, as e.g. the separable functions, can be classified as sparse or approximately sparse matrices following the definitions introduced later.

Therefore, sparse structures of the covariance matrix suffice which is exemplified by the separable CMA-ES [24] which restricts the covariance to a diagonal matrix in case of separability to allow fast progress to the optimal solution. For the general case, a spare structure may not be suitable, however.

For this reason, the paper does not require sparseness of the original covariance matrix, although it would be interesting to see how such a variant would perform on the test suites. Instead, it considers a transformation. As argued in [25], an change of the coordinate system may improve the performance of an evolution strategy. Therefore, an adaptive encoding was introduced. In each iteration, the covariance matrix is adapted following the rules of the CMA-ES. Its spectral decomposition is used to change the basis. The creation of new search points is carried out in the eigenspace of the current covariance matrix and the main search parameters of the CMA-ES are updated there. After selection, the covariance matrix is adapted and utilized for a renewed decoding and encoding.

This paper also addresses a change of the coordination system. However, we address the covariance matrix adaptation and estimation itself which in [25] occurs in the original space. Here, we argue that a switch to the eigenspace of the old covariance matrix $\mathbf{C}^{(g)}$ may be beneficial for the estimation of the covariance matrix itself.

Let the covariance matrix $\mathbf{C}^{(g)}$ be a symmetric, positive definite $N \times N$ matrix. The condition holds for the original adaptation since (3) combines a positive definite with a positive semi-definite matrix. As we will see below, in the case of thresholding the condition may not always be fulfilled. Assuming a positive definite matrix allows carrying out a spectral decomposition: Let $\mathbf{v}_1, \ldots, \mathbf{v}_N$ denote the N eigenvectors with the eigenvalues $\lambda_1, \ldots, \lambda_N$, $\lambda_j > 0$. Note, the eigenvectors form a orthonormal basis of $\mathbb{R}^N$, i.e., $\mathbf{v}_i^{\mathrm{T}}\mathbf{v}_i = 1$ and $\mathbf{v}_i^{\mathrm{T}}\mathbf{v}_j = 0$, if $i \neq j$. We define $\mathbf{V} := (\mathbf{v}_1, \ldots, \mathbf{v}_N)$ as the modal matrix. It then holds that $\mathbf{V}^{-1} = \mathbf{V}^{\mathrm{T}}$. Switching to the eigenspace of $\mathbf{C}^{(g)}$ results in the representation of the covariance matrix

$$\Lambda^{(g)} = \mathbf{V}\mathbf{C}^{(g)}\mathbf{V}^{\mathrm{T}} \tag{6}$$

as a diagonal matrix with the eigenvalues as the diagonal entries. Diagonal matrices are sparse matrices, thus for the estimation of the covariance matrix the more efficient procedures

for sparse structures could be used. However, it is not the goal to re-estimate $\mathbf{C}^{(g)}$ but to estimate the true covariance matrix of the distribution indicated by the sample $\mathbf{z}_{1;\lambda}, \ldots, \mathbf{z}_{\mu;\lambda}$.

Before continuing, it should be noted that several definitions of sparseness exist. Usually, it is demanded that the number of non-zero elements in a row may not exceed a predefined limit $s_0(N) > 0$, i.e.,

$$\max_i \sum_{j=1}^{N} \delta(|a_{ij}| > 0) \leq s_0(N), \tag{7}$$

which should grow only slowly with N. The indicator function δ fulfills $\delta(\cdot) = 1$ if the condition is met and is zero otherwise. This definition can, however, be relaxed to a more general definition of sparseness, also referred to as approximate sparseness. Cai and Liu [22] consider the following uniformity class of sparse matrices

Definition 1. *Let $s_0(N) > 0$ and let $\cdot \succ 0$ denote positive definiteness. Then a class of sparse covariance matrices is defined as*

$$\begin{aligned}\mathcal{U}_q^* &:= \mathcal{U}_q^*(s_0(N)) \\ &= \left\{\Sigma : \Sigma \succ 0, \max_i \sum_{j=1}^{p} (\sigma_{ii}\sigma_{jj})^{\frac{(1-q)}{2}} |\sigma_{ij}|^q \leq s_0(N)\right\}\end{aligned} \tag{8}$$

for some $0 \leq q < 1$.

Definition 1 requires the entries of the covariance matrix to lie within a weighted l_q ball. The weight is given by the variances. Cai and Liu [22] introduce a thresholding estimator that requires the assumption above. Its convergence rate towards the true covariance depends on $s_0(N)(\log(N)/\mu)^{(1-q)/2}$. Therefore, the number $s_0(N) > 0$ should again grow only "slowly" for $N \to \infty$.

Definition 1 leads to the main assumption of the paper. Consider an evolution strategy in the search space. The new sample that is the offspring population has been created with the help of the old covariance matrix. The covariance matrix of the selected sample differs from the previous. The deviations of from its structure stem from finite sampling characteristics and rank-based selection. Assuming that the form of the covariance matrix will not change considerably in one iteration, the new underlying covariance matrix should be sparse in the eigenspace of the old covariance, however.

Assumption 1. *Let $\Sigma^{(g+1)}$ denote the true covariance matrix of the selected offspring. Consider the old covariance $\mathbf{C}^{(g)}$ with its modal matrix $\mathbf{V}$. Then $\hat{\Lambda} = \mathbf{V}\Sigma^{(g+1)}\mathbf{V}^{\mathrm{T}}$ is approximately sparse, i. e. $\hat{\Lambda} \in \mathcal{U}_q^*$ for some $0 \leq q < 1$.*

Assuming the validity of the assumption, we change the coordinate system in order to perform the covariance matrix estimate. Reconsider the normalized (apart from the covariance matrix) mutation vectors $\mathbf{z}_{1;\lambda}, \ldots, \mathbf{z}_{\mu;\lambda}$ that were associated with the μ best offspring. Their representation in the eigenspace reads

$$\hat{\mathbf{z}}_{m;\lambda} = \mathbf{V}^{\mathrm{T}}\mathbf{z}_{m;\lambda} \text{ for } m = 1, \ldots, \mu. \tag{9}$$

The transformed population covariance is then estimated as

$$\hat{\mathbf{C}}_\mu = \sum_{m=1}^{\mu} w_m \hat{\mathbf{z}}_{m;\lambda}\hat{\mathbf{z}}_{m;\lambda}^{\mathrm{T}}. \tag{10}$$

The estimate (10) will be used to compute the final estimator. In the next section, we discuss potential estimators for sparse covariance matrices.

B. Sparse Covariance Matrix Estimation

In recent years, covariance matrix estimation in high-dimensional spaces has received a lot of attention. In the case of sparse covariance matrices, banding, tapering, and thresholding can be applied, see e.g. [26] All three make use of the fact that many entries of the matrix that shall be estimated are actually zero or at least very small. Banding and tapering differ from thresholding in that they assume a specific matrix structure in other words they assume an ordering of the variables which is for instance often the case in time-series analysis. Banding and tapering approaches typically lead to consistent estimators if $\log(N)/\mu \to 0$.

Thresholding does not assume a natural order of the variables. Instead, it discards entries which are smaller than a given threshold $\epsilon > 0$. For a matrix $\mathbf{A}$, the thresholding operator $T_\epsilon(\mathbf{A})$ is defined as

$$T_\epsilon(\mathbf{A}) := (a_{ij}\delta(|a_{ij}| \geq \epsilon))_{N\times N}. \tag{11}$$

The choice of the threshold is critical for the quality of the resulting estimate.

