SGAI

AI
2015

Max Bramer
Miltos Petridis *Editors*

# Research and Development in Intelligent Systems XXXII

## Incorporating Applications and Innovations in Intelligent Systems XXIII

Proceedings of AI-2015,
The Thirty-Fifth SGAI International
Conference on Innovative Techniques
and Applications of Artificial
Intelligence

Springer

# Research and Development in Intelligent Systems XXXII

## Incorporating Applications and Innovations in Intelligent Systems XXIII

Max Bramer · Miltos Petridis
Editors

# Research and Development in Intelligent Systems XXXII

## Incorporating Applications and Innovations in Intelligent Systems XXIII

Proceedings of AI-2015, The Thirty-Fifth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence

Springer

*Editors*

Max Bramer
School of Computing
University of Portsmouth
Portsmouth
UK

Miltos Petridis
School of Computing, Engineering
and Mathematics
University of Brighton
Brighton
UK

ISBN 978-3-319-25030-4 ISBN 978-3-319-25032-8 (eBook)
DOI 10.1007/978-3-319-25032-8

Library of Congress Control Number: 2015952537

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media
(www.springer.com)

# Programme Chairs' Introduction

This volume comprises the refereed papers presented at AI-2015, the Thirty-fifth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, held in Cambridge in December 2015 in both the technical and the application streams. The conference was organised by SGAI, the British Computer Society Specialist Group on Artificial Intelligence.

The technical papers included present new and innovative developments in the field, divided into sections on Knowledge Discovery and Data Mining, Machine Learning and Knowledge Acquisition, and AI in Action. This year's Donald Michie Memorial Award for the best refereed technical paper was won by a paper entitled "Sparse Covariance Matrix Adaptation Techniques for Evolution Strategies" by S. Meyer-Nieberg and E. Kropat (Universität der Bundeswehr, München, Germany).

The application papers included present innovative applications of AI techniques in a number of subject domains. This year, the papers are divided into sections on Applications of Genetic Algorithms, Applications of Intelligent Agents and Evolutionary Techniques, and AI Applications. This year's Rob Milne Memorial Award for the best refereed application paper was won by a paper entitled "Development of Temporal Logic-Based Fuzzy Decision Support System for Diagnosis of Acute Rheumatic Fever/Rheumatic Heart Disease" by Sanjib Raj Pandey, Jixin Ma and Chong Hong Lai (University of Greenwich, UK).

The volume also includes the text of short papers presented as posters at the conference.

On behalf of the conference organising committee we would like to thank all those who contributed to the organisation of this year's programme, in particular the programme committee members, the executive programme committees and our administrators Mandy Bauer and Bryony Bramer.

Max Bramer, Technical Programme Chair, AI-2015.

Miltos Petridis, Application Programme Chair, AI-2015.

# Acknowledgements/Committees

## AI-2015 Conference Committee

Prof. Max Bramer, University of Portsmouth (Conference Chair)
Prof. Max Bramer, University of Portsmouth (Technical Programme Chair)
Prof. Miltos Petridis, University of Brighton (Application Programme Chair)
Dr. Jixin Ma, University of Greenwich (Deputy Application Programme Chair)
Prof. Adrian Hopgood, Sheffield Hallam University (Workshop Organiser)
Rosemary Gilligan, University of Hertfordshire (Treasurer)
Dr. Nirmalie Wiratunga, The Robert Gordon University (Poster Session Organiser)
Andrew Lea, Primary Key Associates Ltd. (AI Open Mic and Panel Session Organiser)
Giovanna Martinez, Nottingham Trent University and Christo Fogelberg, Palantir Technologies (FAIRS 2015)
Prof. Miltos Petridis, University of Brighton and Prof. Thomas Roth-Berghofer, University of West London (UK CBR Organisers)
Mandy Bauer, BCS (Conference Administrator)
Bryony Bramer (Paper Administrator)

## Technical Executive Programme Committee

Prof. Max Bramer, University of Portsmouth (Chair)
Dr. John Kingston, Tribal Group
Prof. Thomas Roth-Berghofer, University of West London
Dr. Nirmalie Wiratunga, Robert Gordon University, Aberdeen

## Applications Executive Programme Committee

Prof. Miltos Petridis, University of Brighton (Chair)
Mr. Richard Ellis, Helyx SIS Ltd
Ms. Rosemary Gilligan, University of Hertfordshire
Dr. Jixin Ma, University of Greenwich (Vice-Chair)
Dr. Richard Wheeler, University of Edinburgh

## Technical Programme Committee

Andreas Albrecht, Middlesex University
Yaxin Bi, University of Ulster
Mirko Boettcher, University of Magdeburg, Germany
Max Bramer, University of Portsmouth
Krysia Broda, Imperial College, University of London
Ken Brown, University College Cork
Frans Coenen, University of Liverpool
Madalina Croitoru, University of Montpellier, France
Bertrand Cuissart, Université de Caen
Ireneusz Czarnowski, Gdynia Maritime University, Poland
Stefan Diaconescu, Softwin, Romania
Nicolas Durand, Aix-Marseille University
Frank Eichinger, CTS EVENTIM AG & Co. KGaA, Hamburg, Germany
Sandra Garcia Esparza, University College Dublin, Ireland
Adriana Giret, Universidad Politecnica de Valencia
Nadim Haque
Arjen Hommersom, University of Nijmegen, The Netherlands
Zina Ibrahim, Kings College, London, UK
John Kingston, Tribal Group
Konstantinos Kotis, University of Piraeus
Ivan Koychev, University of Sofia
Fernando Lopes, LNEG-National Research Institute, Portugal
Roberto Micalizio, Università di Torino
Daniel Neagu, University of Bradford
Lars Nolle, Jade University of Applied Sciences, Germany
Dan O'Leary, University of Southern California
Juan Jose Rodriguez, University of Burgos
Maria Dolores Rodriguez-Moreno, Universidad de Alcal
Thomas Roth-Berghofer, University of West London
Fernando Saenz-Perez, Universidad Complutense de Madrid
Miguel A. Salido, Universidad Politecnica de Valencia
Rainer Schmidt, University Medicine of Rostock, Germany

Sid Shakya, BT TSO—Research & Technology
Frederic Stahl, University of Reading
Simon Thompson, BT Innovate
Jon Timmis, University of York
M.R.C. van Dongen, University College Cork
Graham Winstanley, University of Brighton
Nirmalie Wiratunga, Robert Gordon University

## Application Programme Committee

Hatem Ahriz, Robert Gordon University
Tony Allen, Nottingham Trent University
Ines Arana, Robert Gordon University
Mercedes Arguello, University of Salford
Ken Brown, University College Cork
Richard Ellis, Helyx SIS Ltd
Roger Evans, University of Brighton
Rosemary Gilligan, University of Hertfordshire
John Gordon, AKRI Ltd
Chris Hinde, Loughborough University
Adrian Hopgood, De Montfort University
Stelios Kapetanakis, University of Brighton
Jixin Ma, University of Greenwich
Lars Nolle, Jade University of Applied Sciences
Miltos Petridis, University of Brighton
Miguel A. Salido, Universidad Politecnica de Valencia
Roger Tait, University of Cambridge
Richard Wheeler, Edinburgh Scientific
Patrick Wong, Open University

# Contents

**Research and Development in Intelligent Systems XXXII**

**Best Technical Paper**

**Knowledge Discovery and Data Mining**

**Machine Learning and Knowledge Acquisition**

**AI in Action**

**Short Papers**

**Applications and Innovations in Intelligent Systems XXIII**

**Best Application Paper**

**Applications of Genetic Algorithms**

**Applications of Intelligent Agents and Evolutionary Techniques**

**AI Applications**

## Short Papers

# Research and Development in Intelligent Systems XXXII

# Best Technical Paper

# Sparse Covariance Matrix Adaptation Techniques for Evolution Strategies

**Silja Meyer-Nieberg and Erik Kropat**

**Abstract** Evolution strategies are variants of evolutionary algorithms. In contrast to genetic algorithms, their search process depends strongly on mutation. Since the search space is often continuous, evolution strategies use a multivariate normal distribution as search distribution. This necessitates the tuning and adaptation of the covariance matrix. Modern evolution strategies apply covariance matrix adaptation mechanisms in order to achieve this end. However, the covariance estimation is conducted with small sample sizes compared to the search space dimensionality. Regarding the agreement of sample estimate and true covariance, this represents a potential problem. This paper introduces a new approach by changing the coordinate systems and implements several sparse covariance matrix techniques. The results are evaluated in experiments.

