A New Look at the Covariance Matrix Estimation in Evolution Strategies

Silja Meyer-Nieberg(✉) and E. Kropat

Universität der Bundeswehr München,
Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany
{silja.meyer-nieberg,erik.kropat}@unibw.de

Abstract. Evolution strategies belong to the best performing modern natural computing methods for continuous optimization. This paper takes a new look at the covariance matrix adaptation, a mechanism which is central to the algorithm. The adaptation focusses strongly on the sample covariance. However, as known from modern statistics, this estimate may be of poor quality if certain conditions are not fulfilled. Unfortunately, this is often the case in practice. This paper compares the established methods for the covariance correction in evolution strategies with the approaches in modern statistics. Furthermore, it introduces and evaluates new covariance correction schemes.

Keywords: Evolutionary algorithms · Continuous optimization · Evolution strategies · Covariance · Shrinkage

1 Introduction

Black-Box optimization is an important subcategory of optimization. Over the years, several methods have been developed - ranging from simple pattern search over mesh adaptive methods to natural computing, see e.g. [1,8,10]. This paper focuses on evolution strategies (ESs) which represent well-performing metaheuristics for continuous, non-linear optimization. In recent workshops on black-box optimization, see e.g. [15], variants of this particular subtype of evolutionary algorithms have emerged as one the best performing methods among a broad range of competitors stemming from natural computing. Evolution strategies rely primarily on random changes to move through the search space. These random changes, usually normally distributed random variables, must be controlled by adapting both, the extend and the direction of the movements.

Modern evolution strategies apply therefore covariance matrix and step-size adaptation – with great success. However, most methods use the common estimate of the population covariance matrix as one component to guide the search. Here, there may be room for further improvement, especially with regard to common application cases of evolution strategies which usually concern optimization in high-dimensional search spaces. For efficiency reasons, the population size λ, that is, the number of candidate solutions, is kept below the search space dimensionality N and scales usually with $\mathcal{O}(\log(N))$ or with $\mathcal{O}(N)$. In other words,

E. Pinson et al. (Eds.): ICORES 2014, CCIS 509, pp. 157–172, 2015.
DOI: 10.1007/978-3-319-17509-6_11

either $\lambda \ll N$ or $\lambda \approx N$ which may represent a problem when using the sample covariance matrix. This even more so, since the sample size used in the estimation is just a fraction of the population size. Furthermore, the result is not robust against outliers which may appear in practical optimization which has often to cope with noise. This paper introduces and explores new approaches addressing the first problem by developing a new estimate for the covariance matrix. To our knowledge, these estimators have not been applied to evolution strategies before.

The paper is structured as follows: First, evolution strategies are introduced and common ways to adapt the covariance matrix are described and explained. Afterwards, we point out a potential dangerous weakness of the traditionally used estimate of the population covariance. Candidates for better estimates are presented and described in the following section. We propose and investigate several approaches ranging from a transfer of shrinkage estimators over a maximum entropy covariance selection principle to a new combination of both approaches. The quality of the resulting algorithms is assessed in the experimental test section. Conclusions and possible further research directions constitute the last part of the paper.

1.1 Evolution Strategies

Evolutionary algorithms (EAs) [10] are population-based stochastic search and optimization algorithms including today genetic algorithms, genetic programming, (natural) evolution strategies, evolutionary programming, and differential evolution. As a rule, they require only weak preconditions on the function to be optimized. Therefore, they are applicable in cases when only point-wise function evaluations are possible.

An evolutionary algorithm starts with an initial population of candidate solutions. The individuals are either drawn randomly from the search space or are initialized according to previous information on good solutions. A subset of the parent population is chosen for the creation of the offspring. This process is termed parent selection. Creation normally consists of recombination and mutation. While recombination combines traits from two or more parents, mutation is an unary operator and is realized by random perturbations. After the offspring have been created, survivor selection is performed to determine the next parent population. Evolutionary algorithms differ in the representation of the solutions and in the realization of the selection, recombination, and mutation operators.

Evolution strategies (ESs) [20,22] are a variant of evolutionary algorithms that is predominantly applied in continuous search spaces. Evolution strategies are commonly notated as $(\mu/\rho, \lambda)$-ESs. The parameter μ stands for the size of the parent population. In the case of recombination, ρ parents are chosen randomly and are combined for the recombination result. While other forms exist, recombination usually consists of determining the weighted mean of the parents [4]. 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 essential and can be seen as the main search operator. Afterwards, 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.

There are two main types of evolution strategies: Evolution strategies with "plus"-selection and ESs with "comma"-selection. The first select the μ-best offspring and parents as the next parent population, where ESs with "comma"-selection discard the old parent population completely and take only the best offspring. Methods for adapting the scale factor σ or the full covariance matrix have received a lot of attention (see [19]). The main approaches are described in the following section.

1.2 Covariance Matrix Adaptation

First, the update of the covariance matrix is addressed. In evolution strategies two types exist: one applied in the *covariance matrix adaptation evolution strategy* (CMA-ES) [14] which considers past information from the search and an alternative used by the *covariance matrix self-adaptation evolution strategy* (CMSA-ES) [5] which focusses more on the present population.

The covariance matrix update of the CMA-ES is explained first. The CMA-ES uses weighted intermediate recombination, in other words, it computes the weighted centroid of the μ best individuals of the population. This mean $\mathbf{m}^{(g)}$ is used for creating all offspring by adding a random vector drawn from a normal distribution with covariance matrix $(\sigma^{(g)})^2\mathbf{C}^{(g)}$, i.e., the actual covariance matrix consists of a general scaling factor (or step-size or mutation strength) and the matrix denoting the directions. Following usual notation in evolution strategies this matrix $\mathbf{C}^{(g)}$ will be referred to as *covariance matrix* in the following.

The basis for the CMA update is the common estimate of the covariance matrix using the newly created population. Instead of considering the whole population 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 [14],

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

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

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

To derive reliable estimates larger population sizes are usually necessary which is detrimental with regard to the algorithm's speed. Therefore, past information, that is, past covariance matrizes are usually also considered

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

with parameter $0 \leq c_\mu \leq 1$ determining the effective time-horizon. In CMA-ESs, it has been found that an enhance of the general search direction in the

covariance matrix is usual beneficial. For this, the concepts of the *evolutionary path* and the *rank-one-update* are introduced. As its name already suggests, an evolutionary path considers the path in the search space the population has taken so far. The weighted means serve as representatives. Defining

$$\mathbf{v}^{(g+1)} := \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}$$

the evolutionary path reads

$$\mathbf{p}_c^{(g+1)} := (1 - c_c)\mathbf{p}_c^{(g)} + \sqrt{c_c(2 - c_c)\mu_{\text{eff}}}\left(\frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}\right). \tag{4}$$