Equation (11) represents a example of universal thresholding with a hard thresholding function. Equation (11) can be extended in several ways. On the one hand, the threshold may depend on the entry itself, and on the other hand, instead of the hard threshold applied, a generalized shrinkage function $s_\lambda(\cdot)$ can be used. Following [22] , the function $s_\lambda(\cdot)$ should have the following properties

i) $\exists c > 0$: $s_\lambda(x) \leq c|y|$ $\forall\, x, y$ which satisfy $|x - y| \leq \lambda$,
ii) $s_\lambda(x) = 0$ $\forall x \leq \lambda$,
iii) $|s_\lambda(x) - x| \leq \lambda$ $\forall x \in \mathbb{R}$.

Several functions have been introduced that fulfill i)-iii), as e.g. the soft-thresholding

$$s_\lambda(x) = \mathrm{sign}(x)(|x| - \lambda)_+ \tag{12}$$

or the Lasso

$$s_\lambda(x) = |x|(1 - |\frac{\lambda}{x}|^\eta)_+ \tag{13}$$

with $(x)_+ := \max(0, x)$. In this paper, the threshold λ_{ij} is defined component-wise and not universal. Since its correct choice is difficult to decide a priori, adaptive thresholding is applied as in [22], setting

$$\lambda_{ij} := \lambda_{ij}(\delta) = \delta\sqrt{\frac{\hat{\theta}_{ij}\log N}{\mu}} \tag{14}$$

with $\delta > 0$ can be either chosen as a constant or adapted data driven. The variable $\hat{\theta}_{ij}$ that appears in (14) is obtained as

$$\hat{\theta}_{ij} = \frac{1}{\mu}\sum_{m=1}^{\mu}[(\hat{z}_{mi} - \overline{Z^i})((\hat{z}_{mj} - \overline{Z^j}) - \hat{c}_{ij}^{\mu}]^2 \tag{15}$$

Require: λ, μ, $\mathbf{C}^{(0)}$, $\mathbf{m}^{(0)}$, $\sigma^{(0)}$, τ, c_τ

1: $g = 0$
2: **while** termination criteria not met **do**
3: **for** $l = 1$ **to** λ **do**
4: $\sigma_l = \sigma^{(g)} \exp(\tau \mathcal{N}(0,1))$
5: $\mathbf{x}_l = \mathbf{m}^{(g)} + \sigma_l \bar{\mathcal{N}}(0, \mathbf{C}^{(g)})$
6: $f_l = f(\mathbf{x}_l)$
7: **end for**
8: Select $(\mathbf{x}_{1:\lambda}, \sigma_{1:\lambda}), \ldots, (\mathbf{x}_{\mu:\lambda}, \sigma_{\mu:\lambda})$
9: $\mathbf{m}^{(g+1)} = \sum_{m=1}^{\mu} w_m \mathbf{x}_{m:\lambda}$
10: $\sigma^{(g+1)} = \sum_{m=1}^{\mu} w_m \sigma_{m:\lambda}$
11: $\mathbf{z}_{m;\lambda} = \frac{\mathbf{x}_{m:\lambda} - \mathbf{m}^{(g)}}{\sigma^{(g)}}$ for $m = 1, \ldots, \mu$
12: $\mathbf{V}, \mathbf{D} \leftarrow \mathrm{spectral}(\mathbf{C}^{(g)})$
13: $\hat{\mathbf{z}}_{m;\lambda} = \mathbf{V}^{\mathrm{T}} \mathbf{z}_{m;\lambda}$ for $m = 1, \ldots, \mu$
14: $\hat{\mathbf{C}}_\mu = \sum_{m=1}^{\mu} w_m \hat{\mathbf{z}}_{m;\lambda} \hat{\mathbf{z}}_{m;\lambda}^{\mathrm{T}}$
15: $\hat{\mathbf{C}}_{\mathrm{thres}} = T_{S_{\lambda_{ij}}}(\hat{\mathbf{C}}_\mu)$
16: $\mathbf{C}_\mu = \mathbf{V}^{\mathrm{T}} \hat{\mathbf{C}}_{\mathrm{thres}} \mathbf{V}$
17: $\mathbf{C}^{(g+1)} = (1 - \frac{1}{c_\tau}) \mathbf{C}^{(g)} + \frac{1}{c_\tau} \mathbf{C}_\mu$
18: $g = g + 1$
19: **end while**

Fig. 1. The CMSA-ES with thresholding. The generation counter g is sometimes left out in order to simplify the notation. The symbol *spectral* stands for the spectral decomposition of the matrix into the modal matrix $\mathbf{V}$ and the diagonal matrix containing the eigenvalues $\mathbf{D}$. Rank-based deterministic selection of the μ best offspring is performed in line 8 based on the fitness f.

with $\hat{c}_{ij}^{\mu}$ denoting the (i,j)-entry of $\hat{\mathbf{C}}_\mu^{(g+1)}$, $\hat{z}_{mi}$ the ith component of $\hat{\mathbf{z}}_{m:\lambda}$, and $\overline{Z^i} := (1/\mu) \sum_{m=1}^{\mu} \hat{z}_{mi}$. Other thresholds have been introduced, see e.g. [27] and will be considered in future work.

While thresholding respects symmetry and non-negativeness properties, it results only in asymptotically positive definite matrices. Thus, for finite sample sizes, it does neither preserve nor induce positive definiteness in general. This holds for hard thresholding as well as for most cases of potential thresholding functions. As shown in [28], a positive semi-definiteness can only be guaranteed for a small class of functions for general matrices. In the case that the condition number of the matrix is sufficiently small, the group of functions that preserve positive definiteness can be widened to include also polynomials. In [27], procedures are discussed that result in positive definite matrices. As this paper aims for a proof of concept, it does not consider repair mechanisms.

C. Evolution Strategies with Sparse Covariance Adaptation

Component-wise adaptive thresholding can be integrated readily into evolution strategies. Figure 1 illustrates the main points of the algorithm. There are several ways to design the operator $T_{S_{\lambda_{ij}}}$. The first choice concerns the thresholding function $s_{\lambda_{ij}}(\cdot)$. The second question concerns whether thresholding should be applied to all entries of the covariance matrix (11) or only to the off-diagonal elements. This question is difficult to decide beforehand in the application context considered. Therefore, two variants are investigated

1) CMSA-Thres-ES (abbreviated to Thres): An evolution strategy with CMSA which applies thresholding in the eigenspace of the covariance, using the operator

$$T_{S_{\lambda_{ij}}}(\mathbf{A})_{ij} = s_{\lambda_{ij}}(a_{ij}) \tag{16}$$

and

2) CMSA-Diag-ES (abbreviated to Diag): An ES with covariance matrix adaptation which uses thresholding in the eigenspace of the covariance and excepts the diagonal elements with

$$T_{S_{\lambda_{ij}}}(\mathbf{A})_{ij} = \begin{cases} a_{ij} & \text{if } i = j \\ s_{\lambda_{ij}}(a_{ij}) & \text{if } i \neq j \end{cases} . \tag{17}$$

In statistics, thresholding is often applied only to the off-diagonal entries. Keeping the diagonal unchanged may however result in a too strong reliance on the structure of the old covariance matrix in our case. This may make a change of the search directions difficult. Therefore, both variants are taken into account.

