Simplify Your Covariance Matrix Adaptation Evolution Strategy

Hans-Georg Beyer and Bernhard Sendhoff *Senior Member, IEEE*

***Abstract*—The standard Covariance Matrix Adaptation Evolution Strategy (CMA-ES) comprises two evolution paths, one for the learning of the mutation strength and one for the rank-1 update of the covariance matrix. In this paper it is shown that one can approximately transform this algorithm in such a manner that one of the evolution paths and the covariance matrix itself disappear. That is, the covariance update and the covariance matrix square root operations are no longer needed in this novel so-called Matrix Adaptation (MA) ES. The MA-ES performs nearly as well as the original CMA-ES. This is shown by empirical investigations considering the evolution dynamics and the empirical expected runtime on a set of standard test functions. Furthermore, it is shown that the MA-ES can be used as search engine in a Bi-Population (BiPop) ES. The resulting BiPop-MA-ES is benchmarked using the BBOB COCO framework and compared with the performance of the CMA-ES-v3.61 production code. It is shown that this new BiPop-MA-ES – while algorithmically simpler – performs nearly equally well as the CMA-ES-v3.61 code.**

***Index Terms*—Matrix Adaptation Evolution Strategies, Black Box Optimization Benchmarking.**

I. Introduction

THE Covariance Matrix Adaptation Evolution Strategy (CMA-ES) has received considerable attention as an algorithm for unconstrained real-parameter optimization, i.e. solving the problem

$$\hat{\mathbf{y}} = \arg\operatorname{opt}_{\mathbf{y}\in\mathbb{R}^N}\big(f(\mathbf{y})\big), \tag{1}$$

since its publication in its fully developed form including rank-μ update in [1]. While there are proposals regarding modifications of the original CMA-ES such as the $(1+1)$-CMA-ES [2] and its latest extension in [19], or the active CMA-ES [3], or the CMSA-ES [4], the design presented in [1] has only slightly changed during the years and state-of-the-art presentations such as [5] still rely on that basic design. Taking the advent of the so-called Natural Evolution Strategies (NES) into account, e.g. [6], one sees that even a seemingly principled design paradigm such as the information gain constrained evolution on statistical manifolds did not cause a substantial change in the CMA-ES design. Actually, in order to get NES competitive, those designers had to borrow from and rely on the ideas/principles empirically found by the CMA-ES designers. Yet, it is somehow remarkable why the basic framework of CMA-ES has not undergone a deeper scrutiny to call the algorithm's peculiarities, such as the covariance matrix adaptation, the evolution path and the cumulative step-size adaptation (CSA) into question. Notably in [4] a first attempt has been made to replace the CSA mutation strength control by the classical mutative self-adaptation. While getting rid of any kind of evolution path statistics and thus yielding a much simpler strategy[1], the resulting rank-μ update strategy, the so-called Covariance Matrix Self-Adaptation CMSA (CMSA-ES) does not fully reach the performance of the original CMA-ES in the case of small population sizes. Furthermore, some of the reported performance advantages in [4] were due to a wrongly implemented stalling of the covariance matrix update in the CMA-ES implementation used.

Meanwhile, our theoretical understanding of the evolution dynamics taking place in CMSA- and CMA-ES has advanced. Basically two reasons for the superior CMA-ES performance can be identified in the case of small population sizes:

- *Concentrate evolution along a predicted direction.*
  Using the evolution path information for the covariance matrix update (this rank-1 update was originally used in the first CMA-ES version [7] as the only update) can be regarded as some kind of *time series prediction* of the evolution of the parent in the $\mathbb{R}^N$ search space. That is, the evolution path carries the information in which direction the search steps were most successful. This directional information may be regarded as the most promising one and can be used in the CMA-ES to *shrink* the evolution of the covariance matrix.[2] As a result, optimization in landscape topologies with a predominant search direction (such as the Cigar test function or the Rosenbrock function) can benefit from the path cumulation information.
- *Increased mutation strength.*
  As has been shown by analyzing the dynamic of the CSA on the ellipsoid model in [10], the path-length based mutation strength control yields larger steady state mutation strengths up to a factor of μ (parental population size) compared to ordinary self-adaptive mutation strength control (in the case of non-noisy objective functions). Due to the larger mutation strengths realized, the CSA-based approach can better take advantage of the genetic

H.-G. Beyer is with the Research Center Process and Product Engineering at the Vorarlberg University of Applied Sciences, Dornbirn, Austria, Email: Hans-Georg.Beyer@fhv.at

B. Sendhoff is with the Honda Research Institute Europe GmbH, Offenbach/Main, Germany, Email: Bernhard.Sendhoff@honda-ri.de

[1]By simplicity we mean a reduced complexity of code and especially a decreased number of strategy specific parameters to be fixed by the algorithm designer. Furthermore, the remaining strategy specific parameters have been determined rather by first principles than empirical parameter tuning studies.

[2]Shrinking is a well-known technique for covariance matrix estimation in order to control the undesired growth of the covariance matrix [8]. Considering the covariance matrix update in the CMA-ES from the perspective of shrinking has been done first by Meyer-Nieberg and Kropat [9].

repair effect provided by the intermediate (or weighted) recombination.

The performance advantage vanishes, however, when the population size is chosen sufficiently large. However, considering the CMA-ES as a general purpose algorithm, the strategy should cope with both small and large population sizes. Therefore, it seems that both the rank-1 and the rank-μ update are needed. This still calls in question whether it is necessary to have two evolution paths and an update of the covariance matrix at all.

In this paper we will show that one can drop one of the evolution paths and remove the covariance matrix totally, thus removing a "$\mathbf{p}$" and the "$\mathbf{C}$" from the CMA-ES *without* significantly worsening the performance of the resulting "Matrix Adaptation" (MA)-ES. As a byproduct of the analysis, which is necessary to derive the MA-ES from the CMA-ES, one can draw a clearer picture of the CMA-ES, which is often regarded as a rather complicated evolutionary algorithm.

This work is organized as follows. First, the CMA-ES algorithm is presented in a manner that allows for a simpler analysis of the pseudocode lines. As the next step, the $\mathbf{p}$-evolution path (realized by line C13, Fig. 1, see Section II for details) will be shown to be similar to that of the $\mathbf{s}$-evolution path (realized by line C12, Fig. 1). Both paths can be approximately transformed (asymptotically exact for search space dimensionality $N \to \infty$) into each other by a linear transformation where the transformation matrix is just the matrix square root $\mathbf{M}$ of the covariance matrix $\mathbf{C}$. After removing the $\mathbf{p}$-path from the CMA-ES, the update of the $\mathbf{C}$ matrix will be replaced by transforming the covariance learning to a direct learning of the $\mathbf{M}$ matrix. Thus, one needs no longer the covariance matrix and operations such as Cholesky decomposition or spectral decomposition to calculate the matrix square root. This simplifies the resulting ES considerably both from viewpoint of the algorithmic "complexity" and the numerical algebra operations needed. Furthermore, the resulting $\mathbf{M}$-update allows for a new interpretation of the CMA-ES working principles. However, since the derivations rely on assumptions that are only asymptotically exact for search space dimensionality $N \to \infty$, numerical experiments are provided to show that the novel MA-ES performs nearly equally well as the original CMA-ES. Additionally, the CMA-ES and MA-ES will be used as "search engines" in the BiPop-ES framework [11]. The resulting BiPop-MA-ES will be compared with the original BiPop-CMA-ES and the CMA-ES v3.61 production code using the BBOB COCO test environment [12]. It will be shown that the MA-ES performs well in the BiPop-ES framework. The paper concludes with a summary section.

$(\mu/\mu_w, \lambda)$-CMA-ES

Initialize$\big(\mathbf{y}^{(0)}, \sigma^{(0)}, g := 0, \mathbf{p}^{(0)} := \mathbf{0},$
$\mathbf{s}^{(0)} := \mathbf{0}, \mathbf{C}^{(0)} := \mathbf{I}\big)$ (C1)

Repeat (C2)

$\mathbf{M}^{(g)} := \sqrt{\mathbf{C}^{(g)}}$ (C3)

For $l := 1$ **To** λ (C4)

$\tilde{\mathbf{z}}_l^{(g)} := \boldsymbol{\mathcal{N}}_l(\mathbf{0}, \mathbf{I})$ (C5)

$\tilde{\mathbf{d}}_l^{(g)} := \mathbf{M}^{(g)} \tilde{\mathbf{z}}_l^{(g)}$ (C6)

$\tilde{\mathbf{y}}_l^{(g)} := \mathbf{y}^{(g)} + \sigma^{(g)} \tilde{\mathbf{d}}_l^{(g)}$ (C7)

$\tilde{f}_l^{(g)} := f\big(\tilde{\mathbf{y}}_l^{(g)}\big)$ (C8)

End (C9)

SortOffspringPopulation (C10)

$\mathbf{y}^{(g+1)} := \mathbf{y}^{(g)} + \sigma^{(g)} \left\langle \tilde{\mathbf{d}}^{(g)} \right\rangle_w$ (C11)

$\mathbf{s}^{(g+1)} := (1 - c_s)\mathbf{s}^{(g)} + \sqrt{\mu_{\mathrm{eff}} c_s (2 - c_s)} \left\langle \tilde{\mathbf{z}}^{(g)} \right\rangle_w$ (C12)

$\mathbf{p}^{(g+1)} := (1 - c_p)\mathbf{p}^{(g)} + \sqrt{\mu_{\mathrm{eff}} c_p (2 - c_p)} \left\langle \tilde{\mathbf{d}}^{(g)} \right\rangle_w$ (C13)

$\mathbf{C}^{(g+1)} := (1 - c_1 - c_w)\mathbf{C}^{(g)} + c_1 \mathbf{p}^{(g+1)} \big(\mathbf{p}^{(g+1)}\big)^{\mathrm{T}}$
$+ c_w \left\langle \tilde{\mathbf{d}}^{(g)} \big(\tilde{\mathbf{d}}^{(g)}\big)^{\mathrm{T}} \right\rangle_w$ (C14)

$\sigma^{(g+1)} := \sigma^{(g)} \exp\left[\frac{c_s}{d_\sigma} \left(\frac{\|\mathbf{s}^{(g+1)}\|}{\mathrm{E}\,[\|\boldsymbol{\mathcal{N}}(\mathbf{0}, \mathbf{I})\|]} - 1 \right) \right]$ (C15)

$g := g + 1$ (C16)

Until(termination condition(s) fulfilled) (C17)

Fig. 1. Pseudocode of the CMA-ES with rank > 1 weighted covariance matrix update.

II. The Standard CMA-ES

Although not immediately obvious, the $(\mu/\mu_w, \lambda)$-CMA-ES displayed in Fig. 1 represents basically the one introduced in [1] and discussed in its current form in [5]. While the exposition in [5] contains additional tweaks regarding strategy specific parameters, its essence has been condensed into the pseudocode of Fig. 1. We will discuss the basic operations performed in Fig. 1 and show that those lines are mathematically equivalent to the original CMA-ES [1]. In order to have a clear notion of generational changes, a generation counter $^{(g)}$ is used even though it is not necessarily needed for the functioning in real CMA-ES implementations.

A CMA-ES generation cycle (also referred to as a generation) is performed within the repeat-until-loop (C2–C17). A number of λ offspring (labeled by a tilde on top of the symbols and indexed by the subscript l) is generated within the for-loop (C4–C9) where the parental state $\mathbf{y}^{(g)} \in \mathbb{R}^N$ is mutated according to a normal distribution yielding offspring $\tilde{\mathbf{y}}^{(g+1)} \sim \boldsymbol{\mathcal{N}}\big(\mathbf{y}^{(g)}, (\sigma^{(g)})^2 \mathbf{C}^{(g)}\big)$ with mean $\mathbf{y}^{(g)}$ and covariance $(\sigma^{(g)})^2 \mathbf{C}^{(g)}$. This is technically done by generating at first an isotropically iid normally distributed N-dimensional vector $\tilde{\mathbf{z}}_l^{(g)}$ in line (C5). This vector is transformed by $\mathbf{M}^{(g)}$ into a (direction) vector $\tilde{\mathbf{d}}_l^{(g)}$ where the transformation matrix $\mathbf{M}^{(g)}$ must obey the condition

$$\mathbf{M}^{(g)} \big(\mathbf{M}^{(g)}\big)^{\mathrm{T}} = \mathbf{C}^{(g)}. \tag{2}$$

Therefore, $\mathbf{M}^{(g)}$ may be regarded as a "square root" of the matrix $\mathbf{C}^{(g)}$, explaining the meaning of line (C3). Technically,

$\mathbf{M}^{(g)} = \sqrt{\mathbf{C}^{(g)}}$ is calculated in the CMA-ES using Cholesky or spectral value decomposition. After having generated the search direction $\tilde{\mathbf{d}}_l^{(g)}$, it is scaled with the mutation strength $\sigma^{(g)}$[3], thus forming the mutation vector. This mutation is added to the parental state $\mathbf{y}^{(g)}$ in line (C7) yielding the offspring individual's $\tilde{\mathbf{y}}_l^{(g)}$. The fitness $\tilde{f}$ of the offspring is finally calculated in line (C8). After having generated all λ offspring, the population is sorted (ranked) according to the individual fitnesses in line (C10).