## 1 Introduction

Evolution strategies (ESs) are a variant of evolutionary algorithms often used for continuous black-box optimization. They differ from many other evolutionary algorithms in the role of mutation: While it is only a background operator in genetic algorithms, it represents the main search operator here. Evolution strategies operate with a multivariate normal distribution which is used to generate a population of new search points. Its parameters, the mean and the covariance matrix, must be updated during a run in order for the strategy to reach the vicinity of optimal points fast and reliably. The adaptation of the parameters takes the search history and the present population into account. Due to its importance, research focussed and focusses on the covariance matrix. The main techniques introduced are based on the sample covariance matrix [1]. The usage of this estimator may bear potential improvement points

S. Meyer-Nieberg (✉) · E. Kropat
Department of Computer Science, Universität der Bundeswehr München,
Werner-Heisenberg Weg 39, 85577 Neubiberg, Germany
e-mail: silja.meyer-nieberg@unibw.de

E. Kropat
e-mail: erik.kropat@unibw.de

M. Bramer and M. Petridis (eds.), *Research and Development in Intelligent Systems XXXII*, DOI 10.1007/978-3-319-25032-8_1

within itself: Evolution strategies typically operate with small population or sample sizes. The size of the population does not exceed the search space dimensionality. Estimating the $N \times N$ dimensional covariance matrix with a sample size of $\mu < N$ or $\mu \approx N$ leads to unreliable estimates. All adaptation techniques introduced so far consider correction terms. However, the question remains whether an ES may benefit if techniques developed for and tailored to the task at hand were introduced.

Literature concerning attempts of combining evolutionary algorithms or related approaches with statistical estimation methods of high-dimensional covariance matrices is scarce. So far, we were only able to identify two approaches aside from our own research: In the first [6], the authors investigated estimation of distribution algorithms (EDAs) for continuous spaces. The EDA applied a Gaussian search distortion similar to evolution strategies. The estimation of the covariance matrix resulted however in matrices that were not positive definite. To circumvent the problem, a shrinkage procedure was introduced, see e.g. [13]. Very recently, a shrinkage estimator was integrated into an evolution strategy variant with a single search point [12].

The research presented here is part of an ongoing investigation into alternative estimation techniques for high-dimensional covariances [15, 16]. In [15, 16] Ledoit-Wolf shrinkage estimators were analyzed. While the results were promising, finding the appropriate shrinkage intensity represented a challenge. Therefore, in [14] another computational simple estimation method was introduced: thresholding. Here the work begun in [14] is continued by addressing two of open problems remaining: The first concerns the choice of the thresholding function, the latter the influence of an important parameter of the thresholding.

The paper is structured as follows. First, the evolution strategy variant considered in this paper is introduced. Afterwards, we argue why high-dimensional estimation techniques might improve the performance. The next section introduces the sparse covariance estimation evolution strategy developed. An experimental analysis of the approaches follows, before the paper closes with an outlook on potential future research.

## 2 Evolution Strategies

Evolution strategies (ESs) [18, 19] are used for continuous black-box optimization $f : \mathbb{R}^N \to \mathbb{R}$. Several variants have been introduced (see e.g. [1, 3]). In many cases, a population of $\mu$ parents is used to create a set of $\lambda$ offspring, with $\mu \leq \lambda$. Like all evolutionary algorithms, evolution strategies operate in a sequence of generations. In each generation, the same cycle of processes is carried out. In general, these are parent selection, recombination, mutation, and survivor selection. In the following, the processes are described based on the ES variant considered. Here, all $\mu$ parents contribute to create the offspring. First recombination is performed, that is, the centroid of the parents is computed [3]. All offspring are based on the same origin and differ only in their mutation vector, a normally distributed random variable with zero mean and covariance matrix $\sigma^2\mathbf{C}$ which is added to the mean. After the $\lambda$ offspring $\mathbf{y}_1, \ldots, \mathbf{y}_\lambda$ have been created, the individuals are evaluated. In most cases, the function to be optimized is used directly. In that case, the function is also called fitness. Selection in evolution strategies takes often only the $\lambda$ offspring into account of which the $\mu$ best $\mathbf{y}_{1:\lambda}, \ldots, \mathbf{y}_{\mu:\lambda}$ are chosen.

The most important factor concerning the mutation is the covariance matrix. It must be adapted during the run and fitted to the landscape. Otherwise, the performance may be low. Therefore, research on controlling the mutation has a long tradition in ESs. First approaches were already considered in [18]. The next section describes the variant considered in this paper.

### 2.1 Covariance Matrix Adaptation: The Population Covariance

To our knowledge, covariance matrix adaptation comprises two main classes: one applied in the *covariance matrix adaptation evolution strategy* (CMA-ES) [11] and an alternative used in the *covariance matrix self-adaptation evolution strategy* (CMSA-ES) [4]. Both are based on a variant of the sample covariance, correcting the estimate with information from the search history. The present paper focuses on the CMSA-ES leaving the CMA-ES for future research. One of the reasons is that the CMSA-ES does only include one additional correction term making it easier to assess the effects of the thresholding operator. The CMSA-ES considers the covariance matrix $(\sigma^{(g)})^2\mathbf{C}^{(g)}$ with $\sigma^{(g)}$ denoting general scaling factor (or step-size or mutation strength) and with $\mathbf{C}^{(g)}$ 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 CMSA uses covariance matrix adaptation for the matrix $\mathbf{C}^{(g)}$ and self-adaptation for the mutation strength.

The covariance matrix update is based upon the common estimate of the covariance matrix using the newly created population. However, the sample consists of the selected parents and not of the complete set. Restricting the sample, shall induce a bias towards promising parts of the search space. Since the adaptation of the mutation strength happens separately, the sample is normalized with $\mathbf{z}_{m:\lambda}^{(g+1)} := \left(\mathbf{x}_{m:\lambda}^{(g+1)} - \mathbf{m}^{(g)}\right)/\sigma^{(g)}$ before estimating the covariance, see also [11]. Since the centroid used for the mutation is known, the covariance matrix estimation does not need to re-estimate the mean. The rank-$\mu$ update then obtains the covariance matrix as