For details on the parameters, see e.g. [12]. The evolutionary path gives a general search direction that the ES has taken in the recent past. In order to bias the covariance matrix accordingly, the rank-one-update

$$\mathbf{C}_1^{(g+1)} := \mathbf{p}_c^{(g+1)}(\mathbf{p}_c^{(g+1)})^{\mathrm{T}} \tag{5}$$

is performed and used as a further component of the covariance matrix. A normal distribution with covariance $\mathbf{C}_1^{(g+1)}$ leads towards a one-dimensional distribution on the line defined by $\mathbf{p}_c^{(g+1)}$. With (5) and (3), the final covariance update of the CMA-ES reads

$$\mathbf{C}^{(g+1)} := (1 - c_1 - c_\mu)\mathbf{C}^{(g)} + c_1\mathbf{C}_1^{(g+1)} + c_\mu\mathbf{C}_\mu^{(g+1)}. \tag{6}$$

The CMA-ES is one of the most powerful evolution strategies. However, as pointed out in [5], its scaling behavior with the population size is not good. The alternative approach of the CMSA-ES [5] updates the covariance matrix differently. Considering again the definition (1), the covariance update is a convex combination of the old covariance and the population covariance, i.e., the rank-μ update

$$\mathbf{C}^{(g+1)} := (1 - \frac{1}{c_\tau})\mathbf{C}^{(g)} + \frac{1}{c_\tau}\sum_{m=1}^{\mu} w_m \mathbf{y}_{m:\lambda}^{(g+1)}(\mathbf{y}_{m:\lambda}^{(g+1)})^{\mathrm{T}} \tag{7}$$

with the weights usually set to $w_m = 1/\mu$. See [5] for information on the free parameter c_τ.

1.3 Step-Size Adaptation

The CMA-ES uses the so-called *cumulative step-size adaptation* (CSA) to control the scaling parameter (also called *step-size*, *mutation strength* or *step-length*) [12]. To this end, the CSA determines again an evolutionary path by summating the movement of the population centers

$$\mathbf{p}_\sigma^{(g+1)} = (1 - c_\sigma)\mathbf{p}_\sigma^{(g)} + \sqrt{c_\sigma(2 - c_\sigma)\mu_{\text{eff}}}(\mathbf{C}^{(g)})^{-\frac{1}{2}}\frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}} \tag{8}$$

eliminating the influence of the covariance matrix and the step length. For a detailed description of the parameters, see [12]. The length of the path in (8) is important. In the case of short path lengths, several movement of the centers counteract each other which is an indication that the step-size is too large and should be reduced. If on the other hand, the ES takes several consecutive steps in approximately the same direction, progress and algorithm speed would be improved, if larger changes were possible. Long path lengths, therefore, are an indicator for a required increase of the step length. Ideally, the CSA should result in uncorrelated steps.

After some calculations, see [12], the ideal situation is revealed as standard normally distributed steps, which leads to

$$\ln(\sigma^{(g+1)}) = \ln(\sigma^{(g)}) + \frac{c_\sigma}{d_\sigma}\Big(\frac{\|\mathbf{p}_\sigma^{(g+1)}\| - \mu_{\chi_n}}{\mu_{\chi_n}}\Big) \tag{9}$$

as the CSA-rule. The change is multiplicative in order to avoid numerical problems and results in non-negative scaling parameters. The parameter μ_{χ_n} in (9) stands for the mean of the χ-distribution with n degrees of freedom. If a random variable follows a χ_n^2 distribution, its square root is χ-distributed. The degrees of freedom coincide with the search space dimension. The CSA-rule works well in many application cases. It can be shown, however, that the original CSA encounter problems in large noise regimes resulting in a loss of step-size control and premature convergence. Therefore, uncertainty handling procedures and other safeguards are advisable.

An alternative approach for adapting the step-size is *self-adaptation* first introduced in [20] and developed further in [22]. It subjects the strategy parameters of the mutation to evolution. In other words, the scaling parameter or in its full form, the whole 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: On average good parameters should lead to 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 used nowadays mainly to adapt the step-size or a diagonal covariance matrix. In the case of the mutation strength, usually a log-normal distribution

$$\sigma_l^{(g)} = \sigma_{\text{base}} \exp(\tau \mathcal{N}(0,1)) \tag{10}$$

is used for mutation. The parameter τ is called the *learning rate* and is usually chosen to scale with $1/\sqrt{2N}$. The variable σ_{base} is either the parental scale factor or the result of recombination. 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. The baseline σ_{base} is either the mutation strength of the parent or if recombination is used the recombination result. Self-adaptation with recombination has been shown to be "robust" against noise [3] and is used in the CMSA-ES as update rule for the scaling factor. In [5] it was found that the CMSA-ES performs comparably to the CMA-ES for smaller populations but is less computational expensive for larger population sizes.

2 Concerning the Covariance Estimator

The covariance matrix $\mathbf{C}_\mu$ which appears in (2) and (7) can be interpreted as the sample covariance matrix with sample size μ. Two differences are present. The first using μ instead of $\mu-1$ can be explained by using the known mean instead of an estimate. The second lies in the non-identically distributed random variables of the population since order statistics appear. We will disregard that problem for the time being.

In the case of identically independently distributed random variables, the estimate converges almost surely towards the "true" covariance Σ for $\mu \to \infty$. In addition, the sample covariance matrix is related (in our case equal) to the maximum likelihood (ML) estimator of Σ. Both facts serve a justification to take $\mathbf{C}_\mu$ as the substitute for the unknown true covariance for large μ. However, the quality of the estimate can be quite poor if $\mu < N$ or even $\mu \approx N$.

This was first discovered by Stein [23,24]. Stein's phenomenon states that while the ML estimate is often seen as the best possible guess, its quality may be poor and can be improved in many cases. This holds especially for high-dimensional spaces. The same problem transfers to covariance matrix estimation, see [21]. Also recognized by Stein, in case of small ratios μ/N the eigenstructure of $\mathbf{C}_\mu$ may not agree well with the true eigenstructure of Σ. As stated in [17], the largest eigenvalue has a tendency towards too large values, whereas the smallest shows the opposite behavior. This results in a larger spectrum of the sample covariance matrix with respect to the true covariance for $N/\mu \not\to 0$ for $\mu, N \to \infty$ [2]. As found by Huber [16], a heavy tail distribution leads also to a distortion of the sample covariance.

In statistics, considerable efforts have been made to find more reliable and robust estimates. Owing to the great inportance of the covariance matrix in data mining and other statistical analyses, work is still ongoing. The following section provides a short introduction before focussing on the approach used for evolution strategies.

3 Approaches for Estimating the Covariance

As stated above, the estimation of high-dimensional covariance matrices has received a lot of attention, see e.g. [6]. Several types have been introduced, for example: shrinkage estimators, banding and tapering estimators, sparse matrix transform estimators, and the graphical Lasso estimator. This paper concentrates on shrinkage estimators and on an idea inspired by a maximum entropy

approach. Both classes can be computed comparatively efficiently. Future research will consider other classes of estimators.