The process of ranking is not explicitly displayed, however, it is implicitly covered by the "$m;\lambda$" index notation used (see below) in the calculation of the angular bracket notations $\langle \cdots \rangle$ in lines (C11–C14). Here the m refers to the mth best individual (w.r.t. the objective function values $\tilde{f}_l, l = 1, \ldots, \lambda$) in the population of λ offspring. The angular brackets indicate the process of (weighted) intermediate recombination. Given λ offspring objects $\tilde{\mathbf{x}}_l^{(g)}$, the recombination is defined as

$$\left\langle \tilde{\mathbf{x}}^{(g)} \right\rangle_w := \sum_{m=1}^{\mu} w_m \tilde{\mathbf{x}}_{m;\lambda}^{(g)}, \tag{3}$$

where $\tilde{\mathbf{x}}_{m;\lambda}$ refers to an object (i.e., $\tilde{\mathbf{d}}$, $\tilde{\mathbf{z}}$, and $\tilde{\mathbf{d}}\tilde{\mathbf{d}}^{\mathrm{T}}$, respectively) belonging to the mth best individual (w.r.t. fitness) of the current offspring population comprising λ individuals. Originally, the choice of the weights w_m used in (3) was

$$w_m := \begin{cases} \frac{1}{\mu}, & \text{for } 1 \le m \le \mu, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

also known as intermediate multi-recombination [13]. However, later on and also provided in [5], the weight scheme

$$w_m := \begin{cases} \frac{\ln\left(\frac{\lambda+1}{2}\right) - \ln m}{\sum_{k=1}^{\mu}\left(\ln\left(\frac{\lambda+1}{2}\right) - \ln k\right)}, & \text{for } 1 \le m \le \mu, \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

has been proposed. This proposal is based on the heuristic argument that the best individuals, i.e., those with a small m should have a stronger influence on the weighted sum (3). Note, both weight schemes obey the condition

$$\sum_{m=1}^{\mu} w_m = 1. \tag{6}$$

Calculation of the parent of the new generation $(g+1)$ is done by weighted recombination of the $\mathbf{y}$-vectors of the best μ offspring individuals according to $\mathbf{y}^{(g+1)} := \langle \tilde{\mathbf{y}}^{(g)} \rangle_w$. Using (C6), (C7), and (3), this can alternatively be expressed as

$$\mathbf{y}^{(g+1)} = \left\langle \mathbf{y}^{(g)} + \sigma^{(g)} \tilde{\mathbf{d}}^{(g)} \right\rangle_w = \mathbf{y}^{(g)} + \sigma^{(g)} \left\langle \tilde{\mathbf{d}}^{(g)} \right\rangle_w \tag{7}$$

and is used in line (C11).

The classical CSA path cumulation is done in line (C12). It acts on the isotropically generated $\mathbf{z}$ vectors. This line seems to deviate from the original one provided in [1] and [5]. In the latter publication one finds after adopting the symbols used[4]

$$\begin{aligned} \mathbf{s}^{(g+1)} &= (1-c_s)\mathbf{s}^{(g)} + \\ &\quad + \sqrt{\mu_{\mathrm{eff}} c_s (2-c_s)} \left(\mathbf{C}^{(g)}\right)^{-\frac{1}{2}} \frac{\mathbf{y}^{(g+1)} - \mathbf{y}^{(g)}}{c_{\mathrm{m}} \sigma^{(g)}}. \end{aligned} \tag{8}$$

[3]Note, σ is often referred to as "step-size" of the mutation, however, this is not really true and actually misleading, it is just a scaling factor that allows for a faster adaptation of the length of the mutation.

[4]Note, we also assumed $c_{\mathrm{m}} = 1$, recommended as standard choice in [5].

However, resolving (7) for $\langle \tilde{\mathbf{d}}^{(g)} \rangle_w$, one immediately sees that this is exactly the fraction term in (8). Now, consider the matrix vector product $(\mathbf{C}^{(g)})^{-1/2} \langle \tilde{\mathbf{d}}^{(g)} \rangle_w$ taking (4), (C6), (3), and (C3) into account

$$\begin{aligned} \left(\mathbf{C}^{(g)}\right)^{-\frac{1}{2}} \frac{\mathbf{y}^{(g+1)} - \mathbf{y}^{(g)}}{\sigma^{(g)}} &= \left(\mathbf{C}^{(g)}\right)^{-\frac{1}{2}} \left\langle \tilde{\mathbf{d}}^{(g)} \right\rangle_w \\ &= \sum_{m=1}^{\mu} w_m \left(\mathbf{C}^{(g)}\right)^{-\frac{1}{2}} \tilde{\mathbf{d}}_{m;\lambda}^{(g)} \\ &= \sum_{m=1}^{\mu} w_m \left(\mathbf{C}^{(g)}\right)^{-\frac{1}{2}} \mathbf{M}^{(g)} \tilde{\mathbf{z}}_{m;\lambda}^{(g)} \\ &= \sum_{m=1}^{\mu} w_m \left(\mathbf{C}^{(g)}\right)^{-\frac{1}{2}} \left(\mathbf{C}^{(g)}\right)^{\frac{1}{2}} \tilde{\mathbf{z}}_{m;\lambda}^{(g)} \\ &= \sum_{m=1}^{\mu} w_m \tilde{\mathbf{z}}_{m;\lambda}^{(g)} \\ &= \left\langle \tilde{\mathbf{z}}^{(g)} \right\rangle_w \end{aligned} \tag{9}$$

one sees that (8) of the original CMA-ES is *equivalent* to line (C12). The advantage of line (C12) is, however, that there is absolutely *no need* to perform a back transformation using the inverse of $\mathbf{M}^{(g)}$.

Lines (C12) and (C13) also contain strategy specific constants to be discussed below. Here, we proceed with the second path cumulation used to provide the rank-1 update for the $\mathbf{C}$-matrix in line (C14). Again, line (C13) deviates from the original CMA-ES in [5] where one finds after adopting the symbols used[5]

$$\mathbf{p}^{(g+1)} = (1-c_p)\mathbf{p}^{(g)} + h_\sigma \sqrt{\mu_{\mathrm{eff}} c_p (2-c_p)} \, \frac{\mathbf{y}^{(g+1)} - \mathbf{y}^{(g)}}{c_{\mathrm{m}} \sigma^{(g)}}. \tag{10}$$

As has been shown already above, the fraction term in (10) is exactly $\langle \tilde{\mathbf{d}}^{(g)} \rangle_w$. Thus, we have recovered line (C13).

The $\mathbf{C}$-matrix update takes place in line (C14). The update equation deviates from those given in [5]. The latter reads after adopting the symbols[6]

$$\begin{aligned} \mathbf{C}^{(g+1)} &= \left(1 - c_1 + (1-h_\sigma^2) c_1 c_p (2-c_p)\right) \mathbf{C}^{(g)} \\ &\quad + c_1 \mathbf{p}^{(g+1)} \left(\mathbf{p}^{(g+1)}\right)^{\mathrm{T}} \\ &\quad + c_w \sum_{m=1}^{\mu} w_m \left(\frac{\left(\tilde{\mathbf{y}}_{m;\lambda}^{(g)} - \mathbf{y}^{(g)}\right)}{\sigma^{(g)}} \frac{\left(\tilde{\mathbf{y}}_{m;\lambda}^{(g)} - \mathbf{y}^{(g)}\right)^{\mathrm{T}}}{\sigma^{(g)}} - \mathbf{C}^{(g)} \right). \end{aligned} \tag{11}$$

Also this equation is equivalent to the corresponding line (C14). First note that due to (6) the weights sum of $\mathbf{C}^{(g)}$ in the third line of (11) yields $\mathbf{C}^{(g)}$ and the corresponding c_w can be put into the first parentheses of the rhs of the first line of (11). Now, consider line (C7) and resolve for $\tilde{\mathbf{d}}_l^{(g)}$, one obtains

$$\tilde{\mathbf{d}}_{m;\lambda}^{(g)} = \frac{\tilde{\mathbf{y}}_{m;\lambda}^{(g)} - \mathbf{y}^{(g)}}{\sigma^{(g)}}. \tag{12}$$

[5]We assume $c_{\mathrm{m}} = 1$, recommended as standard choice in [5] and $h_\sigma = 1$.

[6]Note, we also assumed $h_\sigma = 1$.

Thus, one gets for (11)

$$\mathbf{C}^{(g+1)} = (1-c_1-c_w)\mathbf{C}^{(g)} + c_1\mathbf{p}^{(g+1)}\left(\mathbf{p}^{(g+1)}\right)^{\mathrm{T}} + c_w \sum_{m=1}^{\mu} w_m \tilde{\mathbf{d}}_{m;\lambda}^{(g)} \left(\tilde{\mathbf{d}}_{m;\lambda}^{(g)}\right)^{\mathrm{T}} \quad (13)$$

and taking (3) into account one obtains

$$\mathbf{C}^{(g+1)} = (1-c_1-c_w)\mathbf{C}^{(g)} + c_1\mathbf{p}^{(g+1)}\left(\mathbf{p}^{(g+1)}\right)^{\mathrm{T}} + c_w \left\langle \tilde{\mathbf{d}}^{(g)} \left(\tilde{\mathbf{d}}^{(g)}\right)^{\mathrm{T}} \right\rangle_w . \quad (14)$$

This agrees with line (C14). The remaining line (C15) of the CMA-ES pseudocode in Fig. 1 concerns the mutation strength update. It has been directly adopted from [5]. In that line, the actual length of the updated $\mathbf{s}$ evolution path vector is compared with the *expected* length of a random vector with standard normally distributed components. The latter is the expected value $\mathrm{E}[\chi]$ of the χ-distribution with N degrees of freedom. If the actual length is larger than $\mathrm{E}[\chi]$ the whole argument in the e-function is positive and the mutation strength is increased. Conversely, it is decreased. The control rule (C15) is meanwhile well understood [14]. A slightly modified rule

$$\sigma^{(g+1)} = \sigma^{(g)} \exp\left[\frac{1}{2D}\left(\frac{\left\|\mathbf{s}^{(g+1)}\right\|^2}{N} - 1\right)\right], \quad (15)$$

that works without the $\mathrm{E}[\chi]$ value, proposed by Arnold [14], has been analyzed in [10].

Comparing the pseudocode in Fig. 1 with those presented in [1] or [5], we can conclude that the formulation of the CMA-ES can be significantly simplified: One only has to calculate the weighted recombinations of $\tilde{\mathbf{d}}$, $\tilde{\mathbf{z}}$, and $\tilde{\mathbf{d}}\tilde{\mathbf{d}}^{\mathrm{T}}$ according to

$$\left\langle \tilde{\mathbf{z}}^{(g)} \right\rangle_w := \sum_{m=1}^{\mu} w_m \tilde{\mathbf{z}}_{m;\lambda}^{(g)}, \quad (16)$$

$$\left\langle \tilde{\mathbf{d}}^{(g)} \right\rangle_w := \sum_{m=1}^{\mu} w_m \tilde{\mathbf{d}}_{m;\lambda}^{(g)}, \quad (17)$$

$$\left\langle \tilde{\mathbf{d}}^{(g)} \left(\tilde{\mathbf{d}}^{(g)}\right)^{\mathrm{T}} \right\rangle_w := \sum_{m=1}^{\mu} w_m \tilde{\mathbf{d}}_{m;\lambda}^{(g)} \left(\tilde{\mathbf{d}}_{m;\lambda}^{(g)}\right)^{\mathrm{T}}. \quad (18)$$

Furthermore, the calculation of the inverse to the square root of $\mathbf{C}$ as needed in the original CMA-ES, Eq. (8), is no longer needed. While the pseudocode in Fig. 1 is equivalent to the standard CMA-ES, it avoids unnecessarily complex derivations resulting in a clearer exposition of the basic ingredients that seem to be necessary to ensure the functioning of the CMA-ES. However, as will be shown in the next section, this pseudocode can even be transformed into another one using some simple and mild assumptions that allow for removing parts of the evolution path cumulation and the calculation of the $\mathbf{C}$-matrix at all.