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

which is usually a positive semi-definite matrix since $\mu \ll N$. The weights $w_m$ should fulfill $w_1 \geq w_2 \geq \ldots \geq w_m$ with $\sum_{m=1}^{\mu} w_i = 1$. While it is possible to consider unequal weights, the CMSA-ES usually operates with $w_m = 1/\mu$. 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 (1) with the sample covariance and the old covariance

$$\mathbf{C}^{(g+1)} := (1 - \frac{1}{c_\tau})\mathbf{C}^{(g)} + \frac{1}{c_\tau}\mathbf{C}_\mu^{(g+1)} \tag{2}$$

with the weights usually set to $w_m = 1/\mu$ and following [4]

$$c_\tau = 1 + \frac{N(N+1)}{2\mu}. \tag{3}$$

### 2.2 Step-Size Adaptation

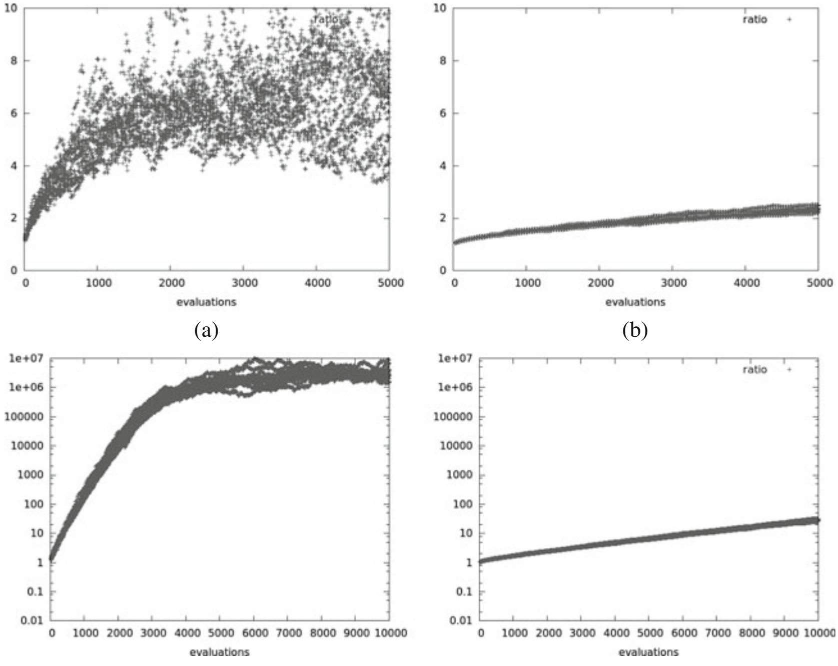
The CMSA implements the step-size using *self-adaptation* first introduced in [18] and developed further in [19]. Here, evolution is used to tune the strategy parameters of the mutation process. In other words, these parameters undergo recombination, mutation, and indirect selection processes. The working 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 $\sigma_l^{(g)} = \sigma^{(g)}\exp(\tau\mathcal{N}(0,1))$ is used for the mutation of the mutation strength. The parameter $\tau$, the *learning rate*, should scale with $1/\sqrt{2N}$. The CMSA-ES often uses recombination. Among others, self-adaptation with recombination improves the performance in the presence of noise [2]. While the recombination of the mutation strength could be realized in several ways, it normally follows the recombination of the objective values in computing the mean of the mutation strengths of the parents. The newly created mutation strength $\sigma_l^{(g)}$ is then used for mutating the objective values of the offspring. If the resulting offspring is sufficiently good, the scale factor is passed to the next generation.

## 3 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.

The sample covariance (1) has a strong influence on the adaptation. However, the good properties of the maximum likelihood estimator hold for the case $\mu \gg N$ and $\mu \gg 1$. In evolution strategies, the sample size seldom exceeds the search space dimension with $\mu < N$. For example, [9] recommends to use $\lambda = \lfloor \log(3N) \rfloor + 4$ offspring and to set the size of the parent population to $\mu = \lfloor \lambda/2 \rfloor$. Thus, a potential problem arises in high-dimensional settings. For $N \to \infty$, we have $\mu/N \to 0$ contradicting the assumptions on which the estimator was based.

In order to assess the problem in evolution strategies, we take a closer look at the eigenvalues of the covariance matrix for some selected functions. Figure 1 shows the development of the ratio of the largest to the smallest eigenvalue of the covariance matrix on the sphere $f(\mathbf{x}) = \|\mathbf{x}\|^2$ and on the discus $f(\mathbf{x}) = 10^6 x_1^2 + \sum_{i=2}^{N} x_i^2$. In the latter case it can be argued that the behavior observed is beneficial. For the sphere, the figures hint at a potential problem: The gap between largest and smallest eigenvalue widens for all runs with the problem being more pronounced for the smaller search space dimensionalities. Furthermore, the extremely small sample size for $N = 10$ causes a large variation between the runs. It is interestingly less distinct in the case of the higher dimensional search spaces. This is probably an effect of the parameter $c_\tau$ which follows $\lim_{N\to\infty} c_\tau(N) = \infty$ as long as $\mu \propto \log(N)$ or $\mu \propto N$. Thus, the influence of the population covariance lessens. In statistics the problem is well-known [20, 21] with a long research tradition concerning approaches to counteract the problematic properties, see e.g. [17] for an overview. Among others, it has been shown that the eigenstructures of the estimate and the covariance do not agree well.



(a)

(b)

(c)

(d)

**Fig. 1** The development of the ratio of the largest to the smallest eigenvalue of the covariance for the CMSA-ES on the sphere and the discus. Shown are the results from 15 runs for each dimensionality. **a** Sphere, N = 10. **b** Sphere, N = 40. **c** Discus, N = 10. **d** Discus, N = 40

### 3.1 Space Transformation

Several types of estimators assume a sparse structure of the covariance matrix. Shortly stated, these approaches work well if many entries are small or even zero. Then, computationally simple estimation techniques can be applied. In the case of evolution strategies, a sparseness assumption may not hold in every situation. The form of the covariance matrix depends strongly on the function landscape and may vary widely in practice. Furthermore, there may not be any information available concerning the type of the function itself. Therefore, the covariance matrix is not considered in the original space but in the eigenspace of the previous covariance matrix $\mathbf{C}^{(g)}$.

Let the covariance matrix $\mathbf{C}^{(g)}$ be a symmetric, positive definite $N \times N$ matrix. The condition holds for the original adaptation since (2) combines a positive definite with a positive semi-definite matrix. As we will see below, in the case of thresholding the assumption may not always be fulfilled. Let $\mathbf{v}_1, \ldots, \mathbf{v}_N$ denote the $N$ eigenvectors with the eigenvalues $\lambda_1, \ldots, \lambda_N$, $\lambda_j > 0$. The definiteness of $\mathbf{C}^{(g)}$ guarantees their existence. 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 $i \neq j$. Define $\mathbf{V} := (\mathbf{v}_1, \ldots, \mathbf{v}_N)$. 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 $\Lambda^{(g)} = \mathbf{V}\mathbf{C}^{(g)}\mathbf{V}^{\mathrm{T}}$ with $\Lambda^{(g)}$ a diagonal matrix containing the eigenvalues. Diagonal matrices are sparse, 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 have been introduced. For instance, the number of non-zero elements in a row may not exceed a predefined limit $s_0(N) > 0$, i.e., $\max_i \sum_{j=1}^{N} \delta(|a_{ij}| > 0) \leq s_0(N)$, which should grow only slowly with $N$. This definition can, however, be relaxed to a more general definition of sparseness, also referred to as approximate sparseness [5] on which the adaptive thresholding considered is based. Applying thresholding in our case requires that the true covariance matrix of the selected set has an approximately sparse structure in the eigenspace of $\mathbf{C}^{(g)}$. Assuming the validity of the assumption, we change the coordinate system in order to perform the covariance matrix estimation. Reconsider the normalized (aside from the covariance matrix) mutation vectors $\mathbf{z}_{1;\lambda}, \ldots, \mathbf{z}_{\mu;\lambda}$ that were associated with the $\mu$ best offspring. Denoting their representation in the eigenspace as $\hat{\mathbf{z}}_{m;\lambda} = \mathbf{V}^{\mathrm{T}}\mathbf{z}_{m;\lambda}$ for $m = 1, \ldots, \mu$ leads to the new population covariance

$$\hat{\mathbf{C}}_\mu = \sum_{i=1}^{\mu} w_i \hat{\mathbf{z}}_{m;\lambda} \hat{\mathbf{z}}_{m;\lambda}^{\mathrm{T}} \tag{4}$$

which is used to derive the final estimate. In the next section, potential techniques for sparse covariance matrices are discussed.