3.1 Shrinkage Estimators

Most (linear) shrinkage estimators use the convex combination

$$\mathbf{S}_{\text{est}}(\rho) = \rho\mathbf{F} + (1-\rho)\mathbf{C}_\mu \tag{11}$$

with $\mathbf{F}$ the *target* to correct the estimate provided by the sample covariance. The parameter $\rho \in]0,1[$ is called the *shrinkage intensity*. Equation (11) is used to shrink the eigenvalues of $\mathbf{C}_\mu$ towards the eigenvalues of $\mathbf{F}$. The shrinkage intensity ρ should be chosen to minimize

$$\mathrm{E}\Big(\|\mathbf{S}_{\text{est}}(\rho) - \Sigma\|_F^2\Big) \tag{12}$$

with $\|\cdot\|_F^2$ denoting the squared Frobenius norm with

$$\|\mathbf{A}\|_F^2 = \frac{1}{N}\mathrm{Tr}\Big[\mathbf{A}\mathbf{A}^{\mathrm{T}}\Big], \tag{13}$$

see [17]. To solve this problem, knowledge of the true covariance Σ would be required which is usually unobtainable.

Starting from (12), Ledoit and Wolf obtained an analytical expression for the optimal shrinkage intensity for the target $\mathbf{F} = \mathrm{Tr}(\mathbf{C}_\mu)/N\,\mathbf{I}$. The result does not make assumptions on the underlying distribution. In the case of $\mu \approx N$ or vastly different eigenvalues, the shrinkage estimator does not differ much from the sample covariance matrix, however.

Other authors introduced different estimators, see e.g. [7] or [6]). Ledoit and Wolfe themselves considered non-linear shrinkage estimators [18]. Most of the approaches require larger computational efforts. In the case of the non-linear shrinkage, for example, the authors are faced with a non-linear, non-convex optimization problem, which they solve by using sequential linear programming [18]. A general analytical expression is unobtainable, however.

Shrinkage estimators and other estimators aside from the standard case have not been used in in evolution strategies before. A literature review resulted in one application in the case of Gaussian based estimation of distribution algorithms albeit with quite a different goal [9]. There, the learning of the covariance matrix during the run lead to non positive definite matrices. A shrinkage procedure was applied to "repair" the covariance matrix towards the required structure. The authors used a similar approach as in [17] but made the shrinkage intensity adaptable.

Interestingly, (3), (6), and (7) of the ES algorithm can be interpreted as a special case of shrinkage. In the case of the CMSA-ES, for example, the estimate is shrunk towards the old covariance matrix. The shrinkage intensity is determined by

$$c_\tau = 1 + \frac{N(N+1)}{2\mu} \tag{14}$$

as $\rho = 1 - 1/c_\tau$. As long as the increase of μ with the dimensionality N is below $\mathcal{O}(N^2)$, the coefficient (14) approaches infinity for $N \to \infty$. Since the contribution of the sample covariance to the new covariance in (7) is weighted with $1/c_\tau$, its influence fades out for increasing dimensions. It is the aim of the paper to investigate whether a further shrinkage can improve the result.

Transferring shrinkage estimators to ESs must take the situation in which the estimation occurs into account since it differs from the assumptions in statistical literature. The covariance matrix $\Sigma = \mathbf{C}^{g-1}$ that was used to create the offspring is known. The sample is based on rank-based selection, however, which differs from the iid case usually considered. Only if there were no selection pressure, the sample $\mathbf{x}_1, \ldots, \mathbf{x}_\mu$ would represent normally distributed random variables. In this context, it is interesting to note that the argumentation in [12] with respect to the setting of the CMA-ES parameter argues to choose the parameter so that the distribution of the random variables remains unchained as long as no selection pressure occurs. In other words, if $\mathbf{p}^{(g)} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}^{(g)})$ then also $\mathbf{p}^{(g+1)}$ for both evolution paths, (4) and (8). However, due to sampling and using the covariance estimate, larger deviations may occur. Applying shrinkage could improve the situation. However, the choice of the target remains. Most shrinkage approaches consider diagonal matrices as shrinkage targets. If we were following that approach, we could choose the matrix $\mathbf{F} = \text{diag}(\mathbf{C}_\mu)$. This would leave the diagonal elements of the sample covariance matrix unchanged decreasing only the off-diagonal entries. However, a shrinkage towards a diagonal does not appear to be a good idea for optimizing functions that are not oriented towards the coordinate system.

3.2 A Maximum Entropy Covariance Estimation

Therefore, we make use of another concept following [25]. Confronted with the problem of determining a reliable covariance matrix by combining a sample covariance matrix with a pooled variance matrix, the authors introduced a *maximum entropy covariance selection principle*. Since a combination of covariance matrices also appears in evolution strategies, a closer look at their approach is interesting. Defining a population matrix $\mathbf{C}_p$ and the sample covariance matrix $\mathbf{S}_i$, the mixture

$$\mathbf{S}_{mix}(\eta) = \eta \mathbf{C}_p + (1 - \eta)\mathbf{S}_i \tag{15}$$

was considered. In departure from usual approaches, focus lay on the combination of the two matrixes that maximizes the entropy. To this end, the coordinate system was changed to the eigenspace of $\mathbf{S}_{mix} = \mathbf{C}_p + (1-\eta)\mathbf{S}_i$. Let $\mathbf{M}_S$ denote the (normalized) eigenvectors of the mixture matrix. The representations of $\mathbf{C}_p$ and $\mathbf{S}_i$ in this coordination system read

$$\begin{aligned} \Phi^C &= \mathbf{M}_S^{\mathrm{T}} \mathbf{C}_p \mathbf{M}_S \\ \Phi^S &= \mathbf{M}_S^{\mathrm{T}} \mathbf{S}_i \mathbf{M}_S. \end{aligned} \tag{16}$$

Both matrices are usually not diagonal. To construct the new estimate for the covariance matrix,

$$\begin{aligned} \Lambda^C &= \mathrm{diag}(\Phi^C) \\ \Lambda^S &= \mathrm{diag}(\Phi^S) \end{aligned} \tag{17}$$

were determined. By taking $\lambda_i = \max(\lambda_i^C, \lambda_i^S)$, a covariance matrix estimate could finally be constructed via $\mathbf{M}_S \Lambda \mathbf{M}_S^{\mathrm{T}}$. The approach maximizes the possible contributions to the principal direction of the mixture matrix and is based on a maximum entropy derivation for the estimation.