III. Removing the $\mathbf{p}$ and the $\mathbf{C}$ from the CMA-ES

The CMA-ES pseudocode in Fig. 1 will be further transformed by first multiplying (C12) with $\mathbf{M}^{(g)}$ and taking (C6) into account

$$\begin{aligned}\mathbf{M}^{(g)}\mathbf{s}^{(g+1)} &= (1-c_s)\mathbf{M}^{(g)}\mathbf{s}^{(g)} + \sqrt{\mu_{\mathrm{eff}} c_s (2-c_s)}\left\langle \mathbf{M}^{(g)}\tilde{\mathbf{z}}^{(g)} \right\rangle_w \\ &= (1-c_s)\mathbf{M}^{(g)}\mathbf{s}^{(g)} + \sqrt{\mu_{\mathrm{eff}} c_s (2-c_s)}\left\langle \tilde{\mathbf{d}}^{(g)} \right\rangle_w . \quad (19)\end{aligned}$$

Now, compare with (C13). Provided that $c_p = c_s$, one immediately sees that

$$c_p = c_s \quad \Leftrightarrow \quad \mathbf{M}^{(g)}\mathbf{s}^{(g)} = \mathbf{p}^{(g)} \quad \Rightarrow \quad \mathbf{M}^{(g)}\mathbf{s}^{(g+1)} = \mathbf{p}^{(g+1)}. \quad (20)$$

Provided that $\mathbf{M}^{(g+1)} \simeq \mathbf{M}^{(g)}$ asymptotically holds for $N \to \infty$, (C13) can be dropped. Under that assumption (C13) is asymptotically a linear transformation of (C12). This is ensured by (24) as long as $c_s \to 0$ for $N \to \infty$. Yet, the question remains whether the assumptions $c_p = c_s$ and $\mathbf{M}^{(g+1)} \simeq \mathbf{M}^{(g)}$ are admissible in practice ($N < \infty$). In order to check how strongly c_p deviates from c_s the quotient c_p/c_s versus the search space dimensionality N is displayed in Fig. 2. The calculation is based on the standard parameter

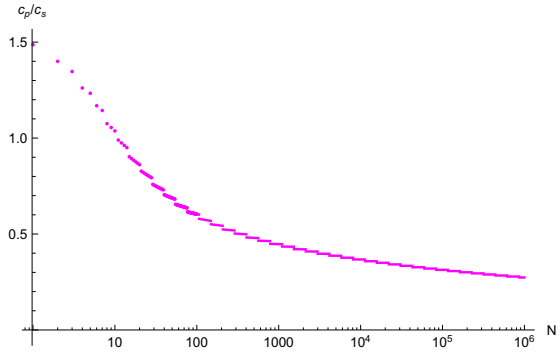

Fig. 2. The c_p/c_s ratio depending on the search space dimension N assuming the strategy parameter choice given by Eqs. (22–24), using the standard population sizing rule (21).

settings provided in [5][7]

$$\lambda = 4 + \lfloor 3\ln N \rfloor, \quad \mu = \left\lfloor \frac{\lambda}{2} \right\rfloor, \quad (21)$$

$$\mu_{\mathrm{eff}} = \frac{1}{\sum_{m=1}^{\mu} w_m^2}, \quad (22)$$

$$c_p = \frac{\mu_{\mathrm{eff}}/N + 4}{2\mu_{\mathrm{eff}}/N + N + 4}, \quad (23)$$

$$c_s = \frac{\mu_{\mathrm{eff}} + 2}{\mu_{\mathrm{eff}} + N + 5}. \quad (24)$$

As one can see, the c_p/c_s ratio is only a slightly decreasing function of N that does not deviate too much from 1. Therefore, one would not expect a much pronounced influence on the performance of the CMA-ES. Similarly, a replacement of

[7] In [5], c_p has been labeled as c_c and c_s as c_σ. We use the indices p and s to refer to the corresponding path vectors in (C12) and (C13).

$\mathbf{M}^{(g)}$ by $\mathbf{M}^{(g+1)}$ on the rhs of (20) should only result in small deviations. This will be confirmed in Section IV.

As a next step, line (C14) will be investigated. Applying Eq. (2) for g and $g+1$, respectively, and using (13) and (C6) one gets

$$\begin{aligned}
&\mathbf{M}^{(g+1)}\left(\mathbf{M}^{(g+1)}\right)^{\mathrm{T}} \\
&= (1-c_1-c_w)\mathbf{M}^{(g)}\left(\mathbf{M}^{(g)}\right)^{\mathrm{T}} \\
&\quad +c_1\mathbf{M}^{(g)}\mathbf{s}^{(g+1)}\left(\mathbf{M}^{(g)}\mathbf{s}^{(g+1)}\right)^{\mathrm{T}} \\
&\quad +c_w\sum_{m=1}^{\mu} w_m\mathbf{M}^{(g)}\tilde{\mathbf{z}}_{m;\lambda}^{(g)}\left(\mathbf{M}^{(g)}\tilde{\mathbf{z}}_{m;\lambda}^{(g)}\right)^{\mathrm{T}} \\
&= (1-c_1-c_w)\mathbf{M}^{(g)}\left(\mathbf{M}^{(g)}\right)^{\mathrm{T}} \\
&\quad +c_1\mathbf{M}^{(g)}\mathbf{s}^{(g+1)}\left(\mathbf{s}^{(g+1)}\right)^{\mathrm{T}}\left(\mathbf{M}^{(g)}\right)^{\mathrm{T}} \\
&\quad +c_w\sum_{m=1}^{\mu} w_m\mathbf{M}^{(g)}\tilde{\mathbf{z}}_{m;\lambda}^{(g)}\left(\tilde{\mathbf{z}}_{m;\lambda}^{(g)}\right)^{\mathrm{T}}\left(\mathbf{M}^{(g)}\right)^{\mathrm{T}} \\
&= (1-c_1-c_w)\mathbf{M}^{(g)}\left(\mathbf{M}^{(g)}\right)^{\mathrm{T}} \\
&\quad +c_1\mathbf{M}^{(g)}\mathbf{s}^{(g+1)}\left(\mathbf{s}^{(g+1)}\right)^{\mathrm{T}}\left(\mathbf{M}^{(g)}\right)^{\mathrm{T}} \\
&\quad +c_w\mathbf{M}^{(g)}\left\langle\tilde{\mathbf{z}}^{(g)}\left(\tilde{\mathbf{z}}^{(g)}\right)^{\mathrm{T}}\right\rangle_w\left(\mathbf{M}^{(g)}\right)^{\mathrm{T}} \\
&= \mathbf{M}^{(g)}\left(\mathbf{M}^{(g)}\right)^{\mathrm{T}} \\
&\quad +c_1\left(\mathbf{M}^{(g)}\mathbf{s}^{(g+1)}\left(\mathbf{s}^{(g+1)}\right)^{\mathrm{T}}\left(\mathbf{M}^{(g)}\right)^{\mathrm{T}}-\mathbf{M}^{(g)}\left(\mathbf{M}^{(g)}\right)^{\mathrm{T}}\right) \\
&\quad +c_w\left(\mathbf{M}^{(g)}\left\langle\tilde{\mathbf{z}}^{(g)}\left(\tilde{\mathbf{z}}^{(g)}\right)^{\mathrm{T}}\right\rangle_w\left(\mathbf{M}^{(g)}\right)^{\mathrm{T}}-\mathbf{M}^{(g)}\left(\mathbf{M}^{(g)}\right)^{\mathrm{T}}\right).
\end{aligned} \tag{25}$$

Pulling out the $\mathbf{M}^{(g)}$ on the rhs of (25) one finally obtains

$$\begin{aligned}
&\mathbf{M}^{(g+1)}\left(\mathbf{M}^{(g+1)}\right)^{\mathrm{T}} \\
&= \mathbf{M}^{(g)}\left[\mathbf{I}+c_1\left(\mathbf{s}^{(g+1)}\left(\mathbf{s}^{(g+1)}\right)^{\mathrm{T}}-\mathbf{I}\right)\right. \\
&\qquad \left.+c_w\left(\left\langle\tilde{\mathbf{z}}^{(g)}\left(\tilde{\mathbf{z}}^{(g)}\right)^{\mathrm{T}}\right\rangle_w-\mathbf{I}\right)\right]\left(\mathbf{M}^{(g)}\right)^{\mathrm{T}}.
\end{aligned} \tag{26}$$

This is a very remarkable result that paves the way for getting rid of the covariance matrix update. Instead of considering $\mathbf{C}$ one can directly evolve the $\mathbf{M}$ matrix if one finds an approach that connects $\mathbf{M}^{(g+1)}$ with $\mathbf{M}^{(g)}$. Technically, one has to perform some kind of "square root" operation on (26). Noting that Eq. (26) is of the form

$$\mathbf{A}\mathbf{A}^{\mathrm{T}} = \mathbf{M}\left[\mathbf{I}+\mathbf{B}\right]\mathbf{M}^{\mathrm{T}}, \tag{27}$$

where $\mathbf{B}$ is a symmetrical matrix, this suggests to expand $\mathbf{A}$ into a matrix power series

$$\mathbf{A} = \mathbf{M}\sum_{k=0}^{\infty}\gamma_k\mathbf{B}^k. \tag{28}$$

As can be easily shown by direct calculation,

$$\mathbf{A} = \mathbf{M}\left(\mathbf{I}+\frac{1}{2}\mathbf{B}-\frac{1}{8}\mathbf{B}^2+\ldots\right) \tag{29}$$

does satisfy (27) up to the power of two in $\mathbf{B}$. Provided that the matrix norm $\|\mathbf{B}\|$ is sufficiently small, one even can break off after the linear $\mathbf{B}$-term yielding

$$\begin{aligned}
\mathbf{M}^{(g+1)} &= \mathbf{M}^{(g)}\left[\mathbf{I}+\frac{c_1}{2}\left(\mathbf{s}^{(g+1)}\left(\mathbf{s}^{(g+1)}\right)^{\mathrm{T}}-\mathbf{I}\right)\right. \\
&\qquad \left.+\frac{c_w}{2}\left(\left\langle\tilde{\mathbf{z}}^{(g)}\left(\tilde{\mathbf{z}}^{(g)}\right)^{\mathrm{T}}\right\rangle_w-\mathbf{I}\right)+\ldots\right].
\end{aligned} \tag{30}$$

Using Eq. (30) as an approximation for $\mathbf{M}^{(g+1)}$ works especially well if c_1 and c_w are sufficiently small. Actually, using the suggested formulae in [5][8]

$$c_1 = \frac{\alpha_{cov}}{(N+1.3)^2+\mu_{\mathrm{eff}}} \tag{31}$$

and

$$c_w = \min\left(1-c_1,\ \alpha_{cov}\frac{\mu_{\mathrm{eff}}+1/\mu_{\mathrm{eff}}-2}{(N+2)^2+\alpha_{cov}\mu_{\mathrm{eff}}/2}\right), \tag{32}$$

it becomes clear that, given the population sizing (21), it holds

$$c_1 = \mathcal{O}\left(\frac{1}{N^2}\right) \quad \text{and} \quad c_w = \mathcal{O}\left(\frac{\ln N}{N^2}\right). \tag{33}$$

That is, at least for sufficiently large search space dimensionalities N, Eq. (30) should yield a similar behavior as the original $\mathbf{C}$-matrix update (C14).

Given the new update (30) we can outline the new simplified CMA-ES *without* covariance matrix and $\mathbf{p}$-path. The new ES may be called Matrix Adaptation ES (MA-ES) and is depicted in Fig. 3. The calculation of $\langle\tilde{\mathbf{z}}^{(g)}\left(\tilde{\mathbf{z}}^{(g)}\right)^{\mathrm{T}}\rangle_w$ in (M11) is done analogously to (3)

$$\left\langle\tilde{\mathbf{z}}^{(g)}\left(\tilde{\mathbf{z}}^{(g)}\right)^{\mathrm{T}}\right\rangle_w := \sum_{m=1}^{\mu} w_m\tilde{\mathbf{z}}_{m;\lambda}^{(g)}\left(\tilde{\mathbf{z}}_{m;\lambda}^{(g)}\right)^{\mathrm{T}}. \tag{34}$$

For the damping constant d_σ in (M12) and (C15), the recommended expression in [5]

$$d_\sigma = 1+c_s+2\max\left(0,\sqrt{\frac{\mu_{\mathrm{eff}}-1}{N+1}}-1\right) \tag{35}$$

is used in the empirical investigations of this paper.

Comparing the MA-ES algorithm in Fig. 3 with that of the original CMA-ES, Fig. 1, one sees that this pseudocode is not only shorter, but more importantly, it is less numerically demanding. Since there is no explicit covariance matrix update/evolution, there is also no matrix square root calculation. The numerical operations to be performed are only matrix-matrix and matrix-vector multiplications. It is to be mentioned here that there exists related work aiming at circumventing the covariance matrix operations. In [2] a $(1+1)$-CMA-ES was proposed that evolves the Cholesky-matrix. This approach has been extended to a multi-parent ES in [19]. Alternatively, considering the field of Natural Evolution Strategies (NES), the xNES [6] evolves a normalized transformation matrix. While the latter approach shares some similarities with the MA-ES update in line (M11), most notably the multiplicative

[8]According to [5], $0 < \alpha_{cov} \leq 2$. Throughout this paper it is assumed that $\alpha_{cov} = 2$.