### 3.2 Sparse Covariance Matrix Estimation

Several methods have been developed for estimating sparse covariance matrices: Among others banding, tapering, and thresholding can be applied, see e.g. [17]. While all three are based on the assumption that many entries of the true covariance are zero, banding and tapering assume an ordering of the variables which is not present in the case of evolution strategies.

Therefore, only thresholding remains. Thresholding discards entries which are smaller than a given threshold $\varepsilon > 0$. For a matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, the thresholding operator $T_\varepsilon(\mathbf{A})$ is defined as

$$T_\varepsilon(\mathbf{A}) := (a_{ij}\delta(|a_{ij}| \geq \varepsilon))_{N \times N} \tag{5}$$

with $\delta(\cdot) = 1$ if the condition is fulfilled and zero otherwise. The choice of the threshold is critical for the quality of the resulting estimate. Equation (5) represents an example of universal thresholding with a hard thresholding function. Soft thresholding is also common, examples of this function class comprise e.g.

$$s_\lambda(x) = \text{sign}(x)(|x| - \lambda)_+ \quad \text{(soft-thresholding)} \tag{6}$$

$$s_\lambda(x) = |x|(1 - |\frac{\lambda}{x}|^\eta)_+ \qquad \text{(Lasso)} \tag{7}$$

with $(x)_+ := \max(0, x)$. Adaptive thresholding which considers the current data for determining the threshold $\lambda_{ij}$ appears as more appropriate for evolution strategies than using constant thresholds. Following [5], we use

$$\lambda_{ij} := \lambda_{ij}(\delta) = \delta\sqrt{\frac{\hat{\theta}_{ij}\log N}{\mu}} \tag{8}$$

where $\delta > 0$ can be either chosen as a constant or be obtained using cross-validation. The variable $\hat{\theta}_{ij}$ in (8) is determined as $\hat{\theta}_{ij} = \frac{1}{\mu}\sum_{m=1}^{\mu}[(\hat{z}_{mi} - \overline{Z^i})(\hat{z}_{mj} - \overline{Z^j}) - \hat{c}^{\mu}_{ij}]^2$ with $\hat{c}^{\mu}_{ij}$ denoting the $(i, j)$-entry of $\hat{\mathbf{C}}^{(g+1)}_{\mu}$, $\hat{z}_{mi}$ the $i$th component of $\hat{\mathbf{z}}_{m:\lambda}$, and $\overline{Z^i} := (1/\mu)\sum_{m=1}^{\mu}\hat{z}_{mi}$.

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. Due to this potential problem, future research will investigate repair mechanisms as well as alternative thresholding functions, see e.g. [7]. Here, the soft-thresholding (6) and the Lasso thresholding function (7) are considered. While it is common to exclude the diagonal entries of the covariance from thresholding, this may not be always appropriate for optimization since the nature of the functions may vary widely. Our previous experiments did not show a clear advantage for either method. Therefore, both versions are taken into account. In combination with the thresholding function, the following four

ES types are investigated: (1) CMSA-Thres-ES (abbreviated to Thres): an evolution strategy with CMSA which applies thresholding in the eigenspace of the covariance with soft-thresholding, (2) CMSA-ThresL-ES (abbreviated to ThresL): the same as above but using the Lasso thresholding, (3) CMSA-Diag-ES (abbreviated to Diag): an ES with covariance matrix adaptation with thresholding in the eigenspace of the covariance, preserving the diagonal elements, and using soft-thresholding, (4) CMSA-DiagL-ES (abbreviated to DiagL): the variation with the Lasso function.

# 4 Experiments

Two series of experiments were conducted: The first with the aim to gain more insight regarding the choice of the parameter $\delta$. Our first approach was to make this parameter data dependent by setting it to $\delta = 2\max(\hat{\mathbf{C}}_\mu)$. Since [5] recommends to use either $\delta = 2$ or to conduct cross-validation, we performed a short experimental analysis and took a closer look at the development of the eigenvalues on the sphere and on the discus. We considered the $\delta = 2, 3$, and 4 for the CMSA-ThresL-ES with the search space dimensionalities set to $N = 10, 20, 40$, and 100.

The second series of experiments compares the different shrinkage variants with the original CMSA-ES. Two thresholding operators, soft thresholding and Lasso thresholding (with $\eta = 4$), are taken into account. The comparison is based on the search space dimensions $N = 10$ and 20. The second series of experiments uses a maximal number of fitness evaluations of $\mathrm{FE}_{\max} = 2 \times 10^5\, N$. While the experiments revealed that longer experiments are necessary in order to derive meaningful findings for the difficult multimodal functions, the task was delegated to future research because of the computing time required.

All strategies start from randomly chosen positions, sampled uniformly from the interval $[-4, 4]^N$. The ESs used $\lambda = \lfloor \log(3N) + 8 \rfloor$ offspring and $\mu = \lceil \lambda/4 \rceil$ parents. An equal setting of weights $w_m$ was used with $w_m = 1/\mu$. 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 conducted. In order not to waste resources, a run is restarted when a search stagnation is observed. The latter is characterized by observing changes of the best values below $10^{-8}$ over the last $10 + \lceil 30N/\lambda \rceil$ generations.

## *4.1 Test Suite Und Performance Measure*

The algorithms were implemented in MATLAB. The paper uses the black box optimization benchmarking (BBOB) software framework and test suite, see [10]. The framework[1] can be used to benchmark and compare continuous black-box optimiz-

[1] Current software and tutorials under http://coco.gforge.inria.fr.

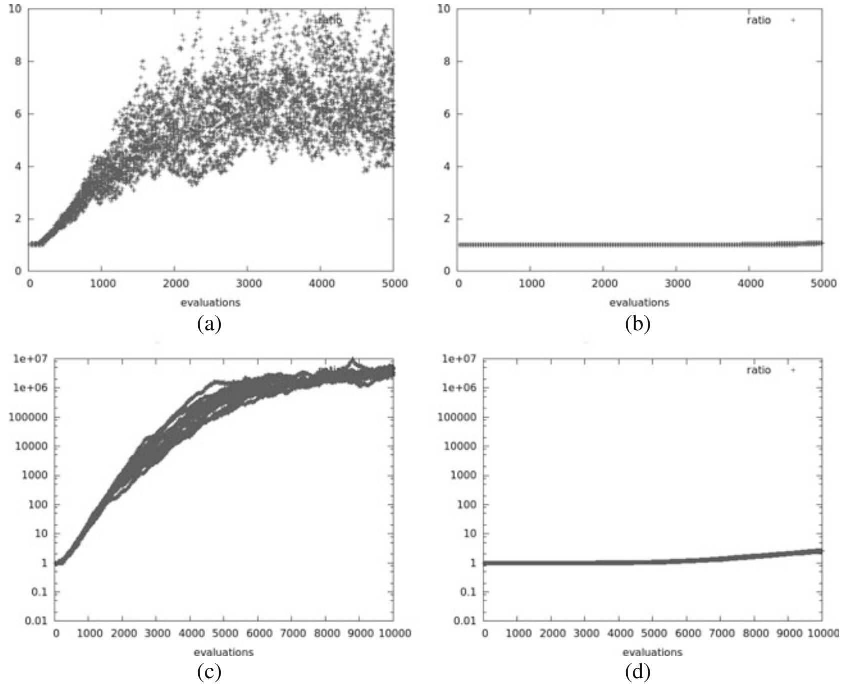
ers and provides easy means to generate tables and figures. This paper considers the 24 noiseless functions of the test suite [8]. In order to lower the possibility that an algorithm benefits from initialisation effects, the position of the optimum is changed from run to run. The test suite comprises four function classes which differ in the degree of difficulty they pose for the optimization: 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), with the last comprising functions with a weak global structure.