3.3 New Covariance Estimators

This paper proposes a combination of a shrinkage estimator and the basis transformation introduced [25] for a use in evolution strategies. This paper focuses on the CMSA-ES. The aim is to switch towards a suitable coordinate system and then either to discard the contributions of the sample covariance that are not properly aligned or to shrink the off-diagonal components. Two choices for the mixture matrix represent themselves. The first

$$\mathbf{S}_{mix} = \mathbf{C}^g + \mathbf{C}_\mu \tag{18}$$

is be chosen in accordance to [25]. The second takes the covariance result that would have been used in the original CMSA-ES

$$\mathbf{S}_{mix} = (1 - c_\tau)\mathbf{C}^g + c_\tau \mathbf{C}_\mu \tag{19}$$

and introduces a single step recursion which may be more appropriate for small population sizes. Both choices will be investigated in this paper. They in turn can be coupled with several further ways to proceed and to construct the new covariance matrix. Switching towards the eigenspace of $\mathbf{S}_{mix}$, results in the covariance matrix representations $\Phi_\mu := \mathbf{M}_S^{\mathrm{T}} \mathbf{C}_\mu \mathbf{M}_S$ and $\Phi_\Sigma := \mathbf{M}_S^{\mathrm{T}} \mathbf{C}^g \mathbf{M}_S$.

1. The first approach for constructing a new estimate of the sample covariance is to apply the principle of maximal contribution to the axes from [25] and to determine

$$\Lambda_\mu = \max\Big(\mathrm{diag}(\Phi_\mu), \mathrm{diag}(\Phi_\Sigma)\Big) \tag{20}$$

   The sample covariance matrix can then be computed as $\mathbf{C}'_\mu = \mathbf{M}_S \Lambda_\mu \mathbf{M}_S^{\mathrm{T}}$.
2. Another approach would be to discard all entries of Φ_μ except the diagonal

$$\Lambda_\mu = \mathrm{diag}(\Phi_\mu) \tag{21}$$

3. A third approach consists of applying a shrinkage estimator like

$$\Phi_\mu^S = (1 - \rho)\Phi_\mu + \rho \mathrm{diag}(\Phi_\mu). \tag{22}$$

   This approach does not discard the off-diagonal entries completely. The shrinkage intensity ρ remains to be determined.

4 Experimental Evaluation

This section describes the experiments that were performed to explore the new approaches. For our investigation, the CMSA-ES version is considered since it operates just with the population covariance matrix and effects from changing the estimate should be easier to discerned. The competitors consist of algorithms which use shrinkage estimators as defined in (18) to (22). This code is not optimized for performance with respect to absolute computing time, since this paper aims at a proof of concept. The experiments are performed for the search space dimensions $N = 2, 5, 10$, and 20. The maximal number of fitness evaluations is $\mathrm{FE}_{\max} = 2 \times 10^4 N$. The CMSA-ES versions use $\lambda = \lfloor \log(3N) + 8 \rfloor$ offspring and $\mu = \lceil \lambda/4 \rceil$ parents. The start position of the algorithms is randomly chosen from a normal distribution with mean zero and standard deviation of 0.5. A run terminates prematurely if the difference between the best value obtained so far and the optimal fitness value $|f_{\text{best}} - f_{\text{opt}}|$ is below a predefined precision set to 10^{-8}. For each fitness function and dimension, 15 runs are used.

4.1 Test Suite

The experiments are performed with the black box optimization benchmarking (BBOB) software framework and the test suite introduced for the black box optimization workshops, see [13]. The aim of the workshop is to benchmark and compare metaheuristics and other direct search methods for continuous optimization. The framework 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.

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 24 noiseless functions [11]. 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).

4.2 Performance Measure

The following performance measure is used in accordance to [13]. 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 [13]. In this paper, we use

$$\mathrm{ERT} = \frac{\#(FEs(f_{\text{best}} \geq f_{\text{target}}))}{\#succ} \tag{23}$$

as an estimate by summing up the fitness evaluations $FEs(f_{\text{best}} \geq f_{\text{target}})$ of each run until the fitness of the best individual is smaller than the target value, divided by all successfull runs (Fig. 1).

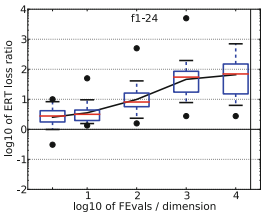
4.3 Results and Discussion

Due to space restrictions, Figure 3 and Table 1 and Fig. 2 show only the results from the best experiments which were achieved for the variant which used (22) together with (19) as the transformation matrix (called CMSA-shr-ES in the following). First of all, it should be noted that there is no significant advantage to either algorithm for the test suite functions. Table 1 and Fig. 2 show the ERT loss ratio with respect to the best result from the BBOB 2009 workshop for predefined budgets given in the first column. The median performance of both algorithms improves with the dimension until the budget of 10^3 – which is interesting. An increase of the budget goes along with a decreased performance which is less pronounced for the CMSA-shr-ES in the case of the larger dimensional space. This indicates that the CMSA-shr-ES may perform more favorable in larger search spaces as envisioned. Further experiments which a larger maximal number of fitness evaluations and larger dimensional spaces will be conducted which should shed more light on the behavior. Furthermore, the decrease in performance with the budget hints at a search stagnation probably due to convergence into local optima. Restart strategies may be beneficial, but since they have to be fitted to the algorithms, we do not apply them in the present paper.

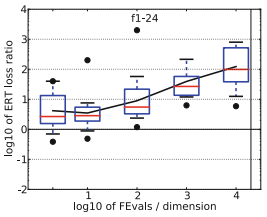
Figure 3 shows the expected running time for reaching the precision of 10^{-8} for all 24 functions and search space dimensionalities. In the case of the separable functions (1–5), both algorithms show a very similar behavior, succeeding in optimizing the first two functions and exhibiting difficulties in the case of the difficult rastrigin variants. On the linear slope, the original CMSA-ES shows fewer expected function evaluations for smaller dimensions which starts to change when the dimensionality is increased. For the functions with ids 6–9, with moderate condition numbers, there are advantages to the CMSA-shr-ES, with the exception of the rotated rosenbrock (9). Most of the functions with high conditioning, ids 10–12, and 14, can be solved by both variants with slightly better results for the CMSA-ES. The sharp ridge (id 13) appears as problematic, with the CMSA-shr-ES showing fewer fitness evaluations for hitting the various precisions goals in Table 1.

Interestingly, the CMSA-shr variant seems to perform better for the difficult multimodal functions, e.g., Gallaghers 101 peak function, a finding which should be explored in more detail. The results for the last two multimodal functions can be explained in part in that the computing resources were insufficient for the optimization. Even the best performing algorithms from the BBOB workshop needed more resources than we used in our experiments.

Further experiments will be conducted in order to shed more light on the behavior. Special attention will be given to the choice of the shrinkage factor, since its setting is unlikely to be optimal and may have influenced the outcome strongly. Furthermore, the question remains whether the population size should be increased for the self-adaptation process. Also, larger search space dimensionalities than $N = 20$ are of interest.