III. Experiments

The experiments are performed for the search space dimensions $N = 10$ and 20. Since we aim for a general approach, the performance of the new techniques should also be analyzed for lower dimensional spaces. The maximal number of fitness evaluations is set to $\mathrm{FE}_{\max} = 2 \times 10^5 N$. The start position of the algorithms is randomly chosen from $[-4, 4]^N$. The population size were chosen as $\lambda = \lfloor \log(3N) + 8 \rfloor$ and $\mu = \lceil \lambda/2 \rceil$. The weights w_m were set to $w_m = 1/\mu$.

A run terminates before reaching the maximal number of evaluations, if the difference between the best value obtained so far and the optimal fitness value $|f_{\mathrm{best}} - f_{\mathrm{opt}}|$ is below a predefined target precision set to 10^{-8}. For each fitness function and dimension, 15 runs are used in accordance to the practice of the black box optimization workshops, see below. If the search stagnates, indicated by changes of the best values being below 10^{-8} for $10 + \lceil 30N/\lambda \rceil$ generations, the ES is restarted. The Lasso thresholding function (13) with $\eta = 4$ was chosen as the thresholding function and by performing a preliminary series of experiments the scaling factor δ in (15) was set to $\delta = 2 \max(\hat{\mathbf{C}}_\mu)$. Both choices can be probably improved. Since the paper strives for a first proof of concept, a detailed investigation of good parameter settings will be performed in future research.

A. Test Suite

For the experiments, the algorithms were implemented in MATLAB. The paper uses black box optimization benchmarking (BBOB) software framework and the test suite introduced for the black box optimization workshops, see [29]. The goal of the workshop is to benchmark and to compare metaheuristics and other direct search methods for continuous optimization. The framework[1] allows the plug-in of algorithms adhering to a common interface and provides a comfortable way of generating the results in form of tables and figures.

[1]Latest version under http://coco.gforge.inria.fr

Sphere	$f(\mathbf{x}) = \|\mathbf{z}\|^2$
Rosenbrock	$f(\mathbf{x}) = \sum_{i=1}^{N-1} 200(z_i^2 - z_{i+1})^2 + (z_i - 1)^2$
Ellipsoidal	$f(\mathbf{x}) = \sum_{i=1}^{N} 10^{6\frac{i-1}{N-1}} z_i^2$
Discus	$f(\mathbf{x}) = 10^6 z_1^2 + \sum_{i=2}^{N} z_i^2$
Rastrigin	$f(\mathbf{x}) = 10\big(N - \sum_{i=1}^{N} \cos(2\pi z_i)\big) + \|\mathbf{z}\|^2$

TABLE I
SOME OF THE TEST FUNCTIONS USED FOR THE COMPARISON OF THE ALGORITHMS.

The test suite contains noisy and noise-less functions with the position of the optimum changing randomly from run to run. This paper focuses on the noise-less test suite which contains 24 functions [30]. They can be divided into four classes: separable functions (function ids 1-5), functions with low/moderate conditioning (ids 6-9), functions with high conditioning (ids 10-14), and two groups of multimodal functions (ids 15-24). Among the unimodal functions with only one optimal point, there are separable functions given by the general formula

$$f(\mathbf{x}) = \sum_{i=1}^{N} f_i(x_i) \tag{18}$$

which can be solved by optimizing each component separately. The simplest member of this class is the (quadratic) sphere with $f(\mathbf{x}) = \|\mathbf{x}\|^2$. Other functions include ill-conditioned functions, like for instance the elliposoidal function, and multimodal functions (Rastrigin) which represent particular challenges for the optimization (Table I). The variable $\mathbf{z}$ denotes a transformation of $\mathbf{x}$ in order to keep the algorithm from exploiting certain particularities of the function, see [30].

B. Performance Measure

The following performance measure is used in accordance to [29]. The expected running time (ERT) gives the expected value of the function evaluations (f-evaluations) the algorithm needs to reach the target value with the required precision for the first time, see [29]. In this paper, we use

$$\mathrm{ERT} = \frac{\#(FEs(f_{\mathrm{best}} \geq f_{\mathrm{target}}))}{\#succ} \tag{19}$$

as an estimate by summing up the fitness evaluations $FEs(f_{\mathrm{best}} \geq f_{\mathrm{target}})$ of each run until the fitness of the best individual is smaller than the target value, divided by all successfull runs.

C. Results and Discussion

The findings are interesting – indicating advantages for thresholding in many but not in all cases. The result of the comparison depends on the function class. In the case of the separable functions with ids 1-5, the strategies behave on the whole very similar in the case of both dimensionalities 10D and 20D. This can be seen in the empirical cumulative distribution functions plots in Fig. 2 and Fig. 3 for example.

Concerning the particular functions, differences are revealed as Tab. II and Tab. III show for the expected running time (ERT) which is provided for several precision targets. The expected running time is provided relative to the best results achieved during the black-box optimization workshop in 2009. The first line of the outcomes for each function reports the ERT of the best algorithm of 2009. However, not only the ERT values but also the number of successes is important. The ERT can only be measured if the algorithm achieved the respective target in the run. If the number of trials where is the full optimization objective has been reached is low then the remaining targets should be discussed with care. If only a few runs contribute to the result, the findings may be strongly influenced by initialization effects. To summarize, only a few cases end with differences that are statistically significant. To achieve this, the algorithm has to perform significantly better than both competing methods – the other thresholding variant and the original CMSA-ES.

In the case of the sphere (function with id 1), slight advantages for the thresholding variants are revealed. A similar observation can be made for the second function, the separable ellipsoid. Here, both thresholded ESs are faster, with the one that only shrinks the off-diagonal elements significantly (Tab. III). This is probably due to the enforced more regular structure.

No strategy is able to reach the required target precision in the case of the separable Rastrigin (id 3) and the separable Rastrigin-Bueche (id 4). Since all strategies only achieve the lowest target precision of 10^1, a comparison is not performed. The linear slope is solved fast by all, with the original CMSA-ES the best strategy.

In the case of the function class containing test functions with low to moderate conditioning, different findings can be made for the two search space dimensionalities. This is also shown by the empirical cumulative distribution functions plots in Fig. 2 and Fig. 3, especially for $N = 10$. Also in the case of $N = 10$, Table II shows that the strategies with thresholding achieve a better performance in a majority of cases. In addition, thresholding that is not applied to the diagonal appears to lead to a well-performing strategy with the exception of f9, the rotated Rosenbrock function, where it lead to the largest expected running times.

The results for f6, the so-called attractive sector, in 10D are astonishing. While the original CMSA-ES could only reach the required target precision in six of the 15 runs, the thresholding variants were able to succeed 14 times (CMSA-Thres-ES) and 13 times (CMSA-Diag-ES). The latter achieved lower expected running times, though. This does not transfer to 20D. Here, only a minority of runs were successfull for all strategies. Experiments with a larger number of fitness evaluations must be conducted in order to investigate the findings more closely.