Following [10], the expected running time (ERT) is used as the performance measure. It is defined as the expected value of the function evaluations ($f$-evaluations) required to reach the target value with the required precision for the first time, see [10]. In this paper, ERT $= \frac{\#(FEs(f_{\text{best}} \geq f_{\text{target}}))}{\#succ}$ is used as an estimate. It is obtained by summing up the evaluations $FEs(f_{\text{best}} \geq f_{\text{target}})$ in each run until the fitness of the best individual is smaller than the target value, divided by the number of all successful runs.

## 4.2 Results and Discussion

First, we describe the results from the parameter dependency experiments. The thresholding should on the one hand "stabilize" the covariance matrix in the sense that the eigenvalues do not diverge unless of course it is required to optimize the function. On the other hand, it should not delay or prohibit the adaptation of the covariance matrix to the function space. Summarizing the effects of operating with a data independent $\delta$ from $\{2, 3, 4\}$, this detrimental behavior can be observed. Thus, Fig. 2 only shows the development of the ratio of the largest and the smallest eigenvalue for the sphere and the discus for two exemplary search space dimensions using the data dependent $\delta$. Comparing Figs. 2 to 1 reveals that for the sphere the variation between the runs is reduced even for the smaller search space. In the case of the discus, the increase of the ratio is slower, which could result in slower convergence.

The findings for the BBOB test suite indicate advantages for thresholding in many cases. The outcome 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 for 10D and 20D. Concerning the particular functions, differences are revealed as Tables 1 and 2 show for the expected running time (ERT) provided for several precision targets. In the case of the sphere (function with id 1) and the separable ellipsoid (id 2), all strategies reach the final precision goal in all runs. For both functions, ESs with thresholding are the fastest. In the case of the sphere, preserving the diagonal elements appears slightly advantageous, however, all variants are close together. For the ellipsoid, the gap widens. Interestingly, two variants remain close together: the CMSA-Thres-ES and the CMSA-DiagL-ES which differ in the thresholding function as well as in the decision whether to subject the diagonal entries to thresholding or not. No strategy reaches the required target precision in the case of the separable



(a) (b) (c) (d)

**Fig. 2** Development of the ratio of the largest and the smallest eigenvalue of the covariance matrix of the CMSA-ThresL-ES on the sphere. Shown are the results from 15 repeats for each dimensionality. **a** Sphere, N = 10. **b** Sphere, N = 40. **c** Discus, N = 10. **d** Discus, N = 40

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 and due to page restrictions the data are removed from the tables. In the case of the linear slope (id 5) all strategies are successful. While the thresholding variants perform better for the smaller search space probably due to the more stable behavior of the covariance adaptation, the advantage is lost for $N = 20$ as Table 2 shows.

In the case of the functions with low to moderate conditioning (id 6–9), the step ellipsoid with id 7 is the most difficult function to optimize for the ESs. Experiments with a larger number of maximal function evaluations will be performed in future research. In the case of the remaining functions, we see a separation between the attractive sector (id 6) and the Rosenbrock variants (ids 8 and 9). In the case of the attractive sector and $N = 10$, see Table 1, the original CMSA-ES could only reach the required target precision in eight of the 15 runs, whereas the thresholding variants resulted only in one or two unsuccessful runs. Increasing the search space dimensionality, causes all runs of the CMSA-ES to be unsuccessful while the thresholding variants still achieve two or three successful runs. On the original Rosenbrock function (id 8), the CMSA-Thres-ES with the soft-thresholding function is the worst performing strategy. For $N = 20$, Table 2, the CMSA-DiagL-ES which uses the Lasso also

**Table 1** Expected running time (ERT in number of function evaluations) divided by the respective best ERT measured during BBOB-2009 in dimension 10

| $\Delta f_{\text{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ | $\Delta f_{\text{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **f1** | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 15/15 | **f11** | 266 | 1041 | 2602 | 2954 | 3338 | 4092 | 4843 | 15/15 |
| CMSA | 4.4(2) | 10(5) | 15(6) | 21(7) | 26(6) | 38(9) | 51(3) | 15/15 | CMSA | 17(12) | 6.7(3) | 3.3(2) | 3.7(2) | 3.8(1) | 3.6(1) | 3.4(0.8) | 15/15 |
| Thres | 3.7(1) | 8.3(1) | 14(2) | 19(2) | 24(3) | 35(6) | 46(4) | 15/15 | Thres | 108(22) | 36(10) | 16(4) | 15(4) | 13(3) | 11(3) | 10(5) | 15/15 |
| Diag | 3.9(1) | 8.5(4) | 13(2) | **18**(3) | 23(5) | **33**(6) | 45(8) | 15/15 | Diag | 19(6) | 7.2(2) | 3.6(1.0) | 3.7(1) | 3.7(0.8) | 3.4(0.7) | 3.0(0.7) | 15/15 |
| DiagL | **3.2**(2) | **7.5**(1) | **12**(2) | 18(3) | **23**(3) | 34(3) | **44**(3) | 15/15 | DiagL | 84(28) | 30(8) | 13(6) | 12(2) | 11(2) | 10(2) | 9.0(3) | 15/15 |
| ThresL | 4.2(3) | 9.2(3) | 14(3) | 18(3) | 24(5) | 35(5) | 46(5) | 15/15 | ThresL | **17**(2) | **6.1**(2) | **3.0**(1) | **3.0**(0.7) | **3.0**(0.6) | **2.9**(1) | **2.7**(1) | 15/15 |
| $\Delta f_{\text{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ | $\Delta f_{\text{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ |
| **f2** | 187 | 190 | 191 | 191 | 193 | 194 | 195 | 15/15 | **f12** | 515 | 896 | 1240 | 1390 | 1569 | 3660 | 5154 | 15/15 |
| CMSA | 70(33) | 85(38) | 95(30) | 104(54) | 109(40) | 117(53) | 120(54) | 15/15 | CMSA | **4.7**(8) | **13**(9) | **15**(8) | **17**(7) | **17**(7) | **10**(4) | **9.2**(4) | 15/15 |
| Thres | **41**(32) | **64**(56) | **78**(63) | **87**(46) | **91**(59) | **97**(62) | **101**(62) | 15/15 | Thres | 30(50) | 36(30) | 37(22) | 39(16) | 37(16) | 18(3) | 14(5) | 15/15 |
| Diag | 67(43) | 103(20) | 124(48) | 133(44) | 141(24) | 149(53) | 153(46) | 15/15 | Diag | 7.4(16) | 16(19) | 18(15) | 19(16) | 20(14) | 11(5) | 9.2(4) | 15/15 |
| DiagL | 55(31) | 73(49) | 88(61) | 97(56) | 101(62) | 107(56) | 111(63) | 15/15 | DiagL | 14(20) | 29(27) | 33(10) | 35(11) | 34(8) | 17(5) | 15(10) | 15/15 |
| ThresL | 71(23) | 88(23) | 100(21) | 109(25) | 113(19) | 120(19) | 125(20) | 15/15 | ThresL | 13(15) | 24(16) | 24(14) | 24(8) | 24(12) | 12(5) | 12(7) | 15/15 |
| $\Delta f_{\text{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ | $\Delta f_{\text{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ |
| **f5** | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 15/15 | **f13** | 387 | 596 | 797 | 1014 | 4587 | 6208 | 7779 | 15/15 |
| CMSA | 12(4) | 17(4) | 18(13) | 18(11) | 18(10) | 18(9) | 18(10) | 15/15 | CMSA | 8.2(20) | **22**(19) | **43**(74) | **65**(101) | **27**(31) | **106**(53) | **377**(559) | 1/15 |
| Thres | **11**(2) | **16**(8) | **16**(4) | **16**(4) | **17**(8) | **17**(3) | **17**(8) | 15/15 | Thres | **4.8**(7) | 30(19) | 73(124) | 219(320) | 135(171) | ∞ | ∞ *2e5* | 0/15 |
| Diag | 15(3) | 23(9) | 24(14) | 24(11) | 24(10) | 24(13) | 24(13) | 15/15 | Diag | 7.4(7) | 45(65) | 67(48) | 124(235) | 74(63) | 236(129) | 382(154) | 1/15 |
| DiagL | 13(4) | 17(6) | 18(4) | 18(6) | 18(11) | 18(11) | 18(8) | 15/15 | DiagL | 12(23) | 52(168) | 71(53) | 164(104) | 117(155) | ∞ | ∞ *2e5* | 0/15 |
| ThresL | 14(4) | 19(6) | 21(6) | 21(9) | 21(11) | 21(7) | 21(10) | 15/15 | ThresL | 6.3(13) | 35(27) | 56(83) | 107(117) | 72(88) | 461(540) | ∞ *2e5* | 0/15 |
| $\Delta f_{\text{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ | $\Delta f_{\text{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ |
| **f6** | 412 | 623 | 826 | 1039 | 1292 | 1841 | 2370 | 15/15 | **f14** | 37 | 98 | 133 | 205 | 392 | 687 | 4305 | 15/15 |
| CMSA | 2.0(0.9) | 2.9(5) | 5.3(17) | 7.7(9) | 13(28) | 21(17) | 112(104) | 8/15 | CMSA | 1.2(0.8) | 2.2(0.9) | 2.9(1.0) | 3.2(1) | **4.1**(1) | 10(5) | 5.5(8) | 15/15 |
| Thres | **1.3**(0.5) | **2.7**(3) | 5.1(1) | 8.0(10) | 13(6) | 24(26) | 35(91) | 13/15 | Thres | 1.1(0.9) | 2.1(0.8) | 2.7(0.5) | **3.2**(0.7) | 10(6) | 30(16) | 14(8) | 15/15 |
| Diag | 2.2(2) | 3.8(1) | **4.2**(5) | 6.0(8) | 13(19) | 23(40) | 43(21) | 12/15 | Diag | 1.3(1) | 2.8(0.6) | 3.1(1.0) | 3.4(1) | 5.2(1) | **8.5**(2) | 5.4(2) | 15/15 |
| DiagL | 1.6(0.6) | 2.8(1) | 4.3(2) | **4.4**(4) | **4.7**(3) | **13**(31) | **21**(45) | 13/15 | DiagL | **0.64**(0.2) | **1.8**(0.7) | 2.7(1) | 3.6(1.0) | 9.1(3) | 23(8) | 9.1(3) | 15/15 |
| ThresL | 1.8(0.6) | 5.4(22) | 7.0(4) | 6.9(11) | 10(26) | 20(2) | 30(27) | 14/15 | ThresL | 0.90(0.7) | 2.2(0.9) | **2.6**(1.0) | 3.2(0.9) | 4.3(2) | 8.7(3) | **4.4**(1) | 15/15 |