CMSA-shr-ES, 5-D

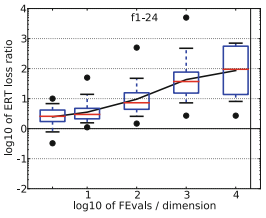


CMSA-shr-ES, 20-D

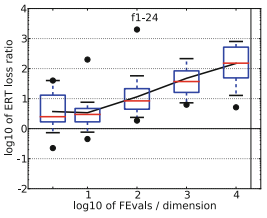
f_1–f_{24} in 5-D, maxFE/D=20000						
#FEs/D	best	10%	25%	**med**	75%	90%
2	0.31	0.97	1.6	2.6	3.9	8.5
10	1.3	1.5	2.0	3.1	4.3	14
100	1.6	2.3	5.3	7.7	16	47
1e3	2.8	7.3	17	49	79	2.7e2
1e4	2.8	6.0	14	66	1.4e2	7.1e2
1e5	2.8	6.0	14	93	5.6e2	2.7e3
RL_{US}/D	2e4	2e4	2e4	2e4	2e4	2e4

f_1–f_{24} in 20-D, maxFE/D=20000						
#FEs/D	best	10%	25%	**med**	75%	90%
2	0.39	0.68	1.4	2.4	13	40
10	0.49	0.86	1.5	2.8	5.1	27
100	1.2	2.3	3.0	5.1	22	59
1e3	6.3	11	14	25	56	2.2e2
1e4	5.9	12	34	81	4.1e2	8.1e2
1e5	5.9	12	42	2.9e2	7.8e2	6.7e3
RL_{US}/D	2e4	2e4	2e4	2e4	2e4	2e4

Fig. 1. The CMSA-shr-ES. ERT loss ratio (in number of f-evaluations divided by dimension) divided by the best ERT seen in GECCO-BBOB-2009 for the target ftarget, or, if the best algorithm reached a better target within the budget, the budget divided by the best ERT. Line: geometric mean. Box-Whisker error bar: 25–75 %-ile with median (box), 10–90 %-ile (caps), and minimum and maximum ERT loss ratio (points). The vertical line gives the maximal number of function evaluations in a single trial in this function subset.



5-D



20-D

f_1–f_{24} in 5-D, maxFE/D=20000						
#FEs/D	best	10%	25%	**med**	75%	90%
2	0.33	0.78	1.6	2.5	3.9	7.1
10	1.1	1.5	2.0	2.8	4.6	18
100	1.5	2.5	4.3	6.8	15	57
1e3	2.7	6.9	15	34	75	5.7e2
1e4	2.7	8.0	12	84	4.7e2	7.1e2
1e5	2.7	8.0	12	68	5.9e2	1.2e3
RL_{US}/D	2e4	2e4	2e4	2e4	2e4	2e4

f_1–f_{24} in 20-D, maxFE/D=20000						
#FEs/D	best	10%	25%	**med**	75%	90%
2	0.23	0.73	1.6	2.3	13	40
10	0.45	0.75	1.4	2.8	4.6	27
100	1.9	2.4	3.9	8.1	21	59
1e3	6.3	7.2	15	35	81	2.4e2
1e4	5.1	12	40	1.5e2	5.0e2	8.1e2
1e5	5.1	12	52	3.8e2	8.6e2	6.7e3
RL_{US}/D	2e4	2e4	2e4	2e4	2e4	2e4

Fig. 2. The CMSA-ES. ERT loss ratio (in number of f-evaluations divided by dimension) divided by the best ERT seen in GECCO-BBOB-2009 for the target ftarget, or, if the best algorithm reached a better target within the budget, the budget divided by the best ERT. Line: geometric mean. Box-Whisker error bar: 25–75 %-ile with median (box), 10–90 %-ile (caps), and minimum and maximum ERT loss ratio (points). The vertical line gives the maximal number of function evaluations in a single trial in this function subset.

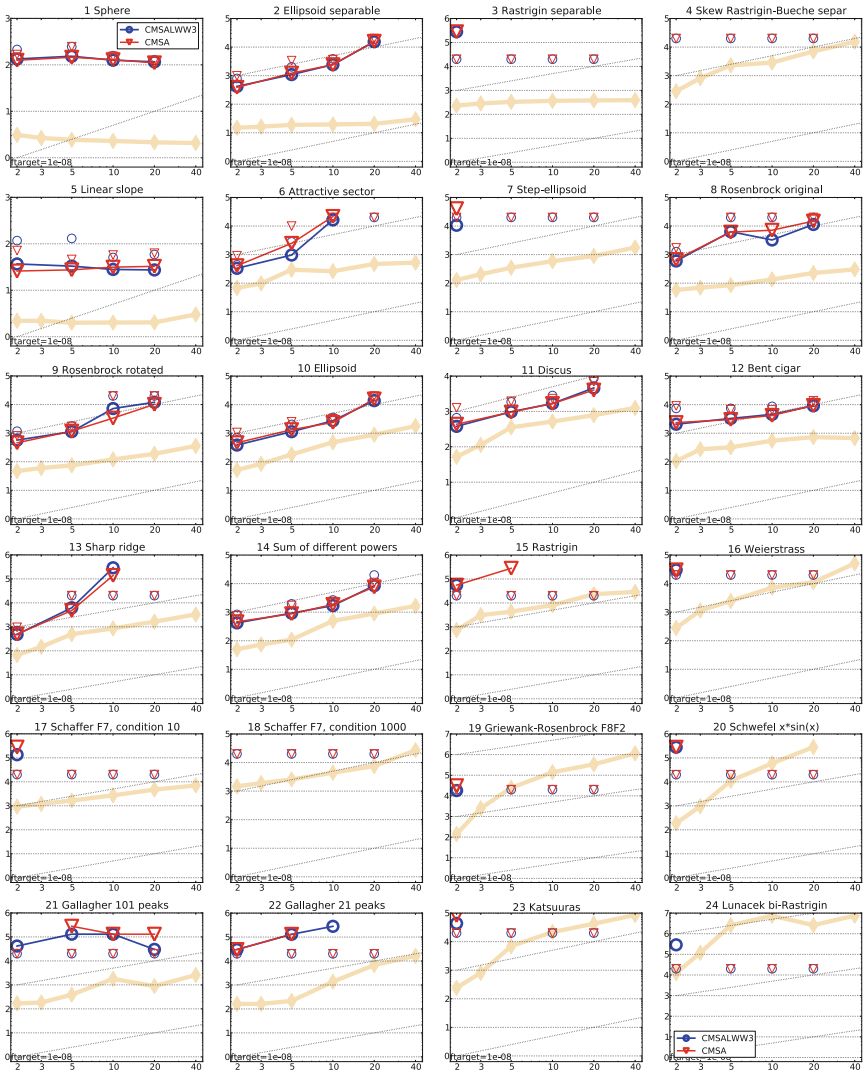


Fig. 3. Expected running time ERT in number of f-evaluations) divided by dimension for target function value as $\log_{10}$ values versus dimension. Different symbols correspond to different algorithms given in the legend of f_1 and f_{24}. Light symbols give the maximum number of function evaluations from the longest trial divided by dimension. Horizontal lines give linear scaling, slanted dotted lines give quadratic scaling. Black stars indicate statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions (six). Legend: .1: CMSA-S is CMSA-shr-ES and 2: CMSA is CMSA-ES.