$\underline{(\mu/\mu_w, \lambda)\text{-MA-ES}}$

Initialize$\left(\mathbf{y}^{(0)}, \sigma^{(0)}, g := 0, \mathbf{s}^{(0)} := \mathbf{0}, \mathbf{M}^{(0)} := \mathbf{I}\right)$ (M1)

Repeat (M2)

For $l := 1$ **To** λ (M3)

$\tilde{\mathbf{z}}_l^{(g)} := \boldsymbol{\mathcal{N}}_l(\mathbf{0}, \mathbf{I})$ (M4)

$\tilde{\mathbf{d}}_l^{(g)} := \mathbf{M}^{(g)}\tilde{\mathbf{z}}_l^{(g)}$ (M5)

$\tilde{f}_l^{(g)} := f\left(\mathbf{y}^{(g)} + \sigma^{(g)}\tilde{\mathbf{d}}_l^{(g)}\right)$ (M6)

End (M7)

SortOffspringPopulation (M8)

$$\mathbf{y}^{(g+1)} := \mathbf{y}^{(g)} + \sigma^{(g)}\left\langle \tilde{\mathbf{d}}^{(g)} \right\rangle_w \quad \text{(M9)}$$

$$\mathbf{s}^{(g+1)} := (1-c_s)\mathbf{s}^{(g)} + \sqrt{\mu_{\text{eff}} c_s (2-c_s)}\left\langle \tilde{\mathbf{z}}^{(g)} \right\rangle_w \quad \text{(M10)}$$

$$\mathbf{M}^{(g+1)} := \mathbf{M}^{(g)}\Big[\mathbf{I} + \frac{c_1}{2}\left(\mathbf{s}^{(g+1)}\left(\mathbf{s}^{(g+1)}\right)^{\mathrm{T}} - \mathbf{I}\right) + \frac{c_w}{2}\left(\left\langle \tilde{\mathbf{z}}^{(g)}\left(\tilde{\mathbf{z}}^{(g)}\right)^{\mathrm{T}} \right\rangle_w - \mathbf{I}\right)\Big] \quad \text{(M11)}$$

$$\sigma^{(g+1)} := \sigma^{(g)} \exp\left[\frac{c_s}{d_\sigma}\left(\frac{\left\|\mathbf{s}^{(g+1)}\right\|}{\mathrm{E}\left[\left\|\boldsymbol{\mathcal{N}}(\mathbf{0}, \mathbf{I})\right\|\right]} - 1\right)\right] \quad \text{(M12)}$$

$$g := g + 1 \quad \text{(M13)}$$

Until(termination condition(s) fulfilled) (M14)

Fig. 3. Pseudocode of the Matrix Adaptation ES (MA-ES).

change of the transformation matrix, xNES does not incorporate and evolve an evolution path (M10). Furthermore, a similar multiplicative $\mathbf{C}$-update for the CMA-ES has been proposed in [20].

The MA-ES allows for a *novel* interpretation of the $\mathbf{M}$ matrix learning process by considering line (M11). The driving force for the change of the $\mathbf{M}$ matrix is given by the deviations of the matrices $\mathbf{s}^{(g+1)}\left(\mathbf{s}^{(g+1)}\right)^{\mathrm{T}}$ and $\left\langle \tilde{\mathbf{z}}^{(g)}\left(\tilde{\mathbf{z}}^{(g)}\right)^{\mathrm{T}} \right\rangle_w$ from the identity matrix $\mathbf{I}$. If the resulting matrix is similar to the identity matrix, then $\mathbf{M}$ is simply scaled by a scalar factor. Furthermore, if the *selected* $\tilde{\mathbf{z}}$ vectors are standard normally distributed, the *expected value* of $\mathbf{M}$ does not change since in that case it can be easily shown that

$$\mathrm{E}\left[\mathbf{s}^{(g+1)}\left(\mathbf{s}^{(g+1)}\right)^{\mathrm{T}}\right] = \mathbf{I} \quad \text{and} \quad \mathrm{E}\left[\left\langle \tilde{\mathbf{z}}^{(g)}\left(\tilde{\mathbf{z}}^{(g)}\right)^{\mathrm{T}} \right\rangle_w\right] = \mathbf{I} \quad (36)$$

does hold. That is, provided that the $\mathbf{M}$ matrix has evolved in such a manner that the components of the selected $\tilde{\mathbf{z}}$ vectors are (nearly) standard normally distributed, the evolution of $\mathbf{M}$ has reached a steady state. Of course, such a stationary behavior can only exist for static quadratic fitness landscapes (i.e., f-functions with a constant Hessian) and more general functions obtained from those quadratic functions by strictly increasing f-transformations (conserving ellipsoidal level sets).

IV. Performance Comparison MA- vs. CMA-ES

The performance of the MA-ES has been extensively tested and compared with the CMA-ES. These investigations are intended to show that the MA-ES exhibits similar performance behaviors as the CMA-ES. To this end, not only standard population sizings ($\lambda < N$) have been considered, but also cases where $\lambda = \mathcal{O}(N^2)$ since large population sizes are needed in global optimization cases as they are commonly encountered when using the CMA-ES as search engine in Bi-Population (BiPop) settings. The comparisons presented consider the evolution dynamics as well as the aggregated statistics in terms of the expected runtime (ERT). ERT is measured as the average number of function evaluations needed to reach a predefined objective function target f_{target}. In the following some of the experiments performed are presented. The complete collection of experimental results are to found in the supplementary material.

TABLE I
Test functions and stopping criterion for the empirical performance evaluation. The initial conditions are $\mathbf{y}^{(0)} = (1, \ldots, 1)^{\mathrm{T}}$ and $\sigma^{(0)} = 1$.

Name	Function	f_{target}
Sphere	$f_{\text{Sp}}(\mathbf{y}) := \sum_{i=1}^{N} y_i^2 \quad (= \|\mathbf{y}\|^2)$	10^{-10}
Cigar	$f_{\text{Cig}}(\mathbf{y}) := y_1^2 + 10^6 \sum_{i=2}^{N} y_i^2$	10^{-10}
Tablet	$f_{\text{Tab}}(\mathbf{y}) := 10^6 y_1^2 + \sum_{i=2}^{N} y_i^2$	10^{-10}
Ellipsoid	$f_{\text{Ell}}(\mathbf{y}) := \sum_{i=1}^{N} 10^{6\frac{i-1}{N-1}} y_i^2$	10^{-10}
Parabolic Ridge	$f_{\text{pR}}(\mathbf{y}) := -y_1 + 100 \sum_{i=2}^{N} y_i^2$	-10^{10}
Sharp Ridge	$f_{\text{sR}}(\mathbf{y}) := -y_1 + 100\sqrt{\sum_{i=2}^{N} y_i^2}$	-10^{10}
Different-Powers	$f_{\text{dP}}(\mathbf{y}) := \sum_{i=1}^{N} (y_i^2)^{1+5\frac{i-1}{N-1}}$	10^{-10}
Rosenbrock	$f_{\text{Ros}}(\mathbf{y}) := \sum_{i=1}^{N-1} \left[100(y_i^2 - y_{i+1})^2 + (y_i - 1)^2\right]$	10^{-10}

A. Using Standard Population Sizing According to Eq. (21)

According to the derivation presented, the MA-ES should perform nearly equally well as the CMA-ES. We will investigate the performance on the set of test functions given in Tab. I. All runs have been initialized with a mutation strength $\sigma^{(0)} = 1$ and $\mathbf{y}^{(0)} = (1, \ldots, 1)^{\mathrm{T}}$. For the first tests, the standard strategy parameter choice given by Eqs. (21-24) together with the recombination weights (5) have been used.

In addition to the test functions in Tab. I, the 24 test functions from the BBOB test environment [12] have been used to evaluated the MA-ES in the BiPop-setting in Section V-B.

1) Mean value dynamics: In Fig. 4 the dynamics of the CMA-ES and the MA-ES are compared on the test functions Sphere, Cigar, Tablet, Ellipsoid, ParabolicRidge, SharpRidge, Rosenbrock, and DifferentPowers for search space dimensionalities $N = 3$ and $N = 30$. The curves displayed are the result

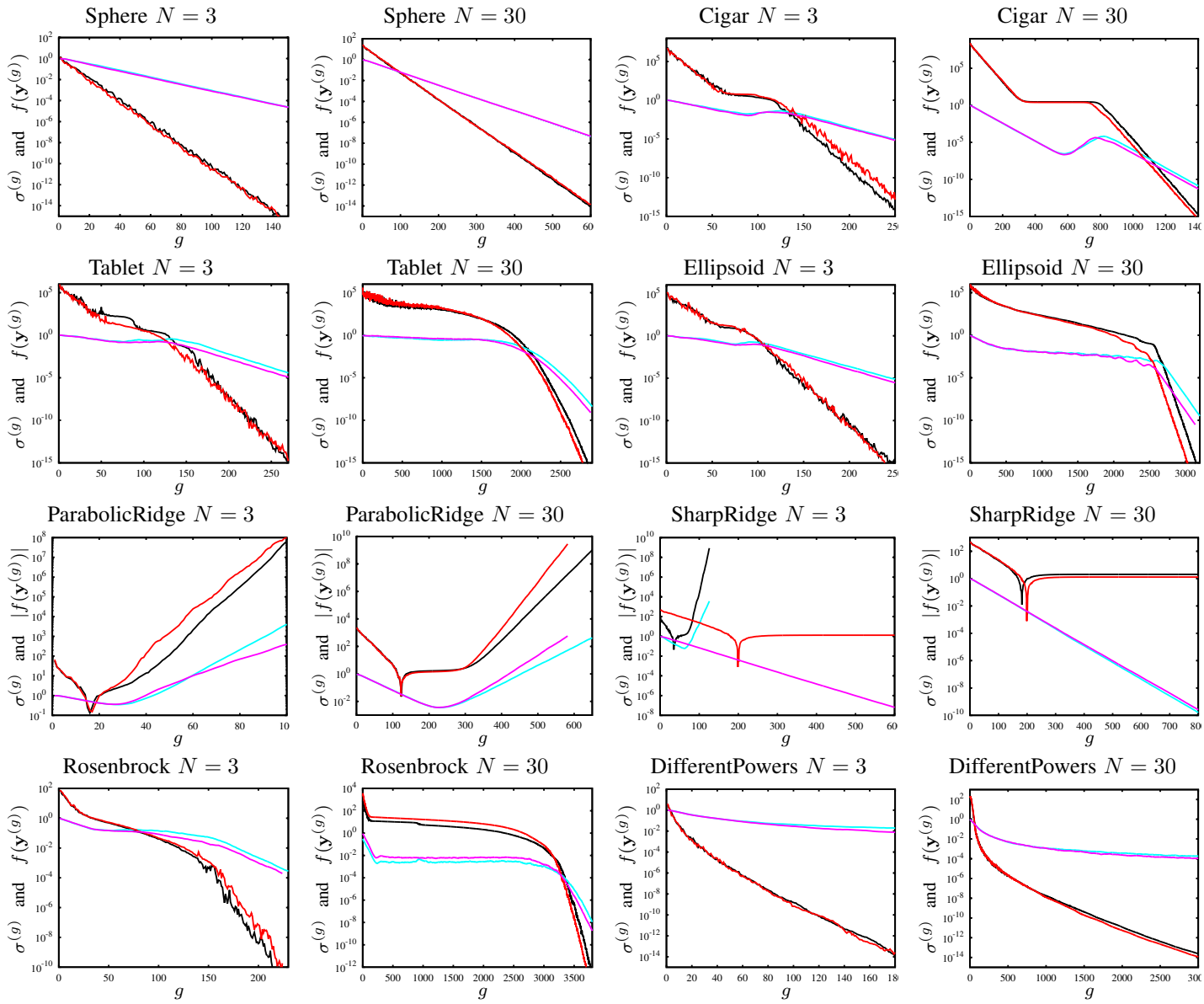

Fig. 4. The σ- and f- or $|f|$-dynamics of CMA-ES and MA-ES on the test functions of Tab. I for $N = 3$ and $N = 30$. The f-dynamics of the CMA-ES are in black and those of the MA-ES are in red. The σ-dynamics of the CMA-ES are in cyan and those of the MA-ES are in magenta. The curves are the averages of 20 independent ES runs (except the SharpRidge $N = 3$ case where 200 runs have been averaged and the Rosenbrock $N = 3$ case where 100 runs have been averaged). Note that the MA-ES exhibits premature convergence on the SharpRidge while the CMA-ES shows this behavior for $N = 30$.

of an averaging over a number of independent single runs. Thus, they are an estimate of the mean value dynamics.

As one can see in Fig. 5, using the standard strategy parameter setting both CMA-ES and MA-ES perform very similarly on the $N = 30$ test instances. Even in the case $N = 3$ this behavior is observed except the SharpRidge where the MA-ES exhibits premature convergence while the CMA-ES is able to follow the ridge exponentially fast. However, increasing N slightly, also the CMA-ES will fail on the SharpRidge. If one wants to improve the behavior on the SharpRidge (while keeping the basic CMA-ES algorithm), the population sizing rule (21) must be changed and much larger λ values must be used. As a general observation it is to be noted that the $N = 3$ case produces rather rugged single run dynamics. Regarding the Rosenbrock function, only those runs have been used for averaging that converged to the global minimum (there is a second, but local minimum that sometimes attracts the ES).