(continued)

**Table 1** (continued)

| $\Delta f_{\mathrm{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ | $\Delta f_{\mathrm{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **f8** | 326 | 921 | 1114 | 1217 | 1267 | 1315 | 1343 | 15/15 | **f18** | 238 | 836 | 7012 | 15928 | 27536 | 37234 | 42708 | 15/15 |
| CMSA | **3.5**(4) | 18(13) | 19(10) | 18(5) | 18(4) | 18(11) | 19(7) | 15/15 | CMSA | 5.8(34) | 289(1098) | 399(606) | ∞ | ∞ | ∞ | ∞ *2e5* | 0/15 |
| Thres | 7.5(7) | 35(46) | 33(13) | 32(31) | 31(33) | 31(31) | 31(8) | 15/15 | Thres | 31(0.4) | 193(177) | 399(278) | ∞ | ∞ | ∞ | ∞ *2e5* | 0/15 |
| Diag | 6.4(7) | **14**(10) | **15**(8) | **15**(8) | **15**(3) | **15**(8) | **16**(6) | 15/15 | Diag | 7.9(0.7) | **182**(161) | 196(128) | ∞ | ∞ | ∞ | ∞ *2e5* | 0/15 |
| DiagL | 6.6(0.3) | 17(4) | 18(3) | 17(7) | 17(4) | 18(2) | 18(4) | 15/15 | DiagL | **5.1**(0.5) | 189(299) | 192(268) | ∞ | ∞ | ∞ | ∞ *2e5* | 0/15 |
| ThresL | 8.5(10) | 18(11) | 18(10) | 18(9) | 18(7) | 18(7) | 18(7) | 15/15 | ThresL | 88(241) | 313(509) | **130**(110) | **184**(270) | ∞ | ∞ | ∞ *2e5* | 0/15 |
| $\Delta f_{\mathrm{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ | $\Delta f_{\mathrm{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ |
| **f9** | 200 | 648 | 857 | 993 | 1065 | 1138 | 1185 | 15/15 | **f21** | 130 | 2236 | 4392 | 4487 | 4618 | 5074 | 11329 | 8/15 |
| CMSA | **2.2**(0.7) | 31(53) | 28(13) | 26(36) | 25(32) | 24(13) | 24(9) | 15/15 | CMSA | 7.6(12) | 20(38) | 17(10) | 17(8) | 16(13) | 15(20) | 6.7(8) | 13/15 |
| Thres | 3.3(1) | 34(22) | 30(16) | 28(7) | 27(2) | 26(12) | 25(12) | 15/15 | Thres | 10(34) | **8.8**(12) | **12**(22) | **12**(39) | **12**(5) | **11**(6) | **4.8**(2) | 13/15 |
| Diag | 7.8(14) | 24(15) | 22(13) | 20(3) | 20(3) | 20(3) | 20(9) | 15/15 | Diag | 8.0(0.8) | 19(22) | 19(29) | 18(36) | 18(19) | 16(25) | 7.3(8) | 13/15 |
| DiagL | 4.9(10) | 35(29) | 31(13) | 29(18) | 27(9) | 27(15) | 26(12) | 15/15 | DiagL | 21(60) | 19(37) | 20(20) | 20(14) | 19(57) | 18(39) | 8.0(9) | 12/15 |
| ThresL | 4.4(5) | **18**(19) | **19**(8) | **18**(12) | **17**(12) | **17**(3) | **17**(7) | 15/15 | ThresL | **5.9**(10) | 17(8) | 15(10) | 15(15) | 14(26) | 13(23) | 5.9(12) | 13/15 |
| $\Delta f_{\mathrm{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ | $\Delta f_{\mathrm{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ |
| **f10** | 1835 | 2172 | 2455 | 2728 | 2802 | 4543 | 4739 | 15/15 | **f22** | 98 | 2839 | 6353 | 6620 | 6798 | 8296 | 10351 | 6/15 |
| CMSA | 6.8(3) | **7.9**(2) | **7.9**(2) | **7.5**(3) | **7.8**(2) | 5.2(0.9) | **5.1**(1) | 15/15 | CMSA | 26(60) | **4.1**(11) | **4.6**(17) | **4.6**(12) | **4.5**(5) | **3.9**(3) | **3.2**(7) | 14/15 |
| Thres | 16(7) | 16(3) | 16(3) | 14(3) | 14(3) | 9.1(2) | 9.0(2) | 15/15 | Thres | 26(26) | 10(17) | 12(15) | 12(11) | 12(7) | 10(8) | 8.4(8) | 13/15 |
| Diag | 8.7(3) | 10(3) | 9.4(2) | 9.0(2) | 9.4(2) | 6.1(1) | 6.1(1) | 15/15 | Diag | **18**(12) | 8.1(14) | 8.6(21) | 8.4(10) | 8.3(7) | 6.9(10) | 5.7(7) | 14/15 |
| DiagL | 14(4) | 14(3) | 14(2) | 13(2) | 13(2) | 8.4(1) | 8.3(1) | 15/15 | DiagL | 60(15) | 8.9(29) | 8.8(13) | 8.8(20) | 8.8(14) | 7.8(13) | 6.5(6) | 14/15 |
| ThresL | **6.6**(2) | 8.3(2) | 8.2(2) | 7.8(1) | 8.0(2) | **5.2**(0.8) | 5.3(1) | 15/15 | ThresL | 30(27) | 8.8(8) | 13(53) | 12(19) | 12(10) | 10(12) | 8.2(16) | 13/15 |