Table 1. ERT in number of function evaluations divided by the best ERT measured during BBOB-2009 given in the respective first row with the central 80 % range divided by two in brackets for different Δf values. #succ is the number of trials that reached the final target $f_{\text{opt}}+10^{-8}$. 1:CMSA-S is CMSA-shr-ES and 2:CMSA is CMSA-ES. Bold entries are statistically significantly better compared to the other algorithm, with $p = 0.05$ or $p = 10^{-k}$ where $k \in \{2, 3, 4, \ldots\}$ is the number following the $\star$ symbol, with Bonferroni correction of 48. A $\downarrow$ indicates the same tested against the best BBOB-2009. 1: CMSA-S is CMSA-shr-ES and 2: CMSA is CMSA-ES.

5-D

Δf	1e+1	1e-1	1e-3	1e-5	1e-7	#succ
$\mathbf{f_1}$	11	12	12	12	12	15/15
1: CMSA-S	2.4(2)	12(4)	25(6)	38(8)	53(8)	15/15
2: CMSA	2.4(2)	12(4)	26(7)	40(8)	54(15)	15/15
$\mathbf{f_2}$	83	88	90	92	94	15/15
1: CMSA-S	26(21)	44(24)	53(30)	56(30)	57(29)	15/15
2: CMSA	19(15)	47(19)	54(23)	60(23)	63(30)	15/15
$\mathbf{f_3}$	716	1637	1646	1650	1654	15/15
1: CMSA-S	22(70)	∞	∞	∞	∞*1.0e5*	0/15
2: CMSA	94(140)	∞	∞	∞	∞*1.0e5*	0/15
$\mathbf{f_4}$	809	1688	1817	1886	1903	15/15
1: CMSA-S	46(62)	∞	∞	∞	∞*1.0e5*	0/15
2: CMSA	186(247)	∞	∞	∞	∞*1.0e5*	0/15
$\mathbf{f_5}$	10	10	10	10	10	15/15
1: CMSA-S	8.6(3)	16(6)	17(6)	17(6)	17(6)	15/15
2: CMSA	8.2(3)	13(6)	14(6)	14(6)	14(6)	15/15
$\mathbf{f_6}$	114	281	580	1038	1332	15/15
1: CMSA-S	2.0(1.0)	2.6(1.0)	3.2(3)	2.5(1)	3.2(3)	15/15
2: CMSA	1.6(0.9)	2.2(0.9)	3.2(3)	3.7(3)	5.8(4)	15/15
$\mathbf{f_7}$	24	1171	1572	1572	1597	15/15
1: CMSA-S	57(5)	1217(1281)	∞	∞	∞*1.0e5*	0/15
2: CMSA	123(3)	∞	∞	∞	∞*1.0e5*	0/15
$\mathbf{f_8}$	73	336	391	410	422	15/15
1: CMSA-S	2.6(1)	88(152)	79(133)	76(125)	75(122)	12/15
2: CMSA	3.0(1)	85(151)	77(131)	74(124)	72(122)	12/15
$\mathbf{f_9}$	35	214	300	335	369	15/15
1: CMSA-S	2.7(2)	15(9)	15(8)	15(7)	15(7)	15/15
2: CMSA	2.9(1)	17(9)	16(7)	16(7)	15(7)	15/15
$\mathbf{f_{10}}$	349	574	626	829	880	15/15
1: CMSA-S	6.7(6)	6.7(5)	7.6(5)	6.3(4)	6.3(3)	15/15
2: CMSA	7.2(4)	7.4(5)	8.3(5)	7.3(5)	7.4(4)	15/15
$\mathbf{f_{11}}$	143	763	1177	1467	1673	15/15
1: CMSA-S	8.8(8)	4.7(2)	3.5(2)	3.1(1)	2.9(1)	15/15
2: CMSA	7.3(6)	4.0(3)	3.3(2)	3.0(2)	2.8(1)	15/15
$\mathbf{f_{12}}$	108	371	461	1303	1494	15/15
1: CMSA-S	14(17)	20(22)	22(19)	10(9)	11(9)	15/15
2: CMSA	9.1(2)	17(21)	19(18)	9.4(9)	10(8)	15/15
$\mathbf{f_{13}}$	132	250	1310	1752	2255	15/15
1: CMSA-S	12(13)	25(16)	7.1(4)	8.2(3)	13(16)	14/15
2: CMSA	5.0(7)	27(25)	7.2(5)	7.2(4)	7.0(3)	14/15
$\mathbf{f_{14}}$	10	58	139	251	476	15/15
1: CMSA-S	1.6(2)	2.9(1)	5.7(2)	8.6(3)	8.1(4)	15/15
2: CMSA	1.3(2)	3.4(1)	5.4(2)	8.7(4)	8.3(4)	15/15
$\mathbf{f_{15}}$	511	19369	20073	20769	21359	14/15
1: CMSA-S	15(1)	∞	∞	∞	∞*1.0e5*	0/15
2: CMSA	132(196)	72(80)	70(77)	68(77)	66(77)	1/15
$\mathbf{f_{16}}$	120	2662	10449	11644	12095	15/15
1: CMSA-S	3.1(3)	526(620)	∞	∞	∞*1.0e5*	0/15
2: CMSA	3.3(3)	245(282)	∞	∞	∞*1.0e5*	0/15
$\mathbf{f_{17}}$	5.2	899	3669	6351	7934	15/15
1: CMSA-S	3.3(5)	98(167)	∞	∞	∞*1.0e5*	0/15
2: CMSA	2.6(3)	223(278)	177(218)	∞	∞*1.0e5*	0/15
$\mathbf{f_{18}}$	103	3968	9280	10905	12469	15/15
1: CMSA-S	70(0.7)	101(126)	∞	∞	∞*1.0e5*	0/15
2: CMSA	0.90(0.7)	22(38)	151(172)	∞	∞*1.0e5*	0/15
$\mathbf{f_{19}}$	1	242	1.2e5	1.2e5	1.2e5	15/15
1: CMSA-S	3.1(2)	2730(3097)	∞	∞	∞*1.0e5*	0/15
2: CMSA	3.0(3)	1186(1448)	∞	∞	∞*1.0e5*	0/15
$\mathbf{f_{20}}$	16	38111	54470	54861	55313	14/15
1: CMSA-S	1.6(1)	∞	∞	∞	∞*1.0e5*	0/15
2: CMSA	1.9(1)	∞	∞	∞	∞*1.0e5*	0/15
$\mathbf{f_{21}}$	41	1674	1705	1729	1757	14/15
1: CMSA-S	377(1220)	388(448)	382(469)	376(434)	370(455)	2/15
2: CMSA	612(1220)	836(970)	821(909)	810(882)	797(939)	1/15
$\mathbf{f_{22}}$	71	938	1008	1040	1068	14/15
1: CMSA-S	514(705)	693(853)	645(793)	626(673)	610(773)	2/15
2: CMSA	513(705)	694(852)	646(794)	626(721)	610(703)	2/15
$\mathbf{f_{23}}$	3.0	14249	31654	33030	34256	15/15
1: CMSA-S	2.0(2)	∞	∞	∞	∞*1.0e5*	0/15
2: CMSA	3.2(3)	∞	∞	∞	∞*1.0e5*	0/15
$\mathbf{f_{24}}$	1622	6.4e6	9.6e6	1.3e7	1.3e7	3/15
1: CMSA-S	33(62)	∞	∞	∞	∞*1.0e5*	0/15
2: CMSA	17(31)	∞	∞	∞	∞*1.0e5*	0/15

