Errata for *Sparse Inverse Covariance Learning for CMA-ES with Graphical Lasso*

Konstantinos Varelas

A second, updated version of the article [1] is presented here, with corrections/additions indicated by the coloured blue text. The following corrections have been done:

- In Section 2.2 (page 3) and Section 2.3 (page 4), the partial correlation matrix is replaced with the absolute partial correlation matrix.
- In line 5 of Algorithm 1 and in line 5 of Algorithm 2, missing absolute values are inserted.

References

[1] Varelas, K., Auger, A. and Hansen, N., 2020, September. Sparse Inverse Covariance Learning for CMA-ES with Graphical Lasso. In International Conference on Parallel Problem Solving from Nature (pp. 707-718). Springer, Cham.

Sparse Inverse Covariance Learning for CMA-ES with Graphical Lasso

Konstantinos Varelas[1,2], Anne Auger[1], and Nikolaus Hansen[1]

[1]Inria, RandOpt team, CMAP, École Polytechnique, France
[2]Thales LAS France SAS, Limours, France
firstname.lastname@inria.fr

Abstract. This paper introduces a variant of the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), denoted as gl-CMA-ES, that utilizes the Graphical Lasso regularization. Our goal is to efficiently solve partially separable optimization problems of a certain class by performing stochastic search with a search model parameterized by a sparse precision, i.e. inverse covariance matrix. We illustrate the effect of the global weight of the l_1 regularizer and investigate how Graphical Lasso with non equal weights can be combined with CMA-ES, allowing to learn the conditional dependency structure of problems with sparse Hessian matrices. For non-separable sparse problems, the proposed method with appropriately selected weights, outperforms CMA-ES and improves its scaling, while for dense problems it maintains the same performance.

1 Introduction

The challenge of estimating covariance or precision, i.e. inverse covariance matrices when the dimension is large, due to the need of large numbers of samples and to the cumulation of significant amounts of estimation errors, leads to the need of discovering efficient high dimensional estimation methods and developing corresponding tools. For this purpose, several approaches have been proposed, often with the assumption of sparsity properties of the matrix to be estimated, which include soft/hard thresholding, l_1 penalization, column-by-column estimation and others [6].

The Covariance Matrix Adaptation Evolution Strategy [9] is a gradient-free continuous optimization algorithm that addresses problems of the form

$$\underset{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^n}{\text{minimize}}\, f(\mathbf{x}) \tag{1}$$

by performing stochastic search in the parameter space $\mathcal{X}$ with an adaptive normal distribution $\mathcal{N}(\mathbf{m}, \sigma^2\mathbf{C})$. The adaptation of the covariance matrix is performed with a learning rate of the order of $\mathcal{O}(1/n^2)$, n being the dimension. Large scale variants which impose different restrictions to the search model have been proposed, in order to increase the adaptation speed [16,2,1,3].

In this paper, we apply one of the high dimensional tools mentioned above, namely the Graphical Lasso [7,4] within CMA-ES, to obtain an algorithm which,

in cases where the Hessian matrix of the objective function has a sparse structure, exploits this property. Eventually, the degrees of freedom of the search model, and thus the adaptation time, can be reduced without deteriorating the performance of CMA-ES.

The paper is organised as follows: in section 2 we describe the fundamentals of CMA-ES as well as of estimation using the Graphical Lasso. In section 3, we introduce the novel algorithm and in section 4 we test the method in selected examples with known sparse structure. Finally, in section 5 we summarize and discuss future steps for further improvement of our approach.

2 Graphical Lasso and CMA-ES

The adaptation of the search distribution in CMA-ES is based on principles of invariance. Through sampling candidate solutions from $\mathcal{N}(\mathbf{m}, \sigma^2\mathbf{C})$ and ranking the corresponding f values, CMA-ES estimates a better distribution via weighted recombination, which eventually generates solutions with improved f-values.

In cases where the objective function f in (1) is convex and quadratic, strong empirical evidence indicate that the search covariance matrix approximates the inverse Hessian of f up to a scalar. This property allows, after adapting $\mathbf{C}$, to optimize any convex quadratic function with the same convergence rate as the Sphere function $f(\mathbf{x}) = \|\mathbf{x}\|_2^2$, and efficiently address ill-conditioned problems.

2.1 Partial separability, Hessian sparsity and conditionally independent models

In large-scale optimization, parsimony becomes essential, either for representing a Hessian matrix, or for developing update techniques and learning the matrix economically. The notion of partial separability [15,8] becomes useful in this case: f is called partially separable if it can be decomposed in a form:

$$f(\mathbf{x}) = \sum_{i=1}^{p} f_i(\mathbf{x}) \tag{2}$$

where each f_i has a large invariant subspace. We recall that the invariant subspace $\mathcal{N}_h$ of a function $h : \mathbb{R}^n \to \mathbb{R}$ is defined as the largest subspace of $\mathbb{R}^n$ such that for all vectors $\mathbf{y} \in \mathcal{N}_h$ and for all $\mathbf{x} \in \mathbb{R}^n$, the relation $h(\mathbf{x} + \mathbf{y}) = h(\mathbf{x})$ holds. In cases where each term f_i is a function of few search coordinates, this property is obviously satisfied (even though partial separability is in general a weaker assumption). If in addition f_i are twice differentiable, the Hessian matrix of f will be sparse.