The same holds for the step ellipsoid (id 7) which cannot be solved with the target precision required by any strategy. Concerning the lower precision targets, sometimes the CMSA-ES and sometimes the CMSA-Diag-ES appears superior. However, more research is required, since the number of runs entering the data for some of the target precisions is low and

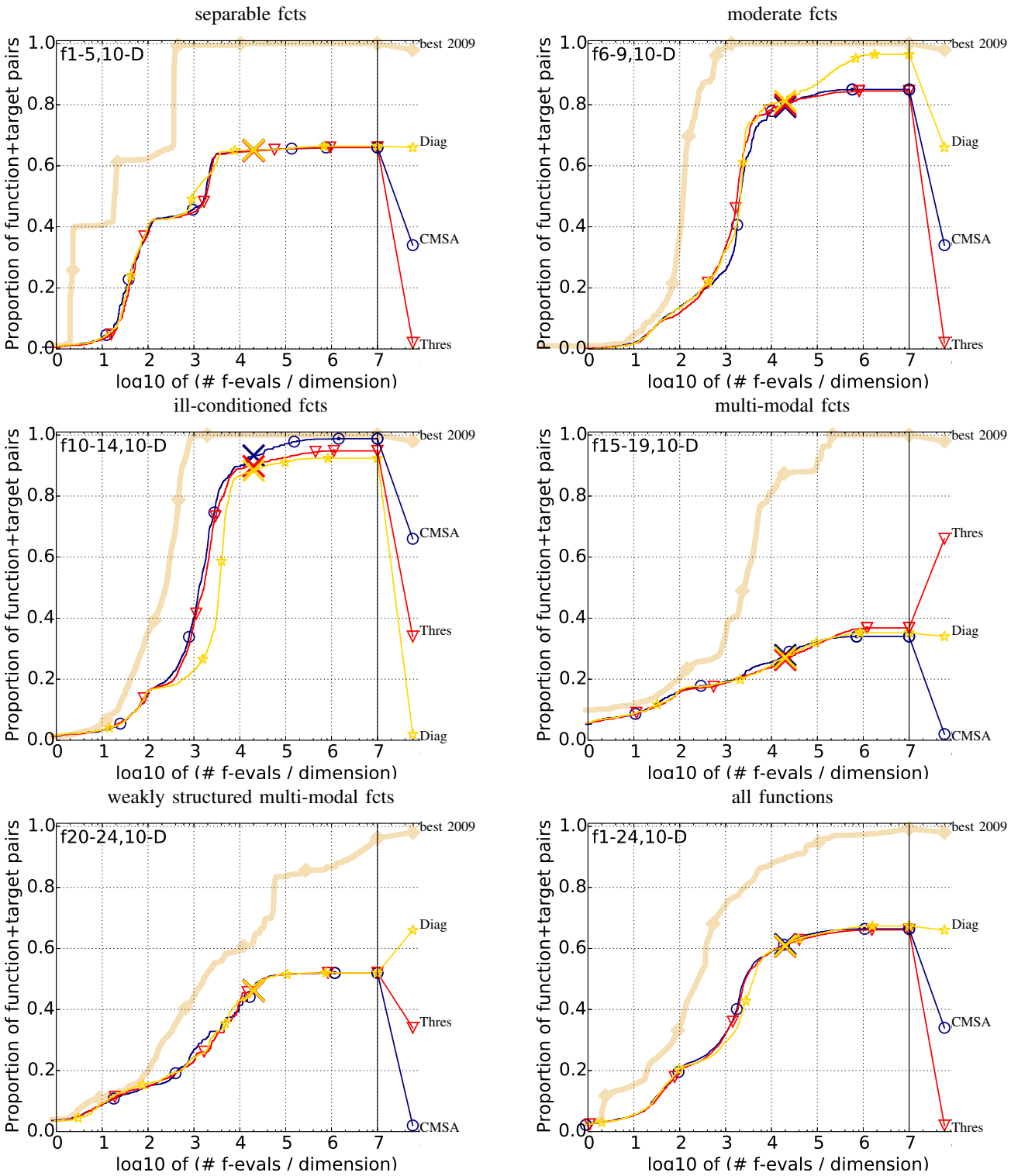

Fig. 2. Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 10-D. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each single target.

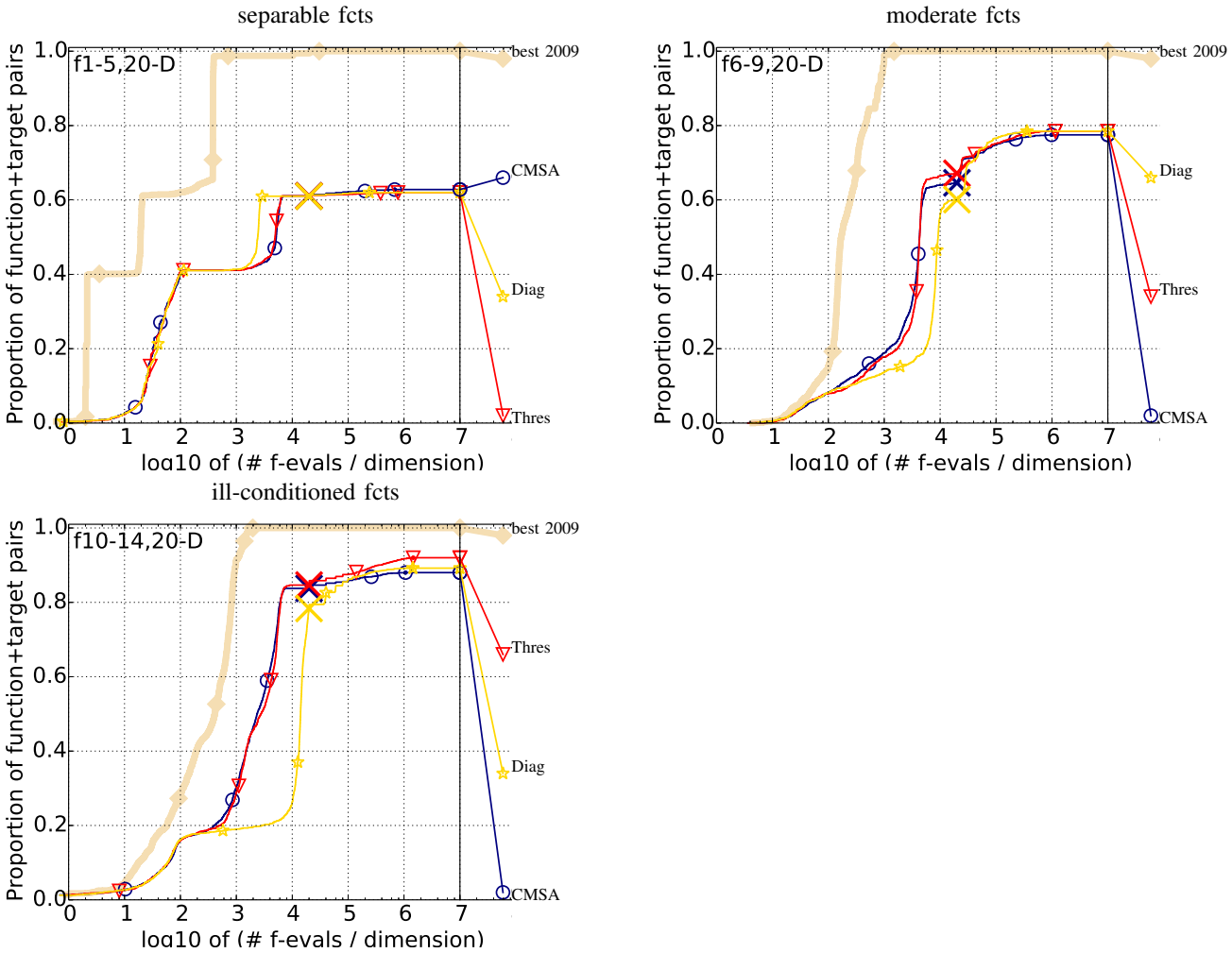

Fig. 3. Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 20-D. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each single target.

initial positions may be influential.

On the original Rosenbrock function (id 8), the CMSA-ES and the CMSA with thresholding show a similar behavior with the CMSA-ES performing better. In contrast, the thresholding variant that leaves the diagonal unchanged exhibits larger expected running times. The roles of the original CMSA-ES and the CMSA-Thres-ES reverse for the rotated Rosenbrock (id 9). Here, the best results can be observed for the thresholding variant. Again, the CMSA-Diag-ES performs worst.