20-D

Δf	1e+1	1e-1	1e-3	1e-5	1e-7	#succ
$\mathbf{f_1}$	43	43	43	43	43	15/15
1: CMSA-S	4.7(1)	15(3)	27(3)	38(2)	48(3)	15/15
2: CMSA	4.7(0.9)	14(2)	24(2)	35(3)	46(3)	15/15
$\mathbf{f_2}$	385	387	390	391	393	15/15
1: CMSA-S	331(232)	567(121)	647(81)	704(91)	755(82)	15/15
2: CMSA	332(106)	561(127)	637(154)	687(182)	786(186)	14/15
$\mathbf{f_3}$	5066	7635	7643	7646	7651	15/15
1: CMSA-S	∞	∞	∞	∞	∞*4.0e5*	0/15
2: CMSA	∞	∞	∞	∞	∞*4.0e5*	0/15
$\mathbf{f_4}$	4722	7666	7700	7758	1.4e5	9/15
1: CMSA-S	∞	∞	∞	∞	∞*4.0e5*	0/15
2: CMSA	∞	∞	∞	∞	∞*4.0e5*	0/15
$\mathbf{f_5}$	41	41	41	41	41	15/15
1: CMSA-S	10(3)	13(4)	13(4)	13(4)	13(4)	15/15
2: CMSA	11(3)	16(4)	16(4)	16(4)	16(4)	15/15
$\mathbf{f_6}$	1296	3413	5220	6728	8409	15/15
1: CMSA-S	1.6(1)	6.4(10)	63(78)	388(475)	682(690)	0/15
2: CMSA	1.7(0.4)	7.0(6)	231(278)	∞	∞*4.0e5*	0/15
$\mathbf{f_7}$	1351	9503	16524	16524	16969	15/15
1: CMSA-S	649(862)	∞	∞	∞	∞*4.0e5*	0/15
2: CMSA	299(444)	∞	∞	∞	∞*4.0e5*	0/15
$\mathbf{f_8}$	2039	4040	4219	4371	4484	15/15
1: CMSA-S	25(9)	49(50)	50(47)	50(49)	50(51)	13/15
2: CMSA	23(14)	67(62)	67(60)	66(58)	66(57)	11/15
$\mathbf{f_9}$	1716	3277	3455	3594	3727	15/15
1: CMSA-S	27(10)	63(64)	64(60)	64(57)	64(54)	13/15
2: CMSA	27(7)	52(8)	54(8)	54(9)	54(8)	14/15
$\mathbf{f_{10}}$	7413	10735	14920	17073	17476	15/15
1: CMSA-S	17(6)	19(5)	15(4)	15(3)	15(3)	15/15
2: CMSA	17(10)	21(5)	17(5)	16(5)	18(5)	14/15
$\mathbf{f_{11}}$	1002	6278	9762	12285	14831	15/15
1: CMSA-S	14(4)	4.1(1)	4.2(1)	5.1(3)	5.5(3)	15/15
2: CMSA	13(5)	3.6(1)	3.8(1.0)	4.6(2)	4.8(1)	15/15
$\mathbf{f_{12}}$	1042	2740	4140	12407	13827	15/15
1: CMSA-S	8.5(16)	23(22)	26(15)	12(5)	12(4)	15/15
2: CMSA	5.8(15)	23(25)	26(16)	12(6)	13(6)	15/15
$\mathbf{f_{13}}$	652	2751	18749	24455	30201	15/15
1: CMSA-S	46(0.8)	583(727)	299(325)	∞	∞*4.0e5*	0/15
2: CMSA	704(921)	946(1163)	∞	∞	∞*4.0e5*	0/15
$\mathbf{f_{14}}$	75	304	932	1648	15661	15/15
1: CMSA-S	2.0(0.6)	2.6(0.5)	5.9(2)	16(5)	6.8(2)	14/15
2: CMSA	2.0(0.6)	2.6(0.7)	6.2(0.8)	16(5)	6.0(2)	15/15
$\mathbf{f_{15}}$	30378	3.1e5	3.2e5	4.5e5	4.6e5	15/15
1: CMSA-S	∞	∞	∞	∞	∞*4.0e5*	0/15
2: CMSA	∞	∞	∞	∞	∞*4.0e5*	0/15
$\mathbf{f_{16}}$	1384	77015	1.9e5	2.0e5	2.2e5	15/15
1: CMSA-S	2.9(3)	∞	∞	∞	∞*4.0e5*	0/15
2: CMSA	107(146)	∞	∞	∞	∞*4.0e5*	0/15
$\mathbf{f_{17}}$	63	4005	30677	56288	80472	15/15
1: CMSA-S	1.1(1)	1399(1498)	∞	∞	∞*4.0e5*	0/15
2: CMSA	1.1(1)	650(749)	∞	∞	∞*4.0e5*	0/15
$\mathbf{f_{18}}$	621	19561	67569	1.3e5	1.5e5	15/15
1: CMSA-S	0.96(0.5)	∞	∞	∞	∞*4.0e5*	0/15
2: CMSA	1.0(0.4)	∞	∞	∞	∞*4.0e5*	0/15
$\mathbf{f_{19}}$	1	3.4e5	6.2e6	6.7e6	6.7e6	15/15
1: CMSA-S	4.8(4)	∞	∞	∞	∞*4.0e5*	0/15
2: CMSA	6.4(9)	∞	∞	∞	∞*4.0e5*	0/15
$\mathbf{f_{20}}$	82	3.1e6	5.5e6	5.6e6	5.6e6	14/15
1: CMSA-S	2.0(0.6)	∞	∞	∞	∞*4.0e5*	0/15
2: CMSA	2.0(0.9)	∞	∞	∞	∞*4.0e5*	0/15
$\mathbf{f_{21}}$	561	14103	14643	15567	17589	15/15
1: CMSA-S	51(1.0)	43(57)	41(55)	39(51)	34(45)	6/15
2: CMSA	179(356)	184(227)	178(191)	167(193)	148(171)	2/15
$\mathbf{f_{22}}$	467	23491	24948	26847	1.3e5	12/15
1: CMSA-S	430(857)	∞	∞	∞	∞*4.0e5*	0/15
2: CMSA	429(857)	∞	∞	∞	∞*4.0e5*	0/15
$\mathbf{f_{23}}$	3.2	67457	4.9e5	8.1e5	8.4e5	15/15
1: CMSA-S	2.1(2)	∞	∞	∞	∞*4.0e5*	0/15
2: CMSA	2.3(3)	∞	∞	∞	∞*4.0e5*	0/15
$\mathbf{f_{24}}$	1.3e6	5.2e7	5.2e7	5.2e7	5.2e7	3/15
1: CMSA-S	∞	∞	∞	∞	∞*4.0e5*	0/15
2: CMSA	∞	∞	∞	∞	∞*4.0e5*	0/15