Motivated by the above observations, we propose a method which is developed based on the CMA-ES principles and performs stochastic search with a distribution with conditionally independent coordinates. Our goal is to reduce if possible the number of free parameters of the search model, thus increase the adaptation speed, without deteriorating the performance of CMA-ES. In the case of a smooth objective function f, this would mean to exploit sparsity of the Hessian. However, no assumption is imposed on the regularity of f.

2.2 Graphical Lasso

The Graphical Lasso was introduced to estimate distributions with a sparse precision, i.e. inverse covariance, matrix. With this property, one introduces parametric models with conditionally independent search coordinates, a procedure also known as covariance selection [5]. In particular, if Σ is the sample estimation of a covariance matrix, the solution of

$$\underset{X \in \mathbb{S}^n_{++}}{\text{minimize}} \operatorname{tr}(\Sigma X) - \log\det X + \alpha \|X\|_1 \tag{3}$$

provides the sparse model estimation, where X represents the precision matrix to be estimated, $\mathbb{S}^n_{++}$ is the set of symmetric positive definite $n \times n$ matrices and the penalty factor α controls the tradeoff between the log-likelihood and the penalization term $\|X\|_1 = \sum_{i,j=1}^n |X_{ij}|$. For $\alpha = 0$, the solution X^* of (3) is $X^* = \Sigma^{-1}$, since the Kullback-Leibler divergence of the distributions parameterized by X and Σ is decomposed as:

$$\begin{aligned} D_{\mathrm{KL}}(\mathcal{N}(\mathbf{0}, \Sigma) || \mathcal{N}(\mathbf{0}, X^{-1})) &= \frac{1}{2}\left(\operatorname{tr}(\Sigma X) - n + \log\left(\frac{\det X^{-1}}{\det \Sigma}\right)\right) \\ &= \frac{1}{2}\left(\operatorname{tr}(\Sigma X) - \log\det X\right) - \frac{1}{2}\left(n + \log\det \Sigma\right) \end{aligned} \tag{4}$$

The l_1 penalization in (3) is employed to force sparsity on the precision matrix X, or equivalently on the absolute partial correlation matrix $|\mathrm{diag} X^{-1/2} X \mathrm{diag} X^{-1/2}|$, and can be viewed as a convex relaxation of the number of non zero entries of X, $\mathbf{Card}(X)$ (which makes (3) a NP-hard problem [4]).

In a black-box scenario where the sparsity structure is unknown, estimating the precision matrix with the Graphical Lasso serves exactly the purpose of discovering this structure. In the context of CMA-ES, in order to learn sparse search models, the candidate solutions are generated from the regularized distribution that solves (3) and the original update rules are used. In practice, the Graphical Lasso is applied to standardized variables, thus when solving (3) Σ is the correlation matrix provided by the CMA-ES update.

2.3 Equal weights and effect on conditioning

Problem (3) imposes the same penalization factor α on all precision entries, and the alternative regularization term $\alpha\|X^-\|_1 = \alpha \sum_{i \neq j} |X_{ij}|$, which penalizes only the off-diagonal entries, has been proposed e.g. in [14]. This kind of penalization leads to a consistent reduction of the axes length ratios learned by CMA-ES after the regularization step, as illustrated in Figure 1 for a 4 dimensional block diagonal case.

Recently, tools for an extension of (3) with non-equal penalization factors, i.e. for solving:

$$\underset{X \in \mathbb{S}^n_{++}}{\text{minimize}} \operatorname{tr}(\Sigma X) - \log\det X + \sum_{i,j=1}^{n} \alpha_{ij} |X_{ij}| \tag{5}$$

with selected $\alpha_{ij} \geq 0$ have been developed [12]. In the following, this formulation is used along with a simple rule for selecting the penalty factors in order to surpass the above effect: precision entries are penalized only if the corresponding absolute partial correlations, i.e. the entries of $|\mathrm{diag}X^{-1/2}X\mathrm{diag}X^{-1/2}|$, are below a threshold τ.