In the case of ill-conditioned functions, the findings are mixed. In general, thresholding without including the diagonal does not appear to improve the performance. The strategy performs worst of all – an indicator that keeping the diagonal unchanged may be sometimes inappropriate due to the space transformation. However, since there are interactions with the choice of the thresholding parameters which may have resulted in comparatively too large diagonal elements, we need to address this issue further before coming to a conclusion. First of all for $N = 10$, all strategies are successfull in all cases for the ellipsoid (id 10), the discus (id 11), the bent cigar (id 12), and the sum of different powers (id 14). Only the CMSA-ES reaches the optimization target in the case of the sharp ridge (id 13). This, however, only twice. The reasons for this require further analysis. Either the findings may be due to a violation of the sparseness assumption or considering that this is only a weak assumption the choice of the thresholding parameters and the function should be reconsidered.

All strategies exhibit problems in the case of the group of multi-modal functions, Rastrigin (id 15), Weierstrass (id 16), Schaffer F7 with condition number 10 (id 17), Schaffer F7 with condition 1000 (id 18), and Griewank-Rosenbrock F8F2 (id 19). Partly, this may be due to the maximal number of fitness evaluations permitted. Even the best performing methods of the 2009 BBOB workshop required more evaluations than we allowed in total. Thus, experiments with larger values for the maximal function evaluations should be conducted in future research. Concerning the preliminary targets with lower precision, the CMSA-ES achieves the best results in a majority

of cases. However, the same argumentation as for the step ellipsoid applies.

In the case of $N = 20$, the number of function evaluations that were necessary in the case of the best algorithms of 2009 to reach even the lower precision target of 10^{-1} exceeds the budget chosen here. Therefore, the function group is excluded from the analysis for $N = 20$ and not shown in Figure 3 and Table III.

The remaining group consists of multi-modal functions with weak global structures. Here, especially the functions with numbers 20 (Schwefel $x\sin(x)$), 23 (Kaatsuuras), and 24 (Lunacek bi-Rastrigin) represent challenges for the algorithms. In the case of $N = 10$, they can only reach the first targets of 10^1 and 10^0. Again, the maximal number of function evaluations should be increased to allow a more detailed analysis on these functions. For the case of the remaining functions, function 21, Gallagher 101 peaks, and function 22, Gallagher 21 peaks, the results indicate a better performance for the CMSA-ES versions with thresholding compared with the original algorithm. Again due to similar reasons as for the first group of multi-modal functions, the results are only shown for $N = 10$.

IV. Conclusions and Outlook

This paper adressed covariance matrix adaptation techniques for evolution strategies. The original versions are based on the sample covariance – an estimator known to be problematic. Especially in high-dimensional search spaces, where the population size does not exceed the search space dimensionality, the agreement of the estimator and the true covariance may be low. Therefore, thresholding, a comparably computationally simple estimation technique, has been integrated into the covariance adaptation process. Thresholding stems from estimation theory for high-dimensional spaces and assumes an approximately sparse structure of the covariance matrix. The matrix entries are therefore thresholded, meaning a thresholding function is applied. The paper considered adaptive entry-wise thresholding. Since the covariance matrix cannot be assumed to be sparse in general, a basis transformation was carried out and the thresholding process was performed in the transformed space. The performance of the resulting new covariance matrix adapting evolution strategies was compared to the original variant on the black-box optimization benchmarking test suite. Two main variants were considered: A CMSA-ES which subjected the complete covariance to thresholding and a variant which left the diagonal elements unchanged. While the latter is more common in statistics, it is not easy to justify its preferation in optimization. The first findings were interesting with the new variants performing better for several function classes. While this is promising, more experiments and analyses are required and will be performed in future research. This concerns e.g. which variant to use since it depended on the function which of the two performed best. Open questions concern among others the choice of the thresholding function and the scaling parameter for the threshold. In this paper, it was selected by a small series of experiments. Making the parameter completely data driven and thus depending on the current sample is the goal of ongoing research.

If the assumption that the representation of true covariance $\Sigma^{(g+1)}$ of the offspring population in the eigenspace of the previous covariance $C^{(g)}$ is approximately sparse should be violated in some cases, then it may be worthwhile to take a closer look at the convex combination of the new and the old covariance matrix. Further work will thus also consider applying thresholding to the traditionally obtained covariance.

References

[1] T. Bäck, C. Foussette, and P. Krause, *Contemporary Evolution Strategies*, ser. Natural Computing. Springer, 2013.

[2] W. Dong and X. Yao, "Covariance matrix repairing in gaussian based EDAs," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, 2007. doi: 10.1109/CEC.2007.4424501 pp. 415–422.

[3] O. Ledoit and M. Wolf, "A well-conditioned estimator for large dimensional covariance matrices," *Journal of Multivariate Analysis Archive*, vol. 88, no. 2, pp. 265–411, 2004.

[4] S. Meyer-Nieberg and E. Kropat, "Adapting the covariance in evolution strategies," in *Proceedings of ICORES 2014*. SCITEPRESS, 2014, pp. 89–99.

[5] ——, "A new look at the covariance matrix estimation in evolution strategies," in *Operations Research and Enterprise Systems*, ser. Communications in Computer and Information Science, E. Pinson, F. Valente, and B. Vitoriano, Eds. Springer International Publishing, 2015, vol. 509, pp. 157–172. ISBN 978-3-319-17508-9. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-17509-6_11

[6] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, ser. Natural Computing Series. Berlin: Springer, 2003.

[7] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Frommann-Holzboog Verlag, 1973.

[8] H.-P. Schwefel, *Numerical Optimization of Computer Models*. Chichester: Wiley, 1981.

[9] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies: A comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.

[10] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[11] H.-G. Beyer and B. Sendhoff, "Covariance matrix adaptation revisited - the CMSA evolution strategy -," in *PPSN*, ser. Lecture Notes in Computer Science, G. Rudolph *et al.*, Eds., vol. 5199. Springer, 2008. ISBN 978-3-540-87699-1 pp. 123–132.

[12] H.-G. Beyer and S. Meyer-Nieberg, "Self-adaptation of evolution strategies under noisy fitness evaluations," *Genetic Programming and Evolvable Machines*, vol. 7, no. 4, pp. 295–328, 2006.

[13] N. Hansen, "The CMA evolution strategy: A comparing review," in *Towards a new evolutionary computation. Advances in estimation of distribution algorithms*, J. Lozano *et al.*, Eds. Springer, 2006, pp. 75–102.

[14] C. Stein, "Inadmissibility of the usual estimator for the mean of a multivariate distribution," in *Proc. 3rd Berkeley Symp. Math. Statist. Prob. 1*, Berkeley, CA, 1956, pp. 197–206.

[15] ——, "Estimation of a covariance matrix," in *Rietz Lecture, 39th Annual Meeting*. Atlanta, GA: IMS, 1975.

[16] J. Schäffer and K. Strimmer, "A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics,," *Statistical Applications in Genetics and Molecular Biology*, vol. 4, no. 1, p. Article 32, 2005.

[17] V. A. Marčenko and L. A. Pastur, "Distribution of eigenvalues for some sets of random matrices," *Sbornik: Mathematics*, vol. 1, no. 4, pp. 457–483, 1967.