5 Conclusions

Evolution strategies are well performing variants of evolutionary algorithms used in continuous optimization. They ultilize normally distributed mutations as their main search procedure. Their performance depends on the control of the mutation process which is governed by adapting step-sizes and covariance matrices. One possible improvement concerns the covariance matrix adaptation which makes use of the sample covariance matrix. In statistical research, this estimate has been identified as not agreeing well with the true covariance for the case of large dimensional spaces and small sample sizes, or more correctly for sample sizes that do not increase sufficiently fast with the dimensionality.

While modern approaches for covariance matrix adaptation correct the estimate, the question arises whether the performance of these evolutionary algorithms may be further improved by applying other estimators for the covariance.

This paper took a closer look at covariance estimation in evolution strategies and provided a comparison with approaches in modern statistics. Furthermore, it introduced and discussed new adaptation schemes for use in optimization. In cases, where the fitness function requires highly different eigenvalues and a rotation other than the cartesian coordinate system. Therefore, a switch towards the eigenspace of the covariance matrix was proposed in this paper and investigated in experiments on the BBOB test suite. While work remains to be done, this paper provided an important first step on the way.

References

1. Audet, C.: A survey on direct search methods for blackbox optimization and their applications. In: Pardalos, P.M., Rassias, T.M. (eds.) Mathematics without Boundaries: Surveys in Interdisciplinary Research. Springer, New York (2013). Also Les Cahiers du GERAD G-2012-53 (2012)
2. Bai, Z.D., Silverstein, J.W.: No eigenvalues outside the support of the limiting spectral distribution of large-dimensional sample covariance matrices. Ann. Probab. **26**(1), 316–345 (1998)
3. Beyer, H.G., Meyer-Nieberg, S.: Self-adaptation of evolution strategies under noisy fitness evaluations. Genet. Program. Evolvable Mach. **7**(4), 295–328 (2006)
4. Beyer, H.G., Schwefel, H.P.: Evolution strategies: a comprehensive introduction. Nat. Comput. **1**(1), 3–52 (2002)
5. Beyer, H.-G., Sendhoff, B.: Covariance matrix adaptation revisited – the CMSA evolution strategy –. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 123–132. Springer, Heidelberg (2008)
6. Chen, X., Wang, Z., McKeown, M.: Shrinkage-to-tapering estimation of large covariance matrices. IEEE Trans. Signal Process **60**(11), 5640–5656 (2012)
7. Chen, Y., Wiesel, A., Eldar, Y.C., Hero, A.O.: Shrinkage algorithms for MMSE covariance estimation. IEEE Trans. Signal Process. **58**(10), 5016–5029 (2010)
8. Sarkar, M., Theuwissen, A.: Introduction. In: Sarkar, M., Theuwissen, A. (eds.) A Biologically Inspired CMOS Image Sensor. SCI, vol. 461, pp. 1–14. Springer, Heidelberg (2013)
9. Dong, W., Yao, X.: Covariance matrix repairing in gaussian based EDAs. In: IEEE Congress on, Evolutionary Computation, CEC 2007, pp. 415–422 (2007)

10. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Natural Computing Series. Springer, Berlin (2003)
11. 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 (2010) 2009/22
12. 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)
13. Hansen, N., Auger, A., Finck, S., Ros, R.: Real-parameter black-box optimization benchmarking 2012: experimental setup. Technical report, INRIA (2012). http://coco.gforge.inria.fr/bbob2012-downloads
14. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation **9**(2), 159–195 (2001)
15. Hansen, N., Auger, A., Ros, R., Finck, S., Pošík, P.: Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In: Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO 2010, pp. 1689–1696. ACM, New York (2010). http://doi.acm.org/10.1145/1830761.1830790
16. Huber, P.J.: Robust Statistics. Wiley, New York (1981)
17. Ledoit, O., Wolf, M.: A well-conditioned estimator for large dimensional covariance matrices. J. Multivar. Anal. Arch. **88**(2), 265–411 (2004)
18. Ledoit, O., Wolf, M.: Non-linear shrinkage estimation of large dimensional covariance matrices. Ann. Stat. **40**(2), 1024–1060 (2012)
19. Meyer-Nieberg, S., Beyer, H.G.: Self-adaptation in evolutionary algorithms. In: Lobo, F., Lima, C., Michalewicz, Z. (eds.) Parameter Setting in Evolutionary Algorithms, pp. 47–76. Springer Verlag, Heidelberg (2007)
20. Rechenberg, I.: Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog Verlag, Stuttgart (1973)
21. Schäffer, J., Strimmer, K.: A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. Stat. Appl. Genet. Mol. Biol. **4**(1), Article 32 (2005)
22. Schwefel, H.P.: Numerical Optimization of Computer Models. Wiley, Chichester (1981)
23. Stein, C.: Inadmissibility of the usual estimator for the mean of a multivariate distribution. In: Proceedings 3rd Berkeley Symposium Mathematical Statistics and Probability, vol. 1, pp. 197–206. Berkeley, CA (1956)
24. Stein, C.: Estimation of a covariance matrix. In: Rietz Lecture, 39th Annual Meeting. IMS, Atlanta, GA (1975)
25. Thomaz, C.E., Gillies, D., Feitosa, R.: A new covariance estimate for bayesian classifiers in biometric recognition. IEEE Trans. Circuits Syst. Video Technol. **14**(2), 214–223 (2004)