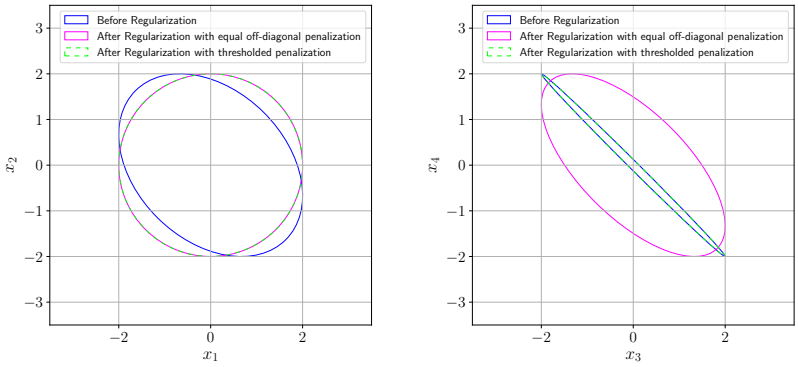

Fig. 1. Equal-density ellipses of the marginal distributions of (x_1, x_2) and (x_3, x_4) before and after regularization with off-diagonal only and with thresholded penalty factors. The random vector $\mathbf{x} = (x_1, \ldots, x_4)$ is distributed as $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ where $\mathbf{C}$ is a block-diagonal covariance matrix of the form $\mathbf{C} = \left(\begin{smallmatrix} \mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 \end{smallmatrix}\right), \mathbf{C}_1, \mathbf{C}_2$ being of size 2×2.
The solid magenta line illustrates the effect of regularization when only off-diagonal elements are penalized with the same factor α. The factor value is chosen as the minimal value that achieves an isotropic distribution for the pair of weakly dependent variables (x_1, x_2) (left, with an axis ratio $\sqrt{2}$), i.e. $\alpha = 1/3$ in this particular example. The search distribution of the strongly dependent pair (x_3, x_4) (right, with an axis ratio $\sqrt{1000}$) is drastically affected. The dashed line (green) corresponds to thresholded regularization according to (5). In this case, only the precision matrix entry corresponding to the pair (x_1, x_2) is penalized, i.e. the chosen factors are: $\alpha_{ij} = 1/3$ if $(i, j) \in \{(1, 2), (2, 1)\}$ else $\alpha_{ij} = 0$.

3 Algorithm

In this section we introduce the proposed algorithm, denoted as gl-CMA-ES. It only uses recombination with positive weights for the update of the covariance matrix, in order to ensure its positive definiteness. The differences with respect

to the original CMA-ES setting with positive recombination weights [1] are highlighted in Algorithm 1, while Algorithm 2 describes the regularization step. The minimization step in line 6 of Algorithm 2 is solved using [12].

For reasons of stability, and since the number of degrees of freedom for the covariance matrix is $n(n+1)/2$, the corresponding learning rate in CMA-ES is of the order of $\mathcal{O}(1/n^2)$. In other large scale variants of CMA-ES, e.g. the Separable CMA-ES [16], the degrees of freedom of the search model are reduced and the adaptation is performed faster. Similarly, in our approach the learning rates depend on the number of non zero entries of the Lasso estimated precision matrix, ranging from $\mathcal{O}(1/n)$ for sparse to $\mathcal{O}(1/n^2)$ for dense matrices. Furthermore, limited memory methods have been proposed [11,13], aiming at reducing the internal space and time complexity of CMA-ES. Such methods, though, do not exploit properties such as separability in order to accelerate the convergence [11,17].

The algorithm coincides with CMA-ES if the threshold is chosen as $\tau = 0$, that is if the l_1 penalization is not applied. If this holds, the sampling matrix $\mathbf{C}_{t+1}^{\text{reg}}$ is equal to $\mathbf{C}_t$ in line 4 of Algorithm 1, thus the candidate solutions are sampled from $\mathcal{N}(\mathbf{m}_t, \sigma_t^2 \mathbf{C}_t)$, see lines 9 and 10. Additionally, the evolution path for the adaptation of the step size follows the same update rule as in CMA-ES, see line 14 of Algorithm 1. The learning rates c_1, c_μ are defined in a compatible way with CMA-ES in line 6, when the precision matrix is fully dense, i.e. when $n_z = n^2$.

Note that the invariance property to strictly monotonic transformations of the objective function f that CMA-ES possesses is maintained in the algorithm. However, invariance to affine transformations of the search space breaks when regularization is applied, i.e. when setting $\tau > 0$.

4 Results

We present experimental results on representative problems included in Table 1, in order to verify whether the proposed approach is able to identify the correct sparse structure of the objective function's Hessian matrix. All experiments were performed with an initial step size $\sigma_0 = 1$ and with a starting point $\mathbf{x}_0$ defined in Table 1.

The first test function f_{ellisub} is constructed by composing the Ellipsoid function f_{elli} with a rotational search space transformation as defined in Table 1. This results in a non-separable ill-conditioned problem with maximal sparsity, since the upper triangular part of the Hessian of f_{ellisub} has exactly one non zero entry. Figure 2 presents the gain in convergence speed in terms of number of function evaluations of gl-CMA-ES compared to the default CMA-ES (with positive recombination weights). It also shows the performance scaling with dimension,

[1] An extension of CMA-ES, called Active CMA-ES [10], that performs recombination with both positive and negative weights using all sampled solutions has been proposed.