2) Expected runtime: In order to compare the N scaling behavior, expected runtime (ERT) experiments have been performed. Since a converging ES does not always reach a predefined objective function target value f_{target}, but may end up in a local optimizer, a reasonable ERT definition must take into account unsuccessful runs. Therefore, the formula

$$\text{ERT} = \left(\frac{1}{P_\text{s}} - 1\right) \text{E}\left[r_\text{u}\right] + \text{E}\left[r_\text{s}\right] \tag{37}$$

derived in [15] has been used for the calculation of the ERT. For $\text{E}\left[r_\text{u}\right]$ the empirical estimate of the runtime for unsuccessful runs has been determined. Likewise, the mean value of the runtime of the successful runs serves as an empirical estimate for $\text{E}\left[r_\text{s}\right]$ and P_s is to be estimated as

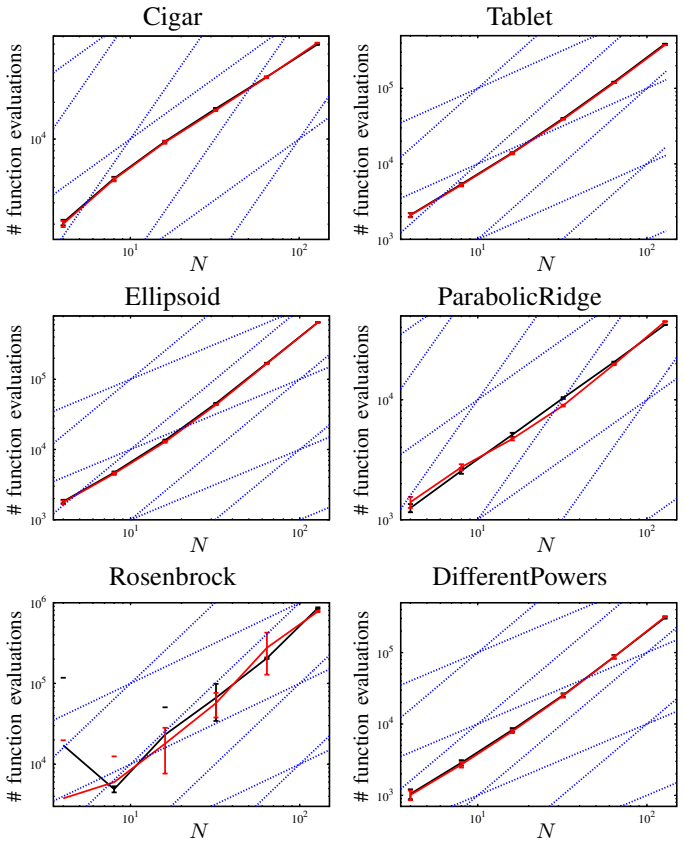

Fig. 5. Expected runtime (in terms of # of function evaluations) and its standard deviation of the CMA-ES, displayed by black data points (and error bars) and of the MA-ES, displayed by red data points (and error bars). The population size λ is given by (21). Data points have been obtained for $N = 4, 8, 16, 32, 64$, and 128. As for the initial conditions and f_{target}, see Tab. I. For comparison purposes, dashed blue lines are displayed in order to represent linear (smaller slope) and quadratic (larger slope) runtime growth behavior. Note, regarding Rosenbrock especially for small N, the standard deviation of the runtime is rather large and only displayed as a horizontal bar above the respective data point.

empirical success probability. Similarly, one can determine the variance of the run length, the formula reads[9]

$$\mathrm{Var}[r] = \left(\tfrac{1}{P_\mathrm{s}} - 1\right)\mathrm{Var}[r_\mathrm{u}] + \mathrm{Var}[r_\mathrm{s}] + \left(\tfrac{1-P_\mathrm{s}}{P_\mathrm{s}^2}\right)\mathrm{E}[r_\mathrm{u}]^2 \,. \quad (38)$$

The ERT experiments performed regard search space dimensionalities $N = 4, 8, 16, 32, 64$, and 128. These investigations exclude the SharpRidge since both CMA-ES and MA-ES exhibit premature convergence for moderately large N (for CMA-ES: $N > 16$ experiments exhibit premature convergence). The results of these experiments are displayed in Fig. 5. Instead of the expected number of generations, the number of function evaluations are displayed versus the search space dimensionality N. The number of function evaluations is simply the product of the expected runtime measured in generations needed (to reach a predefined f target) times the offspring population size λ.

As one can see, both strategies exhibit nearly the same expected runtime performance. Only for the Rosenbrock function there are certain differences. These are mainly due to the effect of occasional convergence of the ES to the local optimizer. This is also the reason for the comparatively large standard deviations $\sqrt{\mathrm{Var}[r]}$ observed.

B. Considering $\lambda = 4N$ Population Sizing

It is well known that the standard population sizing according to (21) is not well suited for multimodal and noisy optimization. First, the case $\lambda = 4N$, $\mu = \lambda/2$ will be considered. In the next section, the $\lambda = 4N^2$ case will be investigated. Since relevant differences in the dynamics (compared to the dynamics of the standard population sizing rule (21) in Fig. 4) can only be observed for the Ridge (and to a certain extend for the Cigar), the graphs are presented in the supplementary material. Regarding the ParabolicRidge and the Cigar one observes a certain performance loss for the MA-ES. Having a closer look at the ERT graphs in Fig. 6 one sees that these differences appear more pronounced at larger search space dimensionalities on peculiar ellipsoid models like the Cigar and the "degenerated Cigar" – the Parabolic Ridge. In those test cases the CMA-ES seems to benefit from the second evolution path: There is a dominating major principal axis in these test functions (being constant in its direction in the search space) in which the mutations should be predominantly directed. It seems that such a direction can be somewhat faster identified using a second evolution path.

C. Considering $\lambda = 4N^2$ Population Sizing

The mean value dynamics of the CMA-ES and the MA-ES with population sizes proportional to the square of the search space dimensionality N are mainly presented in the supplementary material. The f-dynamics of CMA-ES and MA-ES do not deviate markedly and are similar to those of the other population sizings. Some of the experimental results regarding Sphere, Ellipsoid, and Rosenbrock are also displayed in Fig. 7. The curves are the result of an averaging over a number of independent single runs. As one can see, there is a certain performance loss for the MA-ES except for Rosenbrock at $N = 30$ where MA excels.

The mutation strength dynamics of the two ES algorithms on Rosenbrock are interesting on its own. In the case of $N = 30$, σ increases even exponentially fast. Since both the CMA-ES and the MA-ES converge to the optimizer, shrinking the mutation step-size (note, this is *not* σ as often wrongly stated) is predominantly done by the covariance matrix or the $\mathbf{M}$ matrix (in the case of the MA-ES). This can be easily confirmed by considering the dynamics of the maximal and minimal eigenvalues of $\mathbf{C}$ (in the case of the CMA-ES) and $\mathbf{M}\mathbf{M}^\mathrm{T}$ (in the case of the MA-ES), respectively. Therefore, it is a general observation for the large population size $\lambda = 4N^2$ case that adaptation of the step-size is mainly done by the update of $\mathbf{M}$ and $\mathbf{C}$, respectively, and not by σ. This even holds for the Sphere model as shown in Figure 8.

Next, let us consider the expected runtime (ERT) curves in the interval $N = 4$ to $N = 128$ in Figs. 9.[10] Having a

[9]Note, in [15] a wrong formula has been presented. This is the correct one.

[10]Unlike the λ scaling law (21), choosing $\lambda = 4N^2$ prevents the CMA-ES and the MA-ES from premature convergence on the Sharp Ridge. That is why, the ERT curves are displayed for this test function as well.

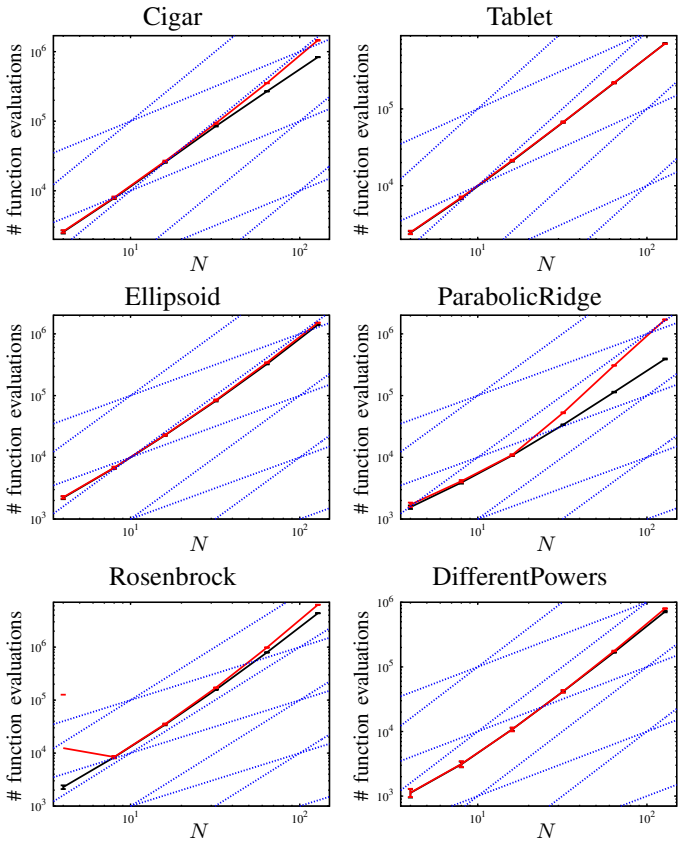

Fig. 6. Expected runtime (in terms of # of function evaluations) and its standard deviation of the CMA-ES, displayed by black data points (and error bars) and of the MA-ES, displayed by red data points (and error bars). The $\lambda = 4N$, $\mu = 2N$ case has been investigated. Data points have been obtained for $N = 4, 8, 16, 32, 64$, and 128. As for the initial conditions and f_{target}, see Tab. I. For comparison purposes, dashed blue lines are displayed in order to represent linear (smaller slope) and quadratic (larger slope) runtime growth behavior. Note, regarding Rosenbrock for $N = 4$ the standard deviation of the runtime is rather large and only displayed as a horizontal bar above the respective data point.

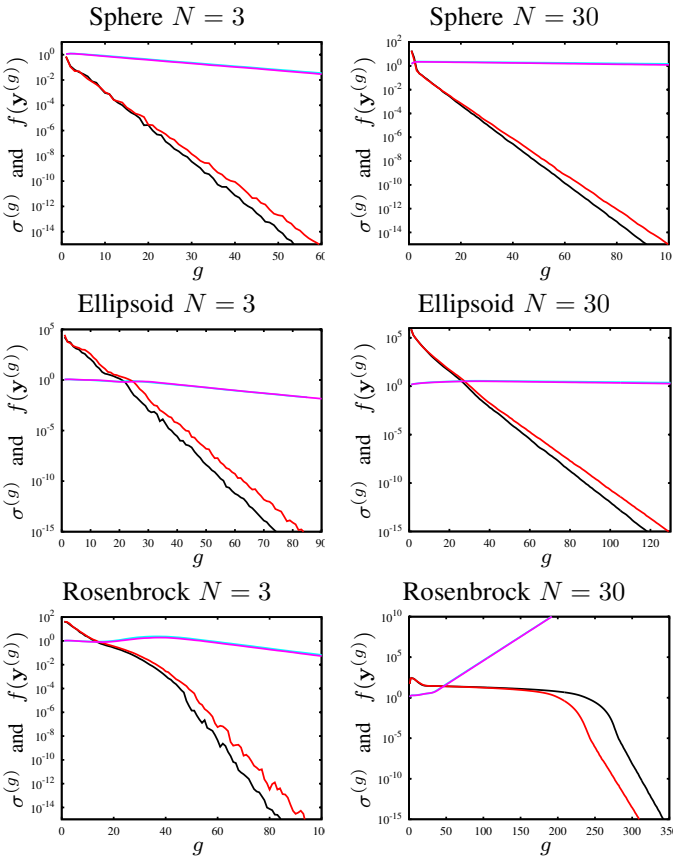

Fig. 7. σ- and f-dynamics of CMA-ES and MA-ES with $\lambda = 4N^2$ on Sphere, Ellipsoid, and Rosenbrock for $N = 3$ (left) and $N = 30$ (right). The f-dynamics of the CMA-ES are in black and those of the MA-ES are in red. The σ-dynamics of the CMA-ES are in cyan and those of the MA-ES are in magenta. The curves are the averages of 20 independent ES runs (except the Rosenbrock N=3 case where 100 runs have been averaged).

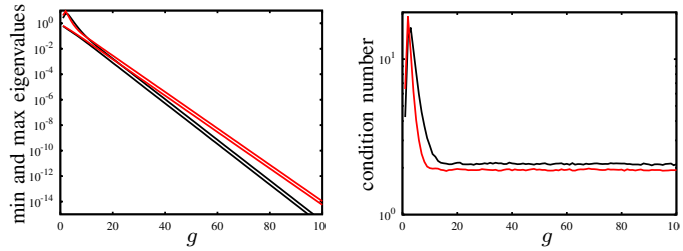

Fig. 8. Left figure: On the evolution of the minimal and the maximal eigenvalues of $\mathbf{C}$ (black curves) and $\mathbf{M}\mathbf{M}^{\mathrm{T}}$ (red curves) for a $(1800/1800_I, 3600)$-CMA-ES and $(1800/1800_I, 3600)$-MA-ES, respectively, on the $N = 30$-dimensional Sphere model. Right figure: Corresponding condition number dynamics.

closer look at the ERT graphs, one sees that - apart from the Parabolic and the Sharp Ridge - both the CMA-ES and the MA-ES exhibit a larger slope than the dotted straight lines $\propto N^2$. That is, ERT as a function of N exhibits a super-quadratic runtime. As for Rosenbrock (see Fig. 9) this might be partially attributed to the change of the local topology that requires a permanent change of the covariance matrix during the evolution. Regarding the ellipsoid-like models such as Sphere, Cigar, Tablet, and Ellipsoid this might indicate that CMA- and MA-ES do not work optimally with large population sizes. This suspicion is also formed by the non-decreasing behavior of the mutation strength σ in Fig. 7 for $N = 30$ indicating a possible failure of the σ-control rule. This raises the research question as to the reasons for this observed behavior and whether one can improve the scaling behavior.