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 $\Delta f$-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

**Table 2** Expected running time (ERT in number of function evaluations) divided by the respective best ERT measured during BBOB-2009 in dimension 20

| $\Delta f_{\mathrm{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ | $\Delta f_{\mathrm{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **f1** | 43 | 43 | 43 | 43 | 43 | 43 | 43 | 15 /15 | **f9** | 1716 | 3102 | 3277 | 3379 | 3455 | 3594 | 3727 | 15/15 |
| CMSA | **4.7**(0.7) | 10 (2) | 14 (2) | 19 (3) | 25 (2) | 35 (3) | 45 (2) | 15 /15 | CMSA | **14**(7) | 51(33) | 52(62) | 51(64) | 51(91) | 49(29) | 48(135) | 12/15 |
| Thres | 4.9(1) | 9.3(1) | 14(2) | 19(2) | 23(2) | 33(1) | 42(1) | 15/15 | Thres | 34(19) | 51(3) | 55(32) | 57(32) | 59(31) | 60(29) | 60(30) | 14/15 |
| Diag | 5.3(3) | 9.5(3) | 14(3) | **18**(1) | **22**(3) | **31**(4) | **40**(4) | 15/15 | Diag | 18(6) | 53(101) | 53(70) | 53(62) | 52(61) | 51(4) | 50(29) | 12/15 |
| DiagL | 4.9(1) | **9.2**(2) | **13**(2) | 18(2) | 23(3) | 33(2) | 42(3) | 15/15 | DiagL | 36(11) | 52(35) | 54(6) | 56(5) | 57(32) | 57(57) | 57(27) | 14/15 |
| ThresL | 4.9(1) | 9.5(1) | 14(1) | 18(3) | 23(2) | 33(4) | 42(4) | 15/15 | ThresL | 15(3) | **20**(6) | **22**(7) | **23**(7) | **23**(7) | **23**(2) | **23**(5) | 15/15 |
| $\Delta f_{\mathrm{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ | $\Delta f_{\mathrm{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ |
| **f2** | 385 | 386 | 387 | 388 | 390 | 391 | 393 | 15/15 | **f10** | 7413 | 8661 | 10735 | 13641 | 14920 | 17073 | 17476 | 15/15 |
| CMSA | 167(44) | 221(53) | 245(51) | 268(51) | 279(35) | 290(40) | 298(39) | 15/15 | CMSA | 8.6(0.8) | **10**(1) | **9.0**(1) | **7.3**(2) | **6.9**(0.8) | **6.3**(0.4) | **6.4**(1.0) | 15/15 |
| Thres | **95**(15) | **109**(12) | **117**(4) | **120**(8) | **124**(7) | **127**(10) | **127**(7) | 15/15 | Thres | 35(3) | 33(2) | 27(2) | 22(2) | 21(1) | 18(2) | 18(1) | 15/15 |
| Diag | 169(46) | 231(41) | 246(67) | 261(41) | 270(37) | 282(32) | 291(30) | 15/15 | Diag | 9.1(4) | 10(2) | 9.1(2) | 7.6(0.9) | 7.1(0.5) | 6.5(0.5) | 6.6(0.4) | 15/15 |
| DiagL | 96(20) | 113(11) | 122(12) | 126(5) | 128(14) | 130(10) | 130(9) | 15/15 | DiagL | 33(2) | 31(3) | 26(2) | 20(2) | 19(2) | 17(1) | 17(1) | 15/15 |
| ThresL | 154(35) | 212(65) | 245(26) | 259(12) | 265(34) | 273(31) | 282(38) | 15/15 | ThresL | **8.4**(3) | 10(2) | 9.1(2) | 7.8(1) | 7.4(1.0) | 6.8(0.5) | 6.8(0.7) | 15/15 |
| $\Delta f_{\mathrm{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ | $\Delta f_{\mathrm{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ |
| **f5** | 41 | 41 | 41 | 41 | 41 | 41 | 41 | 15/15 | **f11** | 1002 | 2228 | 6278 | 8586 | 9762 | 12285 | 14831 | 15/15 |
| CMSA | **11**(4) | **13**(6) | **14**(3) | **14**(4) | **14**(6) | **14**(2) | **14**(5) | 15/15 | CMSA | **12**(2)$^{\star 2}$ | **7.1**(0.9) | **3.0**(0.6) | **2.6**(0.6) | 2.7(0.6) | 2.7(0.7) | 2.7(0.7) | 15/15 |
| Thres | 12(4) | 15(4) | 15(6) | 15(6) | 15(6) | 15(8) | 15(5) | 15/15 | Thres | 260(34) | 130(7) | 48(5) | 36(3) | 32(7) | 26(4) | 22(3) | 15/15 |
| Diag | 11(4) | 15(6) | 16(6) | 16(4) | 16(6) | 16(6) | 16(5) | 15/15 | Diag | 16(3) | 9.1(0.9) | 3.6(0.5) | 3.0(0.5) | 2.9(0.5) | 2.7(0.3) | 2.5(0.5) | 15/15 |
| DiagL | 14(5) | 17(7) | 18(8) | 18(8) | 18(10) | 18(7) | 18(6) | 15/15 | DiagL | 223(54) | 115(18) | 43(8) | 33(7) | 30(3) | 24(4) | 20(3) | 15/15 |
| ThresL | 13(6) | 17(6) | 18(13) | 18(14) | 18(5) | 18(10) | 18(11) | 15/15 | ThresL | 15(2) | 8.1(0.8) | 3.2(0.3) | 2.6(0.3) | **2.6**(0.3) | **2.3**(0.4) | **2.2**(0.3) | 15/15 |

(continued)