[18] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008. doi: 10.1093/biostatistics/kxm045. [Online]. Available: http://biostatistics.oxfordjournals.org/content/9/3/432.abstract

[19] E. Levina, A. Rothman, and J. Zhu, "Sparse estimation of large covariance matrices via a nested lasso penalty," *Ann. Appl. Stat.*, vol. 2, no. 1, pp. 245–263, 03 2008. doi: 10.1214/07-AOAS139. [Online]. Available: http://dx.doi.org/10.1214/07-AOAS139

TABLE II

EXPECTED RUNNING TIME (ERT IN NUMBER OF FUNCTION EVALUATIONS) DIVIDED BY THE RESPECTIVE BEST ERT MEASURED DURING BBOB-2009 IN DIMENSION 10. THE ERT AND IN BRACES, AS DISPERSION MEASURE, THE HALF DIFFERENCE BETWEEN 90 AND 10%-TILE OF BOOTSTRAPPED RUN LENGTHS APPEAR FOR EACH ALGORITHM AND TARGET, THE CORRESPONDING BEST ERT IN THE FIRST ROW. THE DIFFERENT TARGET Δf-VALUES ARE SHOWN IN THE TOP ROW. #SUCC IS THE NUMBER OF TRIALS THAT REACHED THE (FINAL) TARGET $f_{\text{opt}} + 10^{-8}$. THE MEDIAN NUMBER OF CONDUCTED FUNCTION EVALUATIONS IS ADDITIONALLY GIVEN IN *italics*, IF THE TARGET IN THE LAST COLUMN WAS NEVER REACHED. ENTRIES, SUCCEEDED BY A STAR, ARE STATISTICALLY SIGNIFICANTLY BETTER (ACCORDING TO THE RANK-SUM TEST) WHEN COMPARED TO ALL OTHER ALGORITHMS OF THE TABLE, WITH $p = 0.05$ OR $p = 10^{-k}$ WHEN THE NUMBER k FOLLOWING THE STAR IS LARGER THAN 1, WITH BONFERRONI CORRECTION BY THE NUMBER OF INSTANCES.

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f1	22	23	23	23	23	23	23	15/15
CMSA	4.0(3)	8.6(3)	14(4)	19(4)	26(6)	38(8)	50(5)	15/15
Thres	4.2(2)	9.2(3)	14(4)	18(3)	24(3)	35(5)	46(7)	15/15
Diag	**3.2**(1)	**7.5**(2)	**12**(3)	**18**(2)	**23**(4)	**34**(3)	**44**(4)	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f2	187	190	191	191	193	194	195	15/15
CMSA	65(34)	85(21)	96(29)	105(16)	109(25)	113(21)	129(57)	15/15
Thres	71(35)	88(23)	100(27)	109(22)	113(13)	120(18)	125(14)	15/15
Diag	**55**(44)	**73**(54)	**88**(57)	**97**(70)	**101**(57)	**107**(67)	**111**(72)	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f3	1739	3600	3609	3636	3642	3646	3651	15/15
CMSA	20(49)	∞	∞	∞	∞	∞	∞ *2e5*	0/15
Thres	33(36)	∞	∞	∞	∞	∞	∞ *2e5*	0/15
Diag	**11**(6)	∞	∞	∞	∞	∞	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f4	2234	3626	3660	3695	3707	3744	28767	12/15
CMSA	60(59)	∞	∞	∞	∞	∞	∞ *2e5*	0/15
Thres	119(128)	∞	∞	∞	∞	∞	∞ *2e5*	0/15
Diag	**38**(40)	∞	∞	∞	∞	∞	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f5	20	20	20	20	20	20	20	15/15
CMSA	**12**(5)	**17**(4)	**17**(12)	**17**(10)	**17**(10)	**17**(8)	**17**(5)	15/15
Thres	14(8)	19(9)	21(8)	21(7)	21(7)	21(8)	21(8)	15/15
Diag	13(7)	17(9)	18(9)	18(5)	18(9)	18(11)	18(7)	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f6	412	623	826	1039	1292	1841	2370	15/15
CMSA	**1.4**(0.2)	3.3(3)	11(29)	14(32)	19(25)	25(17)	163(268)	6/15
Thres	1.8(1)	5.4(1)	7.0(15)	6.9(14)	10(19)	20(56)	30(52)	14/15
Diag	1.6(0.7)	**2.8**(1)	**4.3**(5)	**4.4**(3)	**4.7**(4)	**13**(84)	**21**(45)	13/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f7	172	1611	4195	5099	5141	5141	5389	15/15
CMSA	4.0(3)	**26**(50)	**85**(72)	∞	∞	∞	∞ *2e5*	0/15
Thres	5.7(5)	103(70)	230(313)	∞	∞	∞	∞ *2e5*	0/15
Diag	**2.4**(4)	32(35)	212(409)	**552**(402)	**548**(622)	**548**(467)	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f8	326	921	1114	1217	1267	1315	1343	15/15
CMSA	**3.3**(0.7)	17(8)	18(7)	18(3)	18(7)	19(11)	19(5)	15/15
Thres	8.5(5)	18(12)	18(9)	18(7)	18(7)	18(7)	18(9)	15/15
Diag	6.6(22)	**17**(2)	**18**(5)	**17**(5)	**17**(5)	**18**(4)	**18**(4)	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f9	200	648	857	993	1065	1138	1185	15/15
CMSA	**2.3**(2)	25(13)	24(10)	22(12)	22(10)	21(10)	21(10)	15/15
Thres	4.4(0.8)	**18**(8)	**19**(11)	**18**(7)	**17**(6)	**17**(14)	**17**(7)	15/15
Diag	4.9(2)	35(17)	31(29)	29(20)	27(15)	27(14)	26(17)	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f10	1835	2172	2455	2728	2802	4543	4739	15/15
CMSA	**6.5**(3)	**7.6**(3)	**7.7**(3)	**7.5**(2)	**7.6**(3)	**4.9**(2)	**4.9**(2)	15/15
Thres	6.6(2)	8.3(2)	8.2(0.9)	7.8(1)	8.0(1)	5.2(1)	5.3(1)	15/15
Diag	14(4)	14(3)	14(3)	13(2)	13(2)	8.4(2)	8.3(2)	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f11	266	1041	2602	2954	3338	4092	4843	15/15
CMSA	**14**(3)	6.2(2)	3.2(1.0)	3.4(0.9)	3.5(1)	3.3(0.6)	3.0(1)	15/15
Thres	17(4)	**6.1**(2)	**3.0**(0.7)	**3.0**(1)	**3.0**(1)	**2.9**(0.8)	**2.7**(0.7)	15/15
Diag	84(47)	30(10)	13(3)	12(3)	11(4)	10(2)	9.0(2)	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f12	515	896	1240	1390	1569	3660	5154	15/15
CMSA	**4.2**(10)	**10**(11)	**13**(10)	**15**(15)	**16**(10)	**8.8**(4)	**7.8**(4)	15/15
Thres	13(14)	24(14)	24(11)	24(11)	24(10)	12(5)	12(4)	15/15
Diag	14(39)	29(26)	33(19)	35(6)	34(7)	17(3)	15(6)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f13	387	596	797	1014	4587	6208	7779	15/15
CMSA	15(24)	**19**(10)	**31**(31)	**58**(48)	**28**(32)	**89**(17)	**185**(103)	2/15
Thres	**6.3**(5)	35(86)	56(25)	107(156)	72(89)	461(427)	∞ *2e5*	0/15
Diag	12(7)	52(113)	71(54)	164(232)	117(330)	∞	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f14	37	98	133	205	392	687	4305	15/15
CMSA	1.1(0.5)	2.2(0.5)	2.8(0.5)	3.5(0.9)	4.3(1)	8.7(2)	4.8(3)	15/15
Thres	0.90(0.7)	2.2(1)	**2.6**(0.7)	**3.2**(1)	**4.3**(1)	**8.7**(2)	**4.4**(3)	15/15
Diag	**0.64**(0.6)	**1.8**(0.8)	2.7(0.8)	3.6(0.6)	9.1(2)	23(8)	9.1(3)	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f15	4774	39246	73643	74669	75790	77814	79834	12/15
CMSA	7.0(10)	∞	∞	∞	∞	∞	∞ *2e5*	0/15
Thres	11(14)	∞	∞	∞	∞	∞	∞ *2e5*	0/15
Diag	**5.8**(6)	∞	∞	∞	∞	∞	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f16	425	7029	15779	45669	51151	65798	71570	15/15
CMSA	**1.0**(1)	1.8(4)	**13**(14)	**31**(57)	∞	∞	∞ *2e5*	0/15
Thres	1.1(0.7)	**1.8**(2)	87(89)	64(58)	∞	∞	∞ *2e5*	0/15
Diag	2.6(3)	2.6(3)	27(37)	∞	∞	∞	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f17	26	429	2203	6329	9851	20190	26503	15/15
CMSA	**0.71**(0.5)	**18**(51)	**23**(38)	**30**(16)	**140**(142)	∞	∞ *2e5*	0/15
Thres	39(140)	34(176)	37(63)	67(123)	∞	∞	∞ *2e5*	0/15
Diag	1.1(1)	34(62)	23(44)	58(90)	146(141)	∞	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f18	238	836	7012	15928	27536	37234	42708	15/15
CMSA	68(234)	**129**(263)	**124**(173)	∞	∞	∞	∞ *2e5*	0/15
Thres	88(33)	313(539)	130(104)	**184**(455)	∞	∞	∞ *2e5*	0/15
Diag	**5.1**(16)	189(315)	192(225)	∞	∞	∞	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f19	1	1	10609	9.8e5	1.4e6	1.4e6	1.4e6	15/15
CMSA	18(3)	1.5e5(8e4)	∞	∞	∞	∞	∞ *2e5*	0/15
Thres	19(11)	1.1e5(1e5)	∞	∞	∞	∞	∞ *2e5*	0/15
Diag	**15**(2)	**8.7e4**(1e5)	∞	∞	∞	∞	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f20	32	15426	5.5e5	5.7e5	5.7e5	5.8e5	5.9e5	15/15
CMSA	1.9(0.9)	25(11)	∞	∞	∞	∞	∞ *2e5*	0/15
Thres	**1.7**(0.8)	21(13)	∞	∞	∞	∞	∞ *2e5*	0/15
Diag	2.1(1)	**20**(23)	∞	∞	∞	∞	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f21	130	2236	4392	4487	4618	5074	11329	8/15
CMSA	9.5(18)	23(41)	20(55)	20(14)	19(22)	17(23)	7.8(12)	12/15
Thres	**5.9**(7)	**17**(15)	**15**(19)	**15**(10)	**14**(28)	**13**(20)	**5.9**(6)	13/15
Diag	21(0.6)	19(13)	20(35)	20(27)	19(69)	18(17)	8.0(13)	12/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f22	98	2839	6353	6620	6798	8296	10351	6/15
CMSA	**25**(61)	**6.3**(8)	13(16)	12(23)	12(25)	10(11)	8.1(12)	13/15
Thres	30(40)	8.8(6)	13(13)	12(11)	12(35)	10(13)	8.2(8)	13/15
Diag	60(377)	8.9(8)	**8.8**(16)	**8.8**(12)	**8.8**(20)	**7.8**(10)	**6.5**(11)	14/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f23	2.8	915	16425	1.8e5	2.0e5	2.1e5	2.1e5	15/15
CMSA	1.5(1)	391(326)	∞	∞	∞	∞	∞ *2e5*	0/15
Thres	2.0(3)	313(575)	∞	∞	∞	∞	∞ *2e5*	0/15
Diag	**1.5**(2)	**165**(103)	∞	∞	∞	∞	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f24	98761	1.0e6	7.5e7	7.5e7	7.5e7	7.5e7	7.5e7	1/15
CMSA	**4.8**(5)	∞	∞	∞	∞	∞	∞ *2e5*	0/15
Thres	6.3(12)	∞	∞	∞	∞	∞	∞ *2e5*	0/15
Diag	6.4(7)	∞	∞	∞	∞	∞	∞ *2e5*	0/15