V. Performance of the MA-ES as Building Block in a BiPop-ES

A. The BiPop-Algorithm

In order to solve multi-modal optimization problems, the CMA-ES should be used in a restart setting with increasing population sizes and/or different initial mutation strengths [16]. An implementation of such a multi-start approach was proposed in terms of the BiPop-CMA-ES [11]. Its pseudocode is presented in Fig. 10.[11] It is the aim of this section to show that replacing the basic version of the CMA-ES given in Fig. 1 by the MA-ES, Fig. 3, does not significantly deteriorate the performance of the BiPop-CMA-ES. The BiPop-CMA-

[11] Since a pseudocode of the BiPop-CMA-ES [11] has not been published, it has been derived here from the verbal description given in [11], [17] and private communications with Ilya Loshchilov.

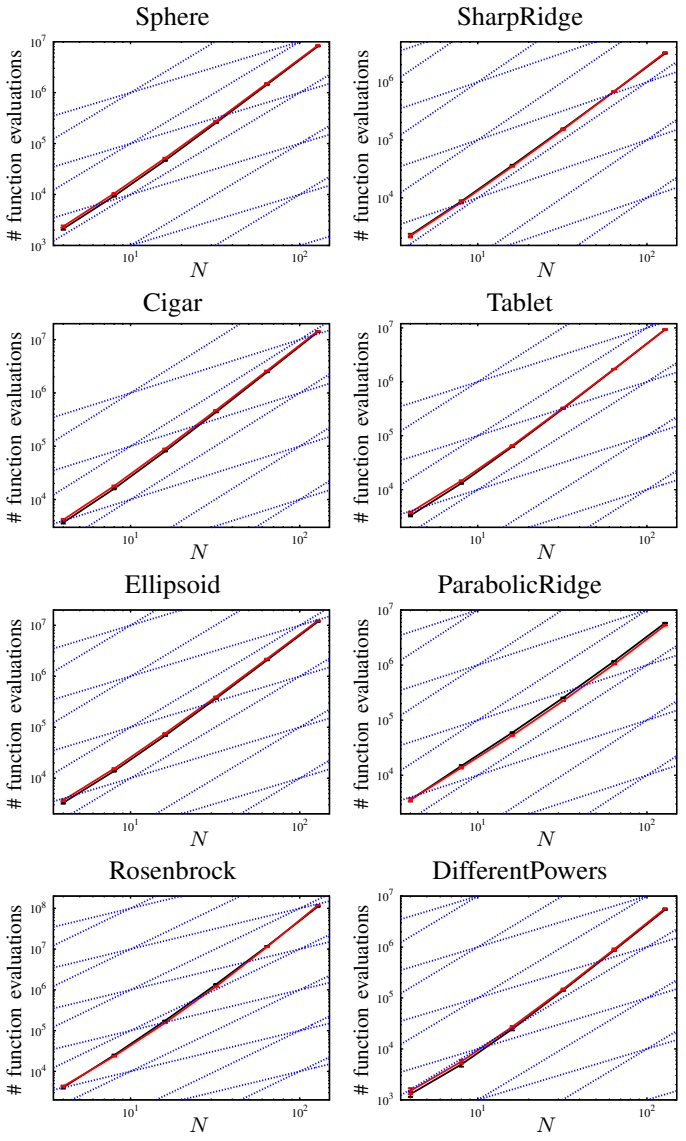

Fig. 9. Expected runtime and its standard deviation of the CMA-ES, displayed by black data points (and error bars) and of the MA-ES, displayed by red data points (and error bars). The $\lambda = 4N^2$, $\mu = 2N^2$ case has been investigated. Data points have been obtained for $N = 4, 8, 16, 32, 64$, and 128. As for the initial conditions and f_{target}, see Tab. I. For comparison purposes, dashed blue lines are displayed in order to represent linear (smaller slope) and quadratic (larger slope) runtime growth behavior.

BiPop-(C)MA-ES for minimization

Initialize$\big(\mathbf{y}_{\text{lower}}, \mathbf{y}_{\text{upper}}, \sigma_{\text{init}}, \lambda_{\text{init}},$
$f_{\text{stop}}, b_{\text{stop}}, g_{\text{stop}} := \infty\big)$ (B1)
$\mathbf{y}_{\text{init}} := \mathbf{u}[\mathbf{y}_{\text{lower}}, \mathbf{y}_{\text{upper}}]$ (B2)
$(\mathbf{y}_{\min}, f_{\min}, b_1) :=$
(C)MA-ES$(\mathbf{y}_{\text{init}}, \sigma_{\text{init}}, \lambda_{\text{init}}, g_{\text{stop}}, TC)$ (B3)
If $f_{\min} \le f_{\text{stop}}$ **Then Return**$(\mathbf{y}_{\min}, f_{\min})$ (B4)
$n := 0;\ n_{\text{small}} := 0;$ (B5)
$b_{\text{large}} := 0;\ b_{\text{small}} := 0;$ (B6)
Repeat (B7)
$n := n + 1$ (B8)
$\lambda := 2^{n - n_{\text{small}}} \lambda_{\text{init}}$ (B9)
$\mathbf{y}_{\text{init}} := \mathbf{u}[\mathbf{y}_{\text{lower}}, \mathbf{y}_{\text{upper}}]$ (B10)
If $n > 2$ **AND** $b_{\text{small}} < b_{\text{large}}$ **Then** (B11)
$\sigma_{\text{sInit}} := \sigma_{\text{init}}/100^{\mathbf{u}[0,1]}$ (B12)
$\lambda_{\text{small}} := \left\lfloor \lambda_{\text{init}} \left(\frac{1}{2}\lambda/\lambda_{\text{init}}\right)^{(\mathbf{u}[0,1])^2} \right\rfloor$ (B13)
$g_{\text{sStop}} := \left\lfloor \frac{1}{2} b_{\text{large}}/\lambda_{\text{small}} \right\rfloor$ (B14)
$(\check{\mathbf{y}}, \check{f}, b_{\text{S}}) :=$
(C)MA-ES$(\mathbf{y}_{\text{init}}, \sigma_{\text{sInit}}, \lambda_{\text{small}}, g_{\text{sStop}}, TC)$ (B15)
$b_{\text{small}} := b_{\text{small}} + b_{\text{S}}$ (B16)
$n_{\text{small}} := n_{\text{small}} + 1$ (B17)
Else (B18)
$(\check{\mathbf{y}}, \check{f}, b_{\text{L}}) :=$ (C)MA-ES$(\mathbf{y}_{\text{init}}, \sigma_{\text{init}}, \lambda, g_{\text{stop}}, TC)$ (B19)
$b_{\text{large}} := b_{\text{large}} + b_{\text{L}}$ (B20)
EndIf (B21)
If $\check{f} \le f_{\min}$ **Then** (B22)
$\mathbf{y}_{\min} := \check{\mathbf{y}};\ f_{\min} := \check{f}$ (B23)
EndIf (B24)
Until$(b_{\text{small}} + b_{\text{large}} + b_1 \ge b_{\text{stop}}$ OR $f_{\min} \le f_{\text{stop}})$ (B25)
Return$(\mathbf{y}_{\min}, f_{\min})$ (B26)

Fig. 10. Pseudocode of the BiPop-CMA-ES and the BiPop-MA-ES.

ES uses the (C)MA-ESs[12] as "search engine" in lines B3, B15, and B19, respectively. In order to utilize the (C)MA-ESs, it is important that these strategies do not waste too many function evaluations b (refers to "budget") in the case of local convergence. That is, the overall performance is also influenced by the termination conditions (abbreviated as TC) used in the (C)MA-ESs. In the original BiPop-CMA-ES paper [11] various TCs have been used in order to get good benchmark results. Since we want to keep the (C)MA-ESs conceptually generic, we use only four termination conditions in the empirical evaluations:

1) maximum number of generations $g \ge g_{\text{stop}}$
2) parental f-value smaller than predefined f_{stop}, i.e., $f^{(g)} < f_{\text{stop}}$ (f-minimization considered)
3) distance change of parent $\mathbf{y}^{(g)}$ in search space smaller than a predefined step length Δ, i.e., $\|\mathbf{y}^{(g+1)} - \mathbf{y}^{(g)}\| < \Delta_{\text{stop}}$, this can be checked by calculating the Euclidean norm of $\sigma^{(g)} \left\langle \tilde{\mathbf{d}}^{(g)} \right\rangle_w$ in (C11), Fig. 1, and (M9), Fig. 3,
4) stagnation of the parental f-values, termination if for $g > G : f(\mathbf{y}^{(g)}) \approx f(\mathbf{y}^{(g-G)})$[13]

The BiPop-ES has two termination conditions in (B25). One regards the best f value $f_{\min}$, i.e., if it drops below f_{stop} the strategy stops. The other criterion terminates the BiPop-

[12](C)MA-ES is used as abbreviation to refer to both the classical CMA-ES and the MA-ES.

[13]In the benchmark experiments, $g > 10$ and $G = 10$ were used. The implementation of "$\approx$" regards real numbers as approximately equal if these numbers differ only in the two least significant digits.

ES if the total number of function evaluations is greater or equal to the predefined stop budget $b_{\rm stop}$. The total function evaluation budget $b_{\rm stop}$ is the sum of function evaluations consumed by the first (C)MA-ES run in (B3), being b_1, and the cumulated function evaluations in (B15), being $b_{\rm small}$, and in (B19), being $b_{\rm large}$. Depending on the number of function evaluations already consumed by (B15) and (B19), the Then-branch (B12–B17) or the Else-branch (B19, B20) is performed. Within the Then-branch (B12)–(B17) the BiPop-ES runs an ES with a randomly derived small population size $\lambda_{\rm small}$ (compared to λ) and also a randomly decreased initial $\sigma_{\rm sInit}$. Here the probabilistic choices are introduced by random numbers u being uniformly distributed in the interval $[0, 1]$. As a result of the transformation in (B12) the probability density of $\sigma_{\rm sInit}$ is proportional to $\frac{1}{\sigma_{\rm sInit}}$. Similarly, one can show that $\lambda_{\rm small}$ obeys a probability density proportional to $\frac{1}{\lambda_{\rm small}\sqrt{\ln \lambda_{\rm small}}}$. That is, the random choices of $\sigma_{\rm sInit}$ and $\lambda_{\rm small}$ are drawn with a strong tendency towards smaller values. That is, together with (B10) (see below), this initiates basically local searches. The Else-branch (B19, B20) is performed with (exponentially) increasing population sizes λ the size of which is determined in (B9), thus performing a rather global search.[14] The criterion $b_{\rm small} < b_{\rm large}$ in (B11) ensures that the total function evaluation budget is approximately evenly devoted to both ESs with small population sizes and the ES with large population size. As a result, the Then-branch (small population sizes) is more often performed than the Else-branch (with large population size). Each (C)MA-ES run is started with an uniformly random initialized $\mathbf{y}$ in the box interval predefined by the vectors $\mathbf{y}_{\rm lower}$ and $\mathbf{y}_{\rm upper}$.

B. Performance Evaluation Using the COCO-Framework

The performance evaluation is realized by the COmparing Continuous Optimizers (COCO) Software Version v15.03 [12][15] using the test functions given in Tab. II. We especially present the empirical cumulative performance graphs that compare the overall performance of the CMA-ES and the MA-ES as search engine in the BiPop-ES framework. Additionally, we compare with Hansen's CMA-ES production code version 3.61.beta or 3.62.beta, respectively (last change: April, 2012).[16] This production code contains ad hoc solutions to improve the performance of the generic CMA-ES. It is expected that this should be visible by better empirical benchmark results compared to the CMA and MA versions. However, extensive tests have shown that the performance advantages in terms of better expected runtimes – while being statistically significant on some test functions – are rather small (see supplementary material).

[14]A deeper, scientifically satisfactory explanation of the details in (B9), (B12)–(B14) can not be given here. Even the original publication [11] does not provide deeper insights and it seems that this is rather an ad hoc choice that meets the needs without a derivation from first principles.

[15]Software and documentation have been obtained from: http://coco.gforge.inria.fr.

[16]Source code downloaded from: https://www.lri.fr/~hansen/count-cmaes-m.php?Down=cmaes.m. There is an ambiguity w.r.t. the version number in the code where one can find both version numbers 3.61.beta and 3.62.beta in the same code.