Algorithm 1. gl-CMA-ES

Set parameters: $\lambda = 4 + \lfloor 3 \ln n \rfloor$, $\mu = \lfloor \lambda/2 \rfloor$, $w_i = \frac{\ln(\mu+\frac{1}{2}) - \ln i}{\sum_{j=1}^{\mu} \ln(\mu+\frac{1}{2}) - \ln j}$ for $i = 1 \dots \mu$, $\mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i^2}$, $c_\sigma = \frac{\mu_w+2}{n+\mu_w+3}$, $d_\sigma = 1 + 2\max\{0, \sqrt{\frac{\mu_w-1}{n+1}} - 1\} + c_\sigma$, $c_c = \frac{4+\mu_w/n}{n+4+2\mu_w/n}$,
Initialize: $\mathbf{p}_t^c \leftarrow \mathbf{0}$, $\mathbf{p}_t^\sigma \leftarrow \mathbf{0}$, $\mathbf{C}_t \leftarrow \mathbf{I}$, $t \leftarrow 0$, τ,
while termination criteria not met **do**
  $\mathbf{C}_{t+1}^{\text{reg}} \leftarrow \text{REGULARIZE}(\mathbf{C}_t, \tau)$,
  $n_z \leftarrow \#|\mathbf{C}_{t+1}^{\text{reg}\ -1}| > 0$,
  $c_1 \leftarrow \frac{2}{(n_z/n+1.3)(n+1.3)+\mu_w}, c_\mu \leftarrow \min\{1-c_1, 2\frac{\mu_w+1/\mu_w-1.75}{(n_z/n+2)(n+2)+\mu_w}\}$,

  for $k \leftarrow 1, \dots, \lambda$ **do**
    $\mathbf{z}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_{t+1}^{\text{reg}})$
    $\mathbf{x}_k \leftarrow \mathbf{m}_t + \sigma_t \mathbf{z}_k$
    $f_k \leftarrow f(\mathbf{x}_k)$
  end for
  $\mathbf{m}_{t+1} \leftarrow \sum_{k=1}^{\mu} w_k \mathbf{x}_{k:\mu}$
  $\mathbf{p}_{t+1}^\sigma \leftarrow (1-c_\sigma)\,\mathbf{p}_t^\sigma + \sqrt{c_\sigma(2-c_\sigma)\mu_w}\,\mathbf{C}_{t+1}^{\text{reg}\ -\frac{1}{2}} \frac{\mathbf{m}_{t+1}-\mathbf{m}_t}{\sigma_t}$,
  $h_\sigma \leftarrow \mathbb{1}_{\|\mathbf{p}_{t+1}^\sigma\| < (1.4+\frac{2}{n+1})\sqrt{1-(1-c_\sigma)^{2(t+1)}}\,\mathbb{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\|}$,
  $\delta(h_\sigma) \leftarrow (1-h_\sigma)c_c(2-c_c)$,
  $\mathbf{p}_{t+1}^c \leftarrow (1-c_c)\,\mathbf{p}_t^c + h_\sigma \sqrt{c_c(2-c_c)\mu_w}\,\frac{\mathbf{m}_{t+1}-\mathbf{m}_t}{\sigma_t}$,
  $\mathbf{C}_{t+1}^\mu \leftarrow \sum_{k=1}^{\mu} w_k \mathbf{z}_{k:\mu} \mathbf{z}_{k:\mu}^T$,
  $\mathbf{C}_{t+1} \leftarrow (1 + c_1\delta(h_\sigma) - c_1 - c_\mu)\mathbf{C}_t + c_1 \mathbf{p}_{t+1}^c \mathbf{p}_{t+1}^{c\ T} + c_\mu \mathbf{C}_{t+1}^\mu$,
  $\sigma_{t+1} \leftarrow \sigma_t \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_{t+1}^\sigma\|}{\mathbb{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\|} - 1\right)\right)$,
  $t \leftarrow t+1$
end while

Algorithm 2. Regularization

function REGULARIZE($\mathbf{C}$, τ) ▷ $\mathbf{C}$ is a covariance matrix
  $\tilde{\mathbf{C}} \leftarrow \text{diag}\mathbf{C}^{-1/2}\mathbf{C}\text{diag}\mathbf{C}^{-1/2}$,
  $\mathbf{P} \leftarrow \tilde{\mathbf{C}}^{-1}$
  $\tilde{\mathbf{P}} \leftarrow \text{diag}\mathbf{P}^{-1/2}\mathbf{P}\text{diag}\mathbf{P}^{-1/2}$,
  $\mathbf{W}_{ij} \leftarrow 1$ if $|\tilde{\mathbf{P}}_{ij}| < \tau$ else 0
  $\mathbf{P}_{\text{reg}} \leftarrow \arg\min_{\mathbf{\Theta} \in \mathbb{S}_{++}^n} \text{tr}(\tilde{\mathbf{C}}\mathbf{\Theta}) - \log\det\mathbf{\Theta} + \sum_{i,j=1}^{n} \mathbf{W}_{ij}|\theta_{ij}|$ ▷ Initialized at $\mathbf{P}$
  $\tilde{\mathbf{C}}_{\text{reg}} \leftarrow \mathbf{P}_{\text{reg}}^{-1}$
  return $\text{diag}\mathbf{C}^{1/2}\tilde{\mathbf{C}}_{\text{reg}}\text{diag}\mathbf{C}^{1/2}$
end function