**Table 2** (continued)

| $\Delta f_{\text{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ | $\Delta f_{\text{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **f6** | 1296 | 2343 | 3413 | 4255 | 5220 | 6728 | 8409 | 15/15 | **f12** | 1042 | 1938 | 2740 | 3156 | 4140 | 12407 | 13827 | 15/15 |
| CMSA | **2.1**(0.8) | **3.4**(6) | 5.4(9) | 18(12) | 51(89) | 255(307) | ∞ *4e5* | 0/15 | CMSA | **3.2**(7) | **7.9**(14) | **13**(6) | **14**(6) | **13**(9) | **5.6**(2) | **5.8**(1) | 15/15 |
| Thres | 5.2(15) | 16(32) | 29(28) | 37(25) | 47(33) | 121(158) | 317(416) | 2/15 | Thres | 114(160) | 150(148) | 279(254) | 309(127) | 480(291) | ∞ | ∞ *4e5* | 0/15 |
| Diag | 3.4(5) | 4.2(2) | 6.1(9) | 14(29) | 34(38) | **64**(137) | 211(184) | 3/15 | Diag | 6.4(0.2) | 15(15) | 17(14) | 18(11) | 17(7) | 6.9(3) | 7.0(2) | 15/15 |
| DiagL | 17(34) | 26(35) | 32(27) | 51(40) | 64(69) | 135(127) | **204**(279) | 3/15 | DiagL | 18(64) | 96(83) | 103(65) | 123(40) | 113(6) | 59(33) | 86(37) | 5/15 |
| ThresL | 3.2(2) | 4.4(6) | **5.0**(6) | **7.9**(12) | **21**(22) | 83(75) | 324(290) | 2/15 | ThresL | 14(27) | 21(20) | 23(13) | 25(15) | 22(8) | 8.6(2) | 8.5(3) | 15/15 |
| $\Delta f_{\text{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ | $\Delta f_{\text{opt}}$ | 1e1 | 1e0 | 1e-1 | 1e-2 | 1e-3 | 1e-5 | 1e-7 | #succ |
| **f8** | 2039 | 3871 | 4040 | 4148 | 4219 | 4371 | 4484 | 15/15 | **f14** | 75 | 239 | 304 | 451 | 932 | 1648 | 15661 | 15/15 |
| CMSA | **12**(8) | **31**(80) | **32**(34) | **32**(53) | **32**(71) | **32**(53) | **32**(93) | 13/15 | CMSA | 2.0(2) | 2.1(0.5) | 2.6(0.4) | 3.3(1) | **4.9**(0.9)$^{\star 3}$ | **12**(2) | 4.1(1) | 15/15 |
| Thres | 28(9) | 70(86) | 73(77) | 74(27) | 75(26) | 75(27) | 75(49) | 11/15 | Thres | **1.7**(1) | 1.9(0.8) | **2.3**(0.5) | 3.2(0.5) | 15(3) | 117(15) | 18(3) | 11/12 |
| Diag | 14(7) | 31(4) | 32(6) | 33(28) | 33(4) | 32(25) | 32(4) | 13/15 | Diag | 1.8(1) | 2.0(0.9) | 2.5(0.8) | 3.2(1) | 7.0(1) | 13(0.9) | 4.0(0.5) | 15/15 |
| DiagL | 28(14) | 82(132) | 84(82) | 85(75) | 86(53) | 86(49) | 85(68) | 10/15 | DiagL | 1.8(1) | 1.9(0.5) | 2.4(0.6) | 3.2(0.4) | 15(4) | 101(32) | 16(2) | 15/15 |
| ThresL | 15(6) | 33(78) | 34(27) | 34(27) | 34(3) | 34(25) | 34(68) | 13/15 | ThresL | 1.9(1) | **1.8**(0.7) | 2.3(0.3) | **3.0**(0.6) | 6.8(0.6) | 14(1) | **3.7**(0.9) | 15/15 |

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 $\Delta f$-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 exhibits slower convergence. Here, the CMSA-ES is marked as the best strategy for many intermediate precision targets. However, the CMSA-Diag-ES and the CMSA-ThresL-ES achieve very similar results. Interesting is the mixture of thresholding target and thresholding function. The interactions will be investigated more closely in future work. In the case of the rotated Rosenbrock (id 9), the CMSA-ThresL-ES shows the best results.

For the ill-conditioned functions (id 10–14), the findings are mixed. On some functions, especially on the ellipsoid (id 10) and the bent cigar (id 12), the original CMSA-ES has the lowest ERT values for the precision targets. For $N = 10$, Table 1, all strategies are successful for the ellipsoid (id 10), the discus (id 11), the bent cigar (id 12), and the sum of different powers (id 14). For the higher-dimensional search space, the bent cigar leads to problems for the CMSA-Thres-ES and the CMSA-DiagL-ES. Again, there appears to be an interaction between thresholding target and function. Only the CMSA-ES and the CMSA-Diag-ES are able to reach the final precision target on the sharp ridge (id 13) for $N = 10$. Since this occurs just once in both cases, more experiments are clearly necessary. Interestingly, differences between the group consisting of the CMSA-ES, the CMSA-ThresL-ES, and the CMSA-Diag-ES and the remaining strategies can be observed. The latter group is unable to achieve comparable performance on f11, f12, and f13 with more unsuccessful runs and larger expected numbers of function evaluations especially for the lower targets.

The group of multi-modal functions represents challenges for all ESs under consideration: The 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) cannot be solved with the final target precision required. Partly, this may be due to the maximal number of fitness evaluations. Even the best performing methods of the 2009 BBOB workshop required more evaluations than we allowed in total. Thus, longer experiments should be conducted in future research. Concerning the preliminary targets with lower precision, thresholding variants often achieve the best results. However, more experiments are required. In the case of $N = 20$, the number of function evaluations necessary for the best algorithms of 2009 to reach even the lower precision target of $10^{-1}$ exceeds our total budget. Therefore, no analysis is attempted and the results are not shown in Table 2.

The last group, the multi-modal functions with weak global structures, are also difficult to solve and struck from Table 2. Only for function 21, Gallagher 101 peaks, and function 22, Gallagher 21 peaks, successful runs are observed for $N = 10$, see Table 1. In the case of the first, the CMSA-Thres-ES achieves the best results, whereas the original CMSA-ES is the best strategy to tackle function 22.

To summarize the findings, thresholding appears as a means to improve the performance. However, we observe an interaction between thresholding function and threshold target that should be analyzed further.

## 5 Conclusions and Outlook

The focus of the paper lay on the covariance matrix adaptation in evolution strategies. In many cases, the sample covariance is used which gives cause for concern regarding that its quality may be poor in situations where the estimation is only based on a small sample. Alternative approaches have been developed in the field of statistics. Evolution strategies require, however, methods that do not increase the computational effort considerably. Therefore, the paper investigated and compared several thresholding techniques which originate from estimation theory for high-dimensional spaces. The performance of the resulting new evolution strategies were compared to the original variant on the black-box optimization benchmarking test suite. The results were promising with the new variants performing better for several function classes. Concerning the variants of thresholding, more experiments and analyses are required in order to identify the best solution and to shed more light on the interaction between thresholding function and thresholding target.

## References

1. Bäck, T., Foussette, C., Krause, P.: Contemporary Evolution Strategies. Natural Computing. Springer, Berlin (2013)
2. Beyer, H.-G., Meyer-Nieberg, S.: Self-adaptation of evolution strategies under noisy fitness evaluations. Genetic Program. Evolvable Mach. **7**(4), 295–328 (2006)
3. Beyer, H.-G., Schwefel, H.-P.: Evolution strategies: a comprehensive introduction. Nat. Comput. **1**(1), 3–52 (2002)
4. Beyer, H.-G., Sendhoff, B.: Covariance matrix adaptation revisited—the CMSA evolution strategy. In: Rudolph, G. et al. (eds.), *PPSN*, volume 5199 of *Lecture Notes in Computer Science*, pp. 123–132. Springer (2008)
5. Cai, T., Liu, W.: Adaptive thresholding for sparse covariance matrix estimation. J. Am. Stat. Assoc. **106**(494), 672–684 (2011)
6. Dong, W., Yao, X.: Covariance matrix repairing in Gaussian based EDAs. In: IEEE Congress on Evolutionary Computation, CEC 2007, pp. 415–422 (2007)
7. Fan, J., Liao, Y., Liu, H.: An overview on the estimation of large covariance and precision matrices. arXiv:1504.02995
8. Finck, S., Hansen, N., Ros, R., Auger, A.: Real-parameter black-box optimization benchmarking 2010: presentation of the noiseless functions. Technical report, Institute National de Recherche en Informatique et Automatique, 2009/22 (2010)
9. Hansen, N.: The CMA evolution strategy: a comparing review. In: Lozano, J. et al. (eds.) Towards a New Evolutionary Computation. Advances in Estimation of Distribution Algorithms, pp. 75–102. Springer (2006)
10. Hansen, N., Auger, A., Finck, S., Ros, R.: Real-parameter black-box optimization benchmarking 2012: experimental setup. Technical report, INRIA (2012)
11. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolut. Comput. **9**(2), 159–195 (2001)
12. Kramer, O.: Evolution strategies with Ledoit-Wolf covariance matrix estimation. In: 2015 IEEE Congress on Evolutionary Computation (IEEE CEC) (2015)
13. Ledoit, O., Wolf, M.: A well-conditioned estimator for large dimensional covariance matrices. J. Multivar. Anal. Arch. **88**(2), 265–411 (2004)