TABLE II
TEST FUNCTIONS OF THE BBOB TEST BED, DESCRIPTION TAKEN FROM [18]

Properties	#	**Name, Description**
Separable	f_1	Sphere
	f_2	Ellipsoid with monotone $\mathbf{y}$-transformation, condition 10^6
	f_3	Rastrigin with both $\mathbf{y}$-transformation, condition 10
	f_4	Skew Rastrigin-Bueche (condition 10), skew-condition 10^2
	f_5	Linear slope, neutral extension outside the domain (not flat)
Low or moderate condition	f_6	Attractive sector function
	f_7	Step-ellipsoid, condition 10^2
	f_8	Rosenbrock, original
	f_9	Rosenbrock, rotated
High condition	f_{10}	Ellipsoid with monotone $\mathbf{y}$-transformation, condition 10^6
	f_{11}	Discus with monotone $\mathbf{y}$-transformation, condition 10^6
	f_{12}	Bent cigar with asymmetric $\mathbf{y}$-transformation, condition 10^6
	f_{13}	Sharp ridge, slope 1:100, condition 10
	f_{14}	Sum of different powers
Multi-modal	f_{15}	Rastrigin with both $\mathbf{y}$-transformations, condition 10
	f_{16}	Weierstrass with monotone $\mathbf{y}$-transformation, condition 10^2
	f_{17}	Schaffer F7 with asymmetric $\mathbf{y}$-transformation, condition 10
	f_{18}	Schaffer F7 with asymmetric $\mathbf{y}$-transformation, condition 10^3
	f_{19}	$f_8 f_2$ composition of 2-D Griewank-Rosenbrock
Multi-modal with weak global structure	f_{20}	Schwefel $x * \sin(x)$ with tridiagonal transformation, condition 10
	f_{21}	Gallagher 101 Gaussian peaks, condition up to 10^3
	f_{22}	Gallagher 21 Gaussian peaks, condition up to 10^3
	f_{23}	Katsuuras repetitive rugged function
	f_{24}	Lunacek bi-Rastrigin, condition 10^2

The benchmark experiments were performed with a stop budget of $b_{\rm stop} = 1000N^2$ function evaluations. The actual number of function evaluations until leaving the Repeat-Until-loop in Fig. 10 will exceed this value by a certain extent (because of simplicity reasons, no additional budget-driven stopping criterion has been incorporated into the set of termination conditions TC within the (C)MA-ESs). The initial offspring population size is determined by (21). The initial mutation strength has been chosen as $\sigma_{\rm init} = 10/\sqrt{N}$, the minimum parental step length was $\Delta_{\rm stop} = 10^{-7}$. The initial parental $\mathbf{y}$ has been chosen uniformly at random in a box interval $\mathbf{y}_{\rm init} \in [-5, 5]^N$. The lag for stagnation detection in f-space was $G = 10$.

The benchmark experiments use the standard settings and testbed described in [12]. The goals of these experiments are

a) to show that exchanging the classical CMA-ES search engine, Fig. 1, by the MA-ES, Fig. 3, does not severely affect the overall performance of the BiPop-ES,
b) to compare the performance with the BiPop-CMA-ES production code version 3.61 to get a feeling how much can be gained by a sophisticatedly tuned CMA-ES.

In the standard BBOB setting, the performance of each al-

gorithm is evaluated on 15 independent (randomly transformed and perturbed) instantiations of each test function. Based on the observed run lengths, ECDF (empirical cumulative distribution function) graphs are generated. These graphs show the percentages of f target values f_{target} reached for a given function value budget b per search space dimensionality N (in the plots, D is used to indicate the dimensionality). The standard BBOB f-target values used are $f_{\text{target}} = \min[f] + 10^k$, $k \in \{-8, \ldots, 2\}$.

Aggregated performance profiles over the whole test function set f_1 – f_{24} (Tab. II) are presented in Fig. 11 and in detailed manner in the supplementary material. The ECDF

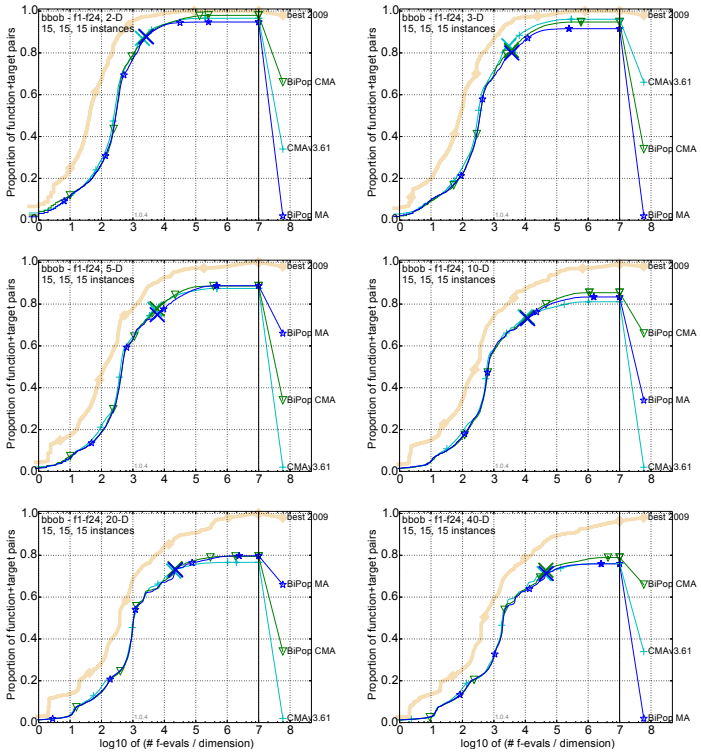

Fig. 11. Accumulated performance profiles of the BiPop-CMA(-ES), the BiPop-MA(-ES), and the production code CMAv3.61 on the set comprising all 24 test functions of Tab. II.

graphs are displayed for search space dimensionalities $N = 2, 3, 5, 10, 20,$ and 40 (in the graphics labeled as <N>-D). The horizontal axes display the logarithm of b/N (where b is labeled as "# f-evals" and N is the "dimension"). Simulation runs are performed up to a function budget b marked by the colored $\times$-crosses (values exceeding this b were obtained by a so-called bootstrap technique, see [12]). Additionally, the "utopia" ECDF profile of the best BBOB 2009 results[17] is displayed. As one can see, up to the respective function budget of about $b_{\text{stop}} = 1000N^2$ all strategies perform very similarly.[18] With respect to the ECDF plots there is not a clear "winner" and the CMAv3.61-ES production code does not consistently perform better than the much simpler generic (C)MA-ES implementations. This basically transfers also to the BBOB COCO ERT investigations (detailed results are to be found in the supplementary material, for a short discussion see below). In the following, we will summarize the main observations regarding ECDF results on the different function classes without presenting the graphs (that can be found in the supplementary material).

Regarding the ECDF performance of the problem class f_1 – f_5 one finds performance differences between BiPop-CMA and BiPop-MA mainly for dimensionality $N = 3$ and less pronounced for $N = 5$. A detailed view (not presented here) on the single function ECDFs reveals that in the $N = 3$ case the f_4 function (Rastrigin-Bueche) is responsible for these differences. This function seems hard for all CMA-versions in the BBOB test if the dimension gets greater than $N = 3$.

As for the low or moderate condition test functions f_6 – f_9 virtually no remarkable differences between the performance of the BiPop-CMA-ES, the BiPop-MA-ES, and the CMAv3.61-ES production code have been found. This statement holds also well for the subset of test functions with high condition f_{10} – f_{14}.

On the subset of multi-modal functions f_{15} – f_{19}, BiPop-CMA-ES and BiPop-MA-ES perform nearly equally well. The CMAv3.61-ES performs a bit better than BiPop-CMA-ES and BiPop-MA-ES. This is due to a certain performance advantage of the CMAv3.61-ES on Schaffer's F7 function f_{17} and f_{18}, which permanently reaches higher ECDF values (about 10% in all dimensions). The reasons for this performance advantage remain unclear.

For the multi-modal test functions with weak global structure, f_{20} – f_{24}, one can observe performance differences between BiPop-CMA-ES and BiPop-MA-ES with a performance advantage of the former. Having a look into the single function ECDF graphs (not shown here), however, does not reveal a clear picture. There are also cases where BiPop-CMA-ES and BiPop-MA-ES perform equally well. Even more astonishing, regarding ECDF the production code CMAv3.61 performs in most of the cases inferior to the BiPop-MA-ES and the BiPop-CMA-ES.

Summarizing the findings of these BBOB ECDF evaluations, w.r.t. the ECDF performance profiles, the newly derived MA-ES does not exhibit a severe performance degradation compared to the CMA-ES. Sometimes, it even performs a bit better than CMA-ES.

Alternatively, one can evaluate and compare the performance of the algorithms using the expected runtime (ERT) graphs (as for the definition of ERT, cp. Eq. (37)). BBOB COCO performance analysis have been done for the precision target $f_{\text{target}} = \min[f] + 10^{-8}$. One finds for BiPop-CMA-ES and BiPop-MA-ES similar performance behaviors. There is a certain performance advantage of the CMAv3.61-ES production code being statistically significant on some of the 24 test functions and search space dimensionalities (especially for the Sphere, Discus, and Attractive Sector function). It seems that CMAv3.61-ES has been "tuned" for the Sphere and Sharp ride. Interestingly, BiPop-CMA-ES and BiPop-MA-ES seems to excel on Gallagher for higher dimensions,

[17]This "utopia" profile represents the best performance results taken from *all* algorithms benchmarked in the 2009 GECCO competition.

[18]Since these graphs are aggregated performance plots, this does not exclude that the strategies perform differently on individual instances. However, a close scrutiny of the single performance plots did not reveal markable differences.

however, the Wilcoxon rank sum test implemented in BBOB did not report statistical significance. Note, there are some test functions where no ES version was able to get sufficiently close to the global optimizer: (Skew) Rastrigin, Schaffer F7, Griewank-Rosenbrock, Schwefel, Katsuuras and Lunacek bi-Rastrigin. However, except of Bent cigar and Sharp ridge all three BiPop-ES versions performed rather similarly. Thus, justifying the statement that one can switch to the MA-ES also from viewpoint of ERT.

VI. Summary and Conclusions

Based on a theoretical analysis of the CMA-ES, Fig. 1, it has been shown under the assumption of equal cumulation time constants $1/c_p$ and $1/c_s$ that one can remove the $\mathbf{p}$-path (line C13). Additionally it has been shown that one can also bypass the evolution of the covariance matrix, thus removing the "C" from the CMA-ES yielding the MA-ES, Fig. 3. The latter step is accompanied by an approximation assumption, which is increasingly violated for increasing population sizes since $c_w \to 1$ in (32) for $N = \text{const.} < \infty$. In spite of that, this algorithmic change (line M11, Fig. 3) does not severely affect the performance on population sizes as large as $\lambda = 4N^2$ as has been shown in Section IV-C. Moreover, using the MA-ES in the BiPop-ES as search engine does not significantly deteriorate the BiPop-CMA-ES performance on the standard BBOB COCO test bed.

The novel MA-ES does not evolve a covariance matrix. Thus, a matrix square root operation in terms of Cholesky or spectral value decomposition is not needed. Only matrix-vector and matrix-matrix operations are to be performed numerically (for example, complex eigenvalues due to numerical imprecision can not appear). This allows potentially for a more stable working ES. Furthermore, taking advantage of GPUs to speed-up calculations should be easy to accomplish.

Besides the increased algorithmic simplicity, the new matrix update rule, line M11 in Fig. 3 and Eq. (30), allows also for a novel interpretation of the $\mathbf{C}$-matrix adaptation via the $\mathbf{M}$-matrix adaptation, because $\mathbf{C} = \mathbf{M}\mathbf{M}^{\mathrm{T}}$: The change and adaptation of the $\mathbf{M}$-matrix is driven by contributions of

a) the deviation of the *selected* isotropically generated $\mathbf{z}\mathbf{z}^{\mathrm{T}}$ matrix from the isotropy matrix (being the identity matrix) and
b) the deviation of the evolution path matrix $\mathbf{s}\mathbf{s}^{\mathrm{T}}$ from isotropy (i.e. spherical symmetry)[19]

Thus, both contributions measure the departure of the ES-system (comprising the fitness function and the transformation in M5) from the Sphere model as seen from the mutations generated in line M4. In other words, from viewpoint of the $\mathbf{z}$-mutations in M4, the (C)MA-ES seeks to transform a function f with general ellipsoidal f-level sets into a function with spherical level sets.

Finally, the novel MA-ES might also be more attractive for a theoretical analysis due to a simpler pseudocode compared to the CMA-ES. There is only one evolution path and the $\mathbf{M}$-matrix update bypasses the $\mathbf{C}$ evolution and its square root operation. This might pave the way for a theoretical analysis of the dynamics of the ES system, yet a challenge for future research.

[19]These deviations must be seen in the sense of an average deviation over time.

References

[1] N. Hansen, S. Müller, and P. Koumoutsakos, "Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES)," *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.

[2] C. Igel, T. Suttorp, and N. Hansen, "A Computational Efficient Covariance Matrix Update and a $(1 + 1)$-CMA for Evolution Strategies," in *GECCO'06: Proceedings of the Genetic and Evolutionary Computation Conference*. New York, NY: ACM Press, 2006, pp. 453–460.

[3] G. Jastrebski and D. Arnold, "Improving Evolution Strategies through Active Covariance Matrix Adaptation," in *Proceedings of the CEC'06 Conference*. Piscataway, NJ: IEEE, 2006, pp. 2814–2821.

[4] H.-G. Beyer and B. Sendhoff, "Covariance Matrix Adaptation Revisited – the CMSA Evolution Strategy," in *Parallel Problem Solving from Nature 10*. Berlin: Springer, 2008, pp. 123–132.

[5] N. Hansen and A. Auger, "Principled Design of Continuous Stochastic Search: From Theory to Practice," in *Theory and Principled Methods for the Design of Metaheuristics*, Y. Borenstein and A. Moraglio, Eds. Berlin: Springer, 2014, pp. 145–180.

[6] T. Glasmachers, T. Schaul, Y. Sun, D. Wierstra, and J. Schmidhuber, "Exponential Natural Evolution Strategies," in *GECCO'10: Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2010, pp. 393–400.