Name	Definition	$\mathbf{x}_0$
Sphere	$f_{\text{sphere}}(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	$3 \cdot \mathbf{1}$
Ellipsoid	$f_{\text{elli}}(\mathbf{x}) = \sum_{i=1}^{n} 10^{6\frac{i-1}{n-1}} x_i^2$	$3 \cdot \mathbf{1}$
Cigar	$f_{\text{cig}}(\mathbf{x}) = x_1^2 + \sum_{i=2}^{n} 10^{6\frac{i-1}{n-1}} x_i^2$	$3 \cdot \mathbf{1}$
Tablet	$f_{\text{tab}}(\mathbf{x}) = 10^6 x_1^2 + \sum_{i=2}^{n} x_i^2$	$3 \cdot \mathbf{1}$
Twoaxes	$f_{\text{twoax}}(\mathbf{x}) = \sum_{i=1}^{\lfloor n/2 \rfloor} 10^6 x_i^2 + \sum_{i=\lfloor n/2 \rfloor+1}^{n} x_i^2$	$3 \cdot \mathbf{1}$
Subspace Rotated Ellipsoid	$f_{\text{ellisub}}(\mathbf{x}) = \sum_{i=2}^{n-1} 10^{6\frac{i-1}{n-1}} x_i^2 + \begin{pmatrix} x_1 & x_n \end{pmatrix} \mathbf{R}^T \begin{pmatrix} 1 & 0 \\ 0 & 10^6 \end{pmatrix} \mathbf{R} \begin{pmatrix} x_1 \\ x_n \end{pmatrix}$	$3 \cdot \mathbf{1}$
Rosenbrock	$f_{\text{rosen}}(\mathbf{x}) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	$\mathbf{0}$
2-Blocks Ellipsoid	$f_{\text{2-blocks elli}}(\mathbf{x}) = f_{\text{elli}}(\mathbf{Bx})$	$3 \cdot \mathbf{1}$
2-Blocks Cigar	$f_{\text{2-blocks cig}}(\mathbf{x}) = f_{\text{cig}}(\mathbf{Bx})$	$3 \cdot \mathbf{1}$
2-Blocks Tablet	$f_{\text{2-blocks tab}}(\mathbf{x}) = f_{\text{tab}}(\mathbf{Bx})$	$3 \cdot \mathbf{1}$
Permuted 2-Block Rotated Ellipsoid	$f_{\text{perm ellisub}}(\mathbf{x}) = f_{\text{elli}}(\mathbf{P}_2\mathbf{B}\mathbf{P}_1\mathbf{x})$	$3 \cdot \mathbf{1}$
Rotated Ellipsoid	$f_{\text{ellirot}}(\mathbf{x}) = f_{\text{elli}}(\mathbf{Qx})$	$3 \cdot \mathbf{1}$
k-Rotated Quadratic	$f_{k\text{-rot}}(\mathbf{x}) = \sum_{i=1}^{n-k} x_i^2 + \begin{pmatrix} x_{n-k+1} & \cdots & x_n \end{pmatrix} \mathbf{R}_k^T \begin{pmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & 10^6 \end{pmatrix} \mathbf{R}_k \begin{pmatrix} x_{n-k+1} \\ \vdots \\ x_n \end{pmatrix}$	$3 \cdot \mathbf{1}$

Table 1. Benchmark functions. The matrix $\mathbf{R}$ ($\mathbf{R}_k$) is a random 2×2 ($k \times k$) rotation matrix drawn from the Haar distribution in SO(2) (SO(k) respectively). The block diagonal matrix $\mathbf{B}$ has the form $\mathbf{B} = \left(\begin{smallmatrix} \mathbf{B}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 \end{smallmatrix}\right)$, where $\mathbf{B}_1$ and $\mathbf{B}_2$ are random rotation matrices of size $\frac{n}{2} \times \frac{n}{2}$, $\mathbf{Q}$ is a random rotation matrix of size $n \times n$ and $\mathbf{P}_1, \mathbf{P}_2$ are random permutation matrices.

compared to other large scale variants of CMA-ES, namely the Separable CMA-ES [16], the VkD-CMA-ES [1] and the dd-CMA-ES [3], as well as with the Active CMA-ES [10], i.e. the algorithm that uses the entire sample population additionally with negative recombination weights.

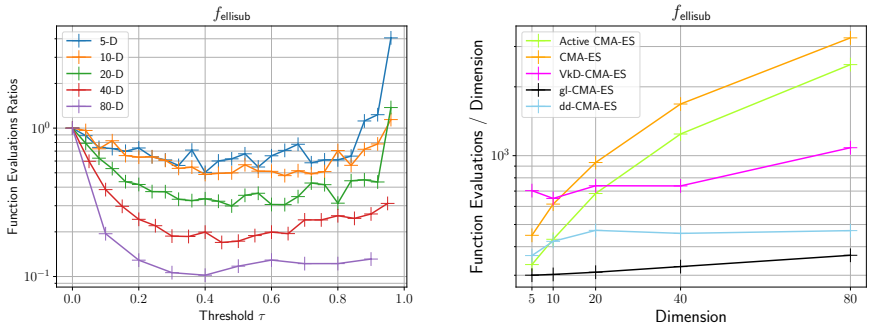

Fig. 2. Left: Ratios of number of function evaluations performed in single runs to reach a precision 10^{-10} close to the global optimal f value of the proposed approach over CMA-ES depending on the regularization threshold τ. One run has been performed per each value of τ. Right: Scaling with dimension of the average number of function evaluations to reach a 10^{-10} precision. The average is taken over 10 independent runs of each algorithm. The chosen threshold value is $\tau = 0.4$.