TABLE III

EXPECTED RUNNING TIME (ERT IN NUMBER OF FUNCTION EVALUATIONS) DIVIDED BY THE RESPECTIVE BEST ERT MEASURED DURING BBOB-2009 IN DIMENSION 20. THE ERT AND IN BRACES, AS DISPERSION MEASURE, THE HALF DIFFERENCE BETWEEN 90 AND 10%-TILE OF BOOTSTRAPPED RUN LENGTHS APPEAR FOR EACH ALGORITHM AND TARGET, THE CORRESPONDING BEST ERT IN THE FIRST ROW. THE DIFFERENT TARGET Δf-VALUES ARE SHOWN IN THE TOP ROW. #SUCC IS THE NUMBER OF TRIALS THAT REACHED THE (FINAL) TARGET $f_{\mathrm{opt}} + 10^{-8}$. THE MEDIAN NUMBER OF CONDUCTED FUNCTION EVALUATIONS IS ADDITIONALLY GIVEN IN *italics*, IF THE TARGET IN THE LAST COLUMN WAS NEVER REACHED. ENTRIES, SUCCEEDED BY A STAR, ARE STATISTICALLY SIGNIFICANTLY BETTER (ACCORDING TO THE RANK-SUM TEST) WHEN COMPARED TO ALL OTHER ALGORITHMS OF THE TABLE, WITH $p = 0.05$ OR $p = 10^{-k}$ WHEN THE NUMBER k FOLLOWING THE STAR IS LARGER THAN 1, WITH BONFERRONI CORRECTION BY THE NUMBER OF INSTANCES.