The second test case is the non-convex Rosenbrock function $f_{\text{rosen}}(\mathbf{x}) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$, for which the Hessian matrix is (globally) tridiagonal. Figure 3 presents the speed-up obtained for different values of the threshold parameter τ and the scaling with dimension. In dimension $n = 5$ the convergence speed is almost the same with the speed of CMA-ES, while in dimension $n = 80$, the method becomes more than 3 times faster. The conditional dependency graphs learned by the proposed approach for 2 different values of τ are shown in Figure 4.

Furthermore, we illustrate the learned conditional dependency pattern for a test function where the number of non zero entries of the Hessian is quadratic with n. In particular, we define $f_{\text{perm ellisub}}(\mathbf{x}) = f_{\text{elli}}(\mathbf{P}_2\mathbf{B}\mathbf{P}_1\mathbf{x})$, where $\mathbf{P}_1, \mathbf{P}_2$ are random permutation matrices and $\mathbf{B}$ a 2-block diagonal rotation matrix, see also Table 1. Figure 5 presents the graph that corresponds to the true Hessian sparsity pattern and the final conditional dependency graph resulting from gl-CMA-ES.

The next example is the function $f_{k-\text{rot}}$, defined in Table 1, which results from an ill-conditioned separable function after performing a (random) rotation in a $k-$dimensional subspace of $\mathbb{R}^n$. This forms a group of k strongly dependent search coordinates (with high probability) and the Hessian's sparsity decreases with increasing k. Figure 6 illustrates the convergence speed for different thresh-

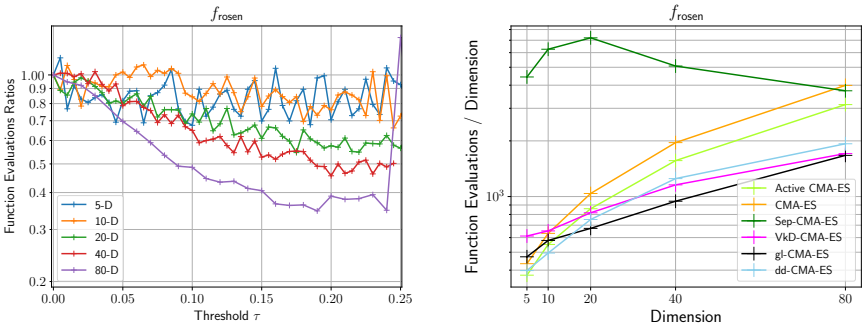

Fig. 3. Left: Ratios of number of function calls performed in single runs to reach a precision of 10^{-10} of the proposed approach over CMA-ES. Right: Performance scaling for $\tau = 0.24$ with averaging over 10 independent runs.

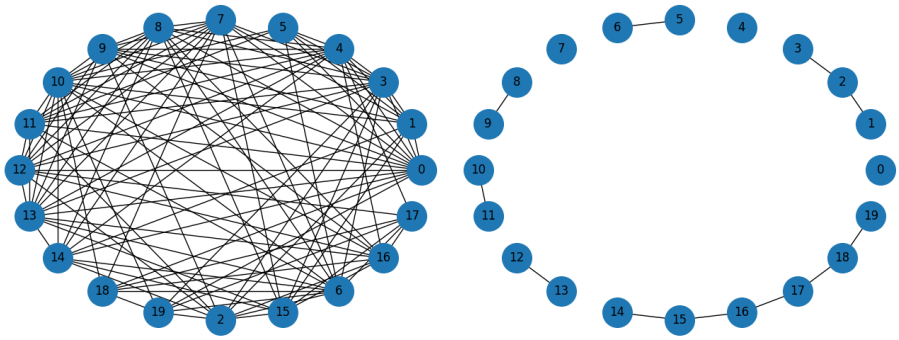

Fig. 4. Conditional dependency graphs in the last iteration of a single run for thresholds $\tau = 0.05$ (left) and $\tau = 0.24$ (right) on the 20-dimensional Rosenbrock function. Edges depict non zero off-diagonal entries of the precision matrix.

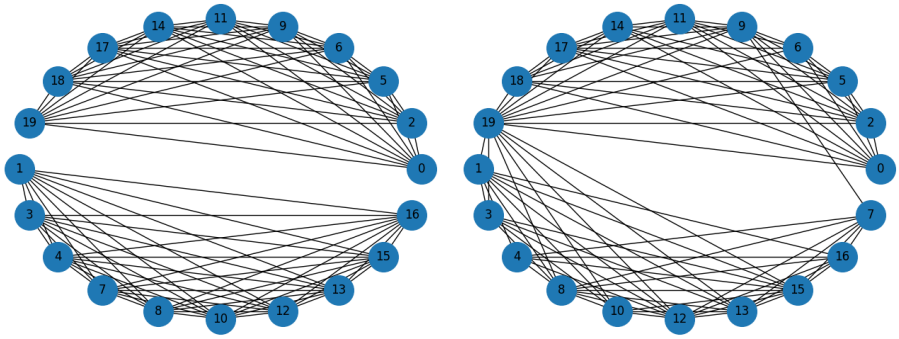

Fig. 5. Adjacency graph of the true Hessian matrix (left) and conditional dependency graph in the last iteration of a single run of gl-CMA-ES with $\tau = 0.1$ on $f_{\text{perm ellisub}}$.

old values and for varying values of k. Threshold values between 0.3 and 0.5 reveal similar and close to optimal performance.

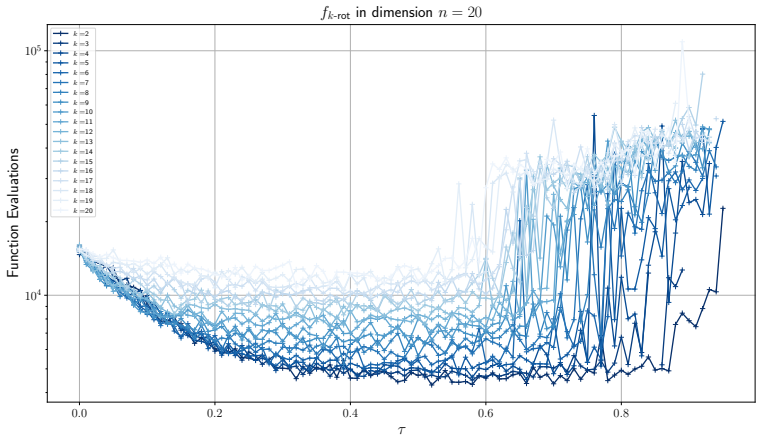

Fig. 6. Number of function evaluations performed by single runs of gl-CMA-ES to reach the global optimum of $f_{k-\mathrm{rot}}$, for different values of k versus the threshold τ. The sparsity of the objective function's Hessian is determined by the block size k. Missing values of the number of function evaluations (typically for large threshold values, which lead to an axis-parallel search model) correspond to single runs where gl-CMA-ES does not reach the optimum within a precision of 10^{-10}. The maximal gain in convergence speed is observed when the sparsity is maximal, i.e. for $k = 2$.

Finally, the performance scaling on the rest of the benchmark functions of Table 1 is shown in Figure 7 for selected threshold parameter values, chosen after preliminary experimentation in a way that the estimated precision's sparsity pattern is not less rich than the Hessian's true pattern. Separable problems allow a choice of a large value for τ and we obtain a behaviour very similar to Separable CMA-ES. Also, the scaling of the method on sparse non-separable problems such as the $f_{2\text{-blocks tablet}}$ and $f_{2\text{-blocks elli}}$ functions is advantageous over all other methods, with the exception of dd-CMA-ES, which shows better scaling on the latter function, due to less conservative learning rate values compared to gl-CMA-ES. For both functions, in dimension $n = 6$, gl-CMA-ES and CMA-ES differ by a factor smaller than 1.3 and in dimension $n = 80$, gl-CMA-ES is more than twice as fast as CMA-ES. One exception is the $f_{2\text{-blocks cigar}}$ function, where all other methods outperform gl-CMA-ES, for dimensions $n \geq 10$. This is also the only case of worse performance compared to CMA-ES indicating that the choice of τ is too large. On fully dense non-separable problems such as the

rotated Ellipsoid function f_{ellirot}, the value $\tau = 0$ reduces to the default setting of CMA-ES and the performance is identical.

5 Discussion

We proposed a method of l_1 regularization within CMA-ES, attempting to increase the adaptation speed of the search distribution. We investigated its behaviour and showed the gain in convergence speed in representative sparse problems and for selected values of the threshold parameter.

The setting of the threshold is crucial for the richness of the search model and thus for the performance of gl-CMA-ES, and good choices depend on properties of the function to be optimized, for example separability. As a result, a future step for improving this approach is the inclusion of an adaptive mechanism for this parameter, rather than being predefined and static.

Furthermore, in this study we only investigate the performance gain in terms of convergence speed through accelerating the covariance matrix adaptation. No focus has been given on the internal cost of the regularization step using the Graphical Lasso. A possible strategy would be to perform this step in the updated search distribution once every a certain number of iterations while sampling with the same regularized search model in between, in cases where the dimension is large and the computational cost becomes a burden.

Finally, in order to guarantee the positive definiteness of the covariance matrix, only positive recombination weights are used, as mentioned in the algorithm's description. Therefore, another interesting aspect and future step for improvement is to employ negative recombination weights.

Acknowledgement The PhD thesis of Konstantinos Varelas is funded by the French MoD DGA/MRIS and Thales Land & Air Systems.