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f1	43	43	43	43	43	43	43	15/15
CMSA	4.9(2)	10(2)	15(2)	19(2)	25(2)	34(2)	45(2)	15/15
Thres	**4.9**(1)	9.5(1)	14(2)	18(2)	**23**(3)	33(3)	**42**(3)	15/15
Diag	4.9(1)	**9.2**(2)	**13**(1)	**18**(2)	23(3)	**33**(2)	42(4)	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f2	385	386	387	388	390	391	393	15/15
CMSA	173(32)	240(38)	265(33)	273(38)	277(34)	285(31)	293(27)	15/15
Thres	154(41)	212(41)	245(25)	259(29)	265(23)	273(26)	282(29)	15/15
Diag	**96**(21)$^{\star 4}$	**113**(14)$^{\star 4}$	**122**(10)$^{\star 4}$	**126**(9)$^{\star 4}$	**128**(7)$^{\star 4}$	**130**(6)$^{\star 4}$	**130**(8)$^{\star 4}$	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f3	5066	7626	7635	7637	7643	7646	7651	15/15
CMSA	∞	∞	∞	∞	∞	∞	∞ *4e5*	0/15
Thres	∞	∞	∞	∞	∞	∞	∞ *4e5*	0/15
Diag	∞	∞	∞	∞	∞	∞	∞ *4e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f4	4722	7628	7666	7686	7700	7758	1.4e5	9/15
CMSA	∞	∞	∞	∞	∞	∞	∞ *4e5*	0/15
Thres	∞	∞	∞	∞	∞	∞	∞ *4e5*	0/15
Diag	∞	∞	∞	∞	∞	∞	∞ *4e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f5	41	41	41	41	41	41	41	15/15
CMSA	**12**(3)	**15**(3)	**15**(6)	**15**(6)	**15**(6)	**15**(6)	**15**(7)	15/15
Thres	13(5)	17(13)	18(12)	18(11)	18(6)	18(11)	18(4)	15/15
Diag	14(8)	17(8)	18(8)	18(8)	18(9)	18(7)	18(10)	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f6	1296	2343	3413	4255	5220	6728	8409	15/15
CMSA	**1.5**(1)	**2.5**(2)	**4.6**(4)	12(22)	34(77)	**80**(138)	331(279)	2/15
Thres	3.2(3)	4.4(4)	5.0(3)	**7.9**(19)	**21**(65)	83(117)	324(190)	2/15
Diag	17(33)	26(32)	32(37)	51(28)	64(69)	135(279)	**204**(252)	3/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f7	1351	4274	9503	16523	16524	16524	16969	15/15
CMSA	622(740)	∞	∞	∞	∞	∞	∞ *4e5*	0/15
Thres	2059(1547)	∞	∞	∞	∞	∞	∞ *4e5*	0/15
Diag	**339**(291)	∞	∞	∞	∞	∞	∞ *4e5*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f8	2039	3871	4040	4148	4219	4371	4484	15/15
CMSA	**11**(4)	**30**(32)	**31**(7)	**31**(52)	**31**(50)	**31**(28)	**31**(47)	13/15
Thres	15(12)	33(55)	34(77)	34(50)	34(25)	34(25)	34(44)	13/15
Diag	28(13)	82(36)	84(104)	85(125)	86(121)	86(47)	85(48)	10/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f9	1716	3102	3277	3379	3455	3594	3727	15/15
CMSA	17(12)	40(68)	41(96)	41(31)	41(5)	40(31)	40(4)	13/15
Thres	**15**(8)	**20**(6)	**22**(8)	**23**(4)	**23**(8)	**23**(2)	**23**(5)	15/15
Diag	36(7)	52(4)	54(36)	56(5)	57(5)	57(4)	57(31)	14/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f10	7413	8661	10735	13641	14920	17073	17476	15/15
CMSA	10(5)	11(2)	9.2(2)	**7.8**(2)	**7.3**(1)	**6.7**(1)	**6.8**(0.6)	15/15
Thres	**8.4**(3)	**10**(3)	**9.1**(1)	7.8(1)	7.4(0.9)	6.8(0.7)	6.8(0.6)	15/15
Diag	33(3)	31(2)	26(2)	20(2)	19(1)	17(2)	17(1)	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f11	1002	2228	6278	8586	9762	12285	14831	15/15
CMSA	**12**(2)$^{\star 2}$	**7.5**(1)	**3.1**(0.4)	**2.6**(0.3)	**2.6**(0.3)	2.5(0.5)	2.5(0.3)	15/15
Thres	15(1)	8.1(0.6)	3.2(0.4)	2.6(0.2)	2.6(0.4)	**2.3**(0.4)	**2.2**(0.5)	15/15
Diag	223(44)	115(28)	43(10)	33(8)	30(5)	24(3)	20(3)	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f12	1042	1938	2740	3156	4140	12407	13827	15/15
CMSA	**2.5**(0.2)	**10**(9)	**13**(8)	**15**(8)*	**14**(6)*	**5.9**(2)	**6.2**(3)	15/15
Thres	14(25)	21(20)	23(9)	25(13)	22(5)	8.6(3)	8.5(3)	15/15
Diag	18(0.2)	96(84)	103(70)	123(125)	113(73)	59(80)	86(109)	5/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f13	652	2021	2751	3507	18749	24455	30201	15/15
CMSA	156(614)	545(891)	2037(1600)	∞	∞	∞	∞ *4e5*	0/15
Thres	156(0.6)	**227**(693)	2037(3199)	1598(2167)	**299**(635)	∞	∞ *4e5*	0/15
Diag	**46**(0.7)	298(346)	**946**(1745)	**1598**(1768)	∞	∞	∞ *4e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f14	75	239	304	451	932	1648	15661	15/15
CMSA	**1.8**(1.0)	1.9(0.4)	2.5(0.6)	3.2(0.3)	**5.2**(0.8)$^{\star 3}$	**11**(1)$^{\star 2}$	4.2(1)	15/15
Thres	1.9(1)	**1.8**(0.4)	**2.3**(0.3)	**3.0**(0.4)	6.8(0.4)	14(1)	**3.7**(0.8)	15/15
Diag	1.8(1)	1.9(1)	2.4(0.2)	3.2(0.4)	15(5)	107(42)	17(8)	14/15

[20] T. J. Fisher and X. Sun, "Improved Stein-type shrinkage estimators for the high-dimensional multivariate normal covariance matrix," *Computational Statistics & Data Analysis*, vol. 55, no. 5, pp. 1909 – 1918, 2011. doi: http://dx.doi.org/10.1016/j.csda.2010.12.006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167947310004743

[21] X. Chen, Z. Wang, and M. McKeown, "Shrinkage-to-tapering estimation of large covariance matrices," *Signal Processing, IEEE Transactions on*, vol. 60, no. 11, pp. 5640–5656, 2012. doi: 10.1109/TSP.2012.2210546

[22] T. Cai and W. Liu, "Adaptive thresholding for sparse covariance matrix estimation," *Journal of the American Statistical Association*, vol. 106, no. 494, pp. 672–684, 2011.

[23] J. Fan, Y. Liao, and M. Mincheva, "Large covariance estimation by thresholding principal orthogonal complements," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 75, no. 4, pp. 603–680, 2013.

[24] R. Ros and N. Hansen, "A simple modification in cma-es achieving linear time and space complexity," in *Parallel Problem Solving from Nature–PPSN X*. Springer, 2008, pp. 296–305.

[25] N. Hansen, "Adaptive encoding: How to render search coordinate system invariant," in *Parallel Problem Solving from Nature – PPSN X*, ser. Lecture Notes in Computer Science, G. Rudolph, T. Jansen, N. Beume, S. Lucas, and C. Poloni, Eds. Springer Berlin Heidelberg, 2008, vol. 5199, pp. 205–214. ISBN 978-3-540-87699-1. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-87700-4_21

[26] M. Pourahmadi, *High-Dimensional Covariance Estimation: With High-Dimensional Data*. John Wiley & Sons, 2013.

[27] J. Fan, Y. Liao, and H. Liu, "An overview on the estimation of large covariance and precision matrices," arXiv:1504.02995.

[28] D. Guillot and B. Rajaratnam, "Functions preserving positive definiteness for sparse matrices," *Transactions of the American Mathematical Society*, vol. 367, no. 1, pp. 627–649, 2015.

[29] N. Hansen, A. Auger, S. Finck, and R. Ros, "Real-parameter black-box optimization benchmarking 2012: Experimental setup," INRIA, Tech. Rep., 2012. [Online]. Available: http://coco.gforge.inria.fr/bbob2012-downloads

[30] S. Finck, N. Hansen, R. Ros, and A. Auger, "Real-parameter black-box optimization benchmarking 2010: Presentation of the noiseless functions," Institute National de Recherche en Informatique et Automatique, Tech. Rep., 2010, 2009/22.