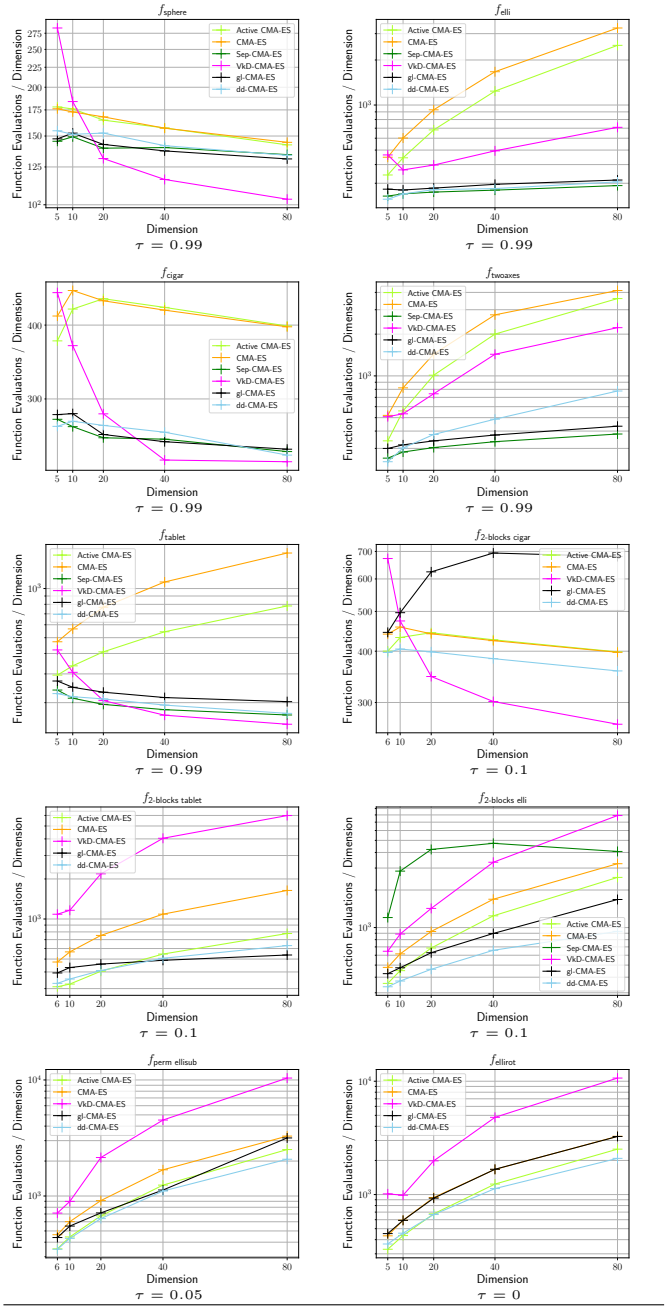

Fig. 7. Scaling on benchmark functions for selected thresholds. The performance measure is the number of function evaluations to reach a target precision 10^{-10} close to the global optimal f value. Average is taken over 10 independent runs of each method.

References

1. Akimoto, Y., Hansen, N.: Online model selection for restricted covariance matrix adaptation. In: Parallel Problem Solving from Nature (PPSN 2016). pp. 3–13. Springer (2016)
2. Akimoto, Y., Hansen, N.: Projection-based restricted covariance matrix adaptation for high dimension. In: Genetic and Evolutionary Computation Conference (GECCO 2016). pp. 197–204. Denver, United States (Jul 2016)
3. Akimoto, Y., Hansen, N.: Diagonal acceleration for covariance matrix adaptation evolution strategies. Evolutionary computation pp. 1–31 (2019)
4. d'Aspremont, A., Banerjee, O., El Ghaoui, L.: First-order methods for sparse covariance selection. SIAM Journal on Matrix Analysis and Applications 30(1), 56–66 (2008)
5. Dempster, A.P.: Covariance selection. Biometrics pp. 157–175 (1972)
6. Fan, J., Liao, Y., Liu, H.: An overview of the estimation of large covariance and precision matrices. The Econometrics Journal 19(1), C1–C32 (2016)
7. Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics 9(3), 432–441 (2008)
8. Griewank, A., Toint, P.: On the unconstrained optimization of partially separable functions. In: Nonlinear Optimization 1981, pp. 301–312. Academic press (1982)
9. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evol. Comput. 9(2), 159–195 (2001)
10. Jastrebski, G.A., Arnold, D.V.: Improving evolution strategies through active covariance matrix adaptation. In: 2006 IEEE international conference on evolutionary computation. pp. 2814–2821. IEEE (2006)
11. Knight, J.N., Lunacek, M.: Reducing the space-time complexity of the cma-es. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation. pp. 658–665. GECCO '07, Association for Computing Machinery, New York, NY, USA (2007), https://doi.org/10.1145/1276958.1277097
12. Laska, J., Narayan, M.: skggm 0.2.7: A scikit-learn compatible package for Gaussian and related Graphical Models (Jul 2017), https://doi.org/10.5281/zenodo.830033
13. Loshchilov, I.: LM-CMA: an alternative to L-BFGS for large scale black-box optimization. Evolutionary Computation 25, 143–171 (2017)
14. Mazumder, R., Hastie, T.: Exact covariance thresholding into connected components for large-scale graphical lasso. The Journal of Machine Learning Research 13(1), 781–794 (2012)
15. Nocedal, J., Wright, S.: Numerical optimization. Springer Science & Business Media (2006)
16. Ros, R., Hansen, N.: A simple modification in CMA-ES achieving linear time and space complexity. In: Parallel Problem Solving from Nature (PPSN 2008). pp. 296–305. Springer (2008)
17. Varelas, K., Auger, A., Brockhoff, D., Hansen, N., ElHara, O.A., Semet, Y., Kassab, R., Barbaresco, F.: A comparative study of large-scale variants of CMA-ES. In: International Conference on Parallel Problem Solving from Nature. pp. 3–15. Springer (2018)