[7] N. Hansen and A. Ostermeier, "Completely Derandomized Self-Adaptation in Evolution Strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[8] O. Ledoit and M. Wolf, "Honey, I Shrunk the Sample Covariance Matrix," *The Journal of Portfolio Management*, vol. 30, no. 4, pp. 110–119, 2004.

[9] S. Meyer-Nieberg and E. Kropat, "A New Look at the Covariance Matrix Estimation in Evolution Strategies," in *Operations Research and Enterprise Systems*. Springer, Berlin, 2014, pp. 157–172.

[10] H.-G. Beyer and M. Hellwig, "The Dynamics of Cumulative Step-Size Adaptation on the Ellipsoid Model," *Evolutionary Computation*, vol. 24, no. 1, pp. 25–57, 2016.

[11] N. Hansen, "Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed," in *Workshop Proceedings of the GECCO Genetic and Evolutionary Computation Conference*. ACM, 2009, pp. 2389–2395.

[12] S. Finck, N. Hansen, R. Ros, and A. Auger, "COCO Documentation, Release 15.03," LRI, Orsay, Tech. Rep., 2015. [Online]. Available: http://coco.gforge.inria.fr

[13] H.-G. Beyer, *The Theory of Evolution Strategies*, Natural Computing Series. Heidelberg: Springer, 2001.

[14] D. Arnold and H.-G. Beyer, "Performance Analysis of Evolutionary Optimization With Cumulative Step Length Adaptation," *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 617–622, 2004.

[15] A. Auger and N. Hansen, "Performance Evaluation of an Advanced Local Search Evolutionary Algorithm," in *Congress on Evolutionary Computation, CEC'05*. IEEE, 2005, pp. 1777–1784.

[16] I. Loshchilov, M. Schoenauer, and M. Sebag, "Alternative Restart Strategies for CMA-ES," in *Parallel Problem Solving from Nature 12, Part I*. Berlin: Springer, 2012, pp. 296–305.

[17] I. Loshchilov, "CMA-ES with Restarts for Solving CEC 2013 Benchmark Problem," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 369–376.

[18] N. Hansen, S. Finck, R. Ros, and A. Auger, "Real-Parameter Black-Box Optimization Benchmarking 2010: Noiseless Functions Definitions (compiled Nov. 17, 2015)," INRIA, Tech. Rep. RR-6829, 2015. [Online]. Available: http://hal.inria.fr/inria-00362633/en/

[19] O. Krause, D.R. Arbones, and C. Igel, "CMA-ES with optimal covariance Update and Storage Complexity," in *Proc. Adv. Neural Inf. Process. Syst.* Barcelona, Spain, 2016, pp. 370–378.

[20] O. Krause and T. Glasmachers, "CMA-ES with Multiplicative Covariance Matrix Updates," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*. Madrid, Spain, 2015, pp. 281–288.

VII. Supplementary Material: Simulations

This appendix presents a collection of additional simulations. The underlying experiments are explained in the respective sections of the main text.

A. Evolution dynamics for the $\lambda = 4N$ and $\lambda = 4N^2$ cases

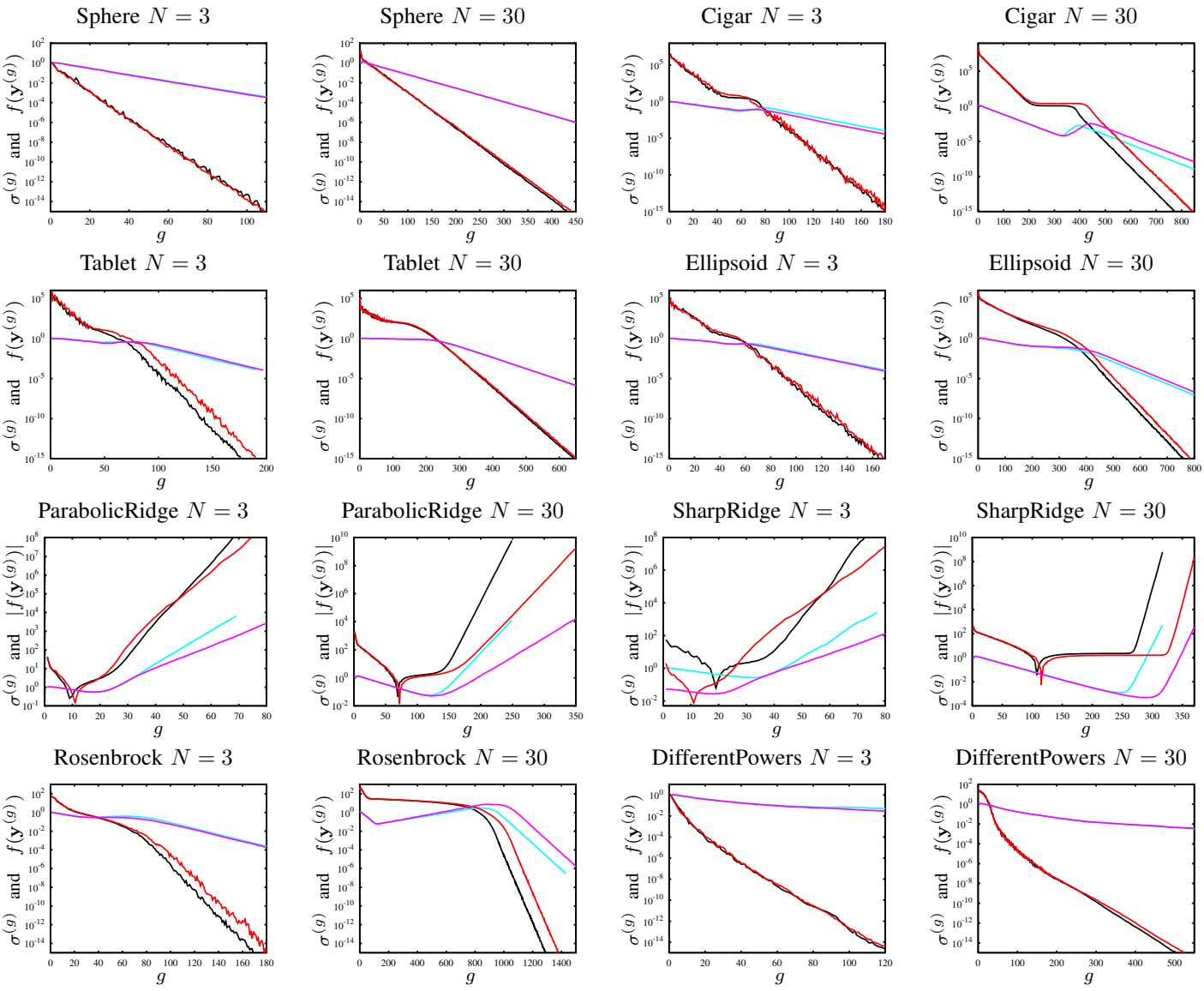

Fig. 12. The σ- and f- or $|f|$-dynamics of CMA-ES and MA-ES on the test functions of Tab. I for $N = 3$ and $N = 30$. The population sizes $\lambda = 12$, $\mu = 6$ for $N = 3$ and $\lambda = 120$, $\mu = 60$ for $N = 30$ have been used. The f-dynamics of the CMA-ES are in black and those of the MA-ES are in red. The σ-dynamics of the CMA-ES are in cyan and those of the MA-ES are in magenta. The curves are the averages of 20 independent ES runs (except the SharpRidge $N = 3$ case where 200 runs have been averaged and the Rosenbrock $N = 3$ case where 100 runs have been averaged). Note that the MA-ES exhibits premature convergence on the SharpRidge while the CMA-ES shows this behavior for the larger $N = 30$.

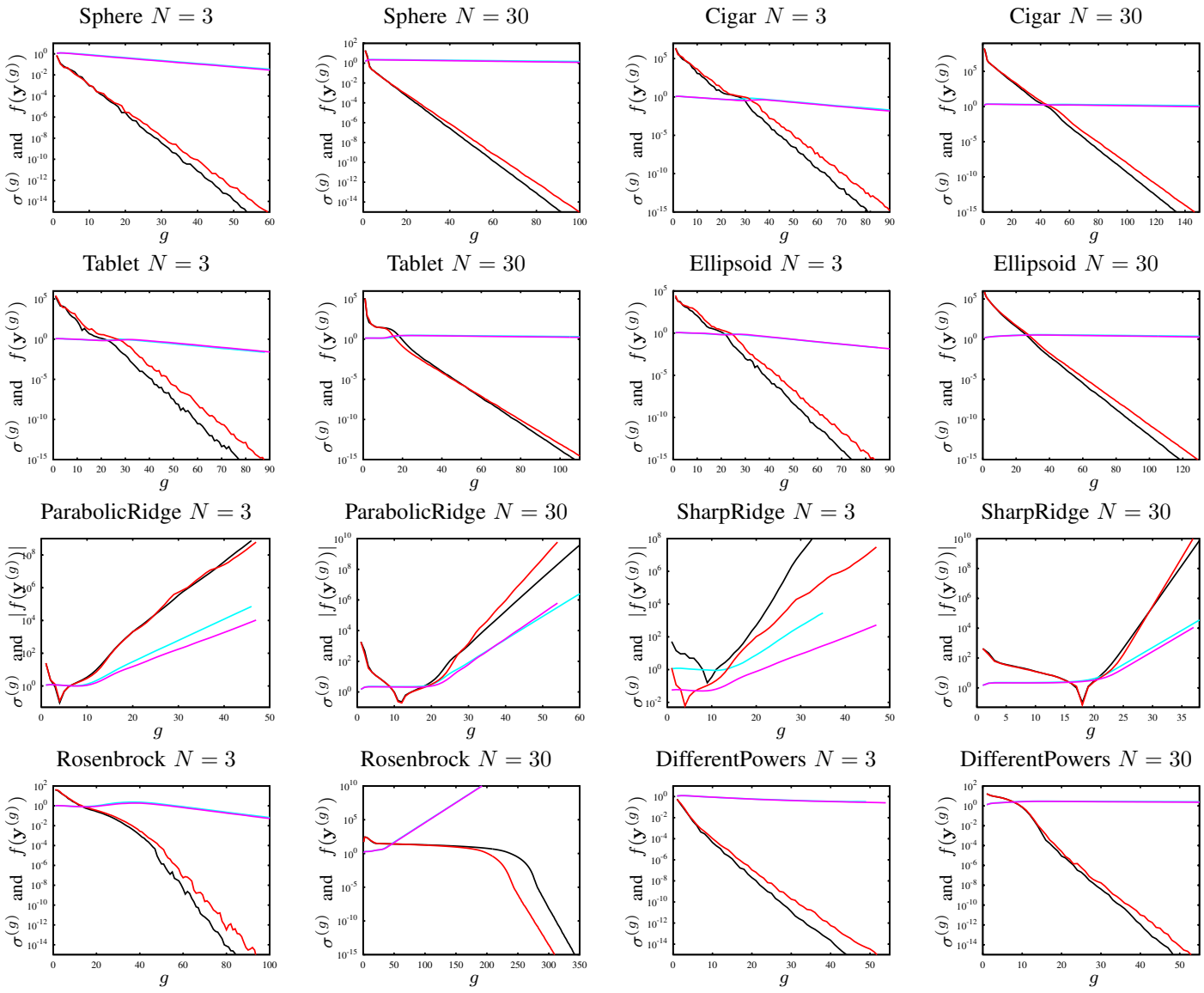

Fig. 13. The σ- and f- or $|f|$-dynamics of CMA-ES and MA-ES on the test functions of Tab. I for $N = 3$ and $N = 30$. The population sizes $\lambda = 36$, $\mu = 18$ for $N = 3$ and $\lambda = 3600$, $\mu = 1800$ for $N = 30$ have been used. The f-dynamics of the CMA-ES are in black and those of the MA-ES are in red. The σ-dynamics of the CMA-ES are in cyan and those of the MA-ES are in magenta. The curves are the averages of 20 independent ES runs (except the SharpRidge $N = 3$ case where 200 runs have been averaged and the Rosenbrock $N = 3$ case where 100 runs have been averaged). Note that the MA-ES exhibits premature convergence on the SharpRidge while the CMA-ES shows this behavior for the larger $N = 30$.

B. BBOB COCO Performance Evaluation: ECDF-Graphs

In Fig. 14, the aggregated performance profiles of the functions $f_1 - f_5$ are displayed for search space dimensionalities $N = 2, 3, 5, 10, 20$, and 40 (in the graphics labeled as <N>-D). The horizontal axes display the logarithm of b/N (where b is labeled as "# f-evals" and N is the "dimension"). Simulation runs are performed up to a function budget b marked by the colored $\times$-crosses (values exceeding this b were obtained by a so-called bootstrap technique, see [12]). Additionally, the "utopia" ECDF profile of the best BBOB 2009 results[20] is displayed. As one can see, up to the respective function budget of about $b_{\mathrm{stop}} = 1000N^2$ all strategies perform very similarly.[21] Performance deviations in Fig 14 are mainly observed for dimensionality $N = 3$ making a difference between BiPop-CMA and BiPop-MA. There is also a less pronounced difference for $N = 5$. A detailed view on the single function ECDFs reveals that in the $N = 3$ case the f_4 function (Rastrigin-Bueche) is responsible for the performance differences. This function seems hard for all CMA-versions in the BBOB test if the dimension gets greater than $N = 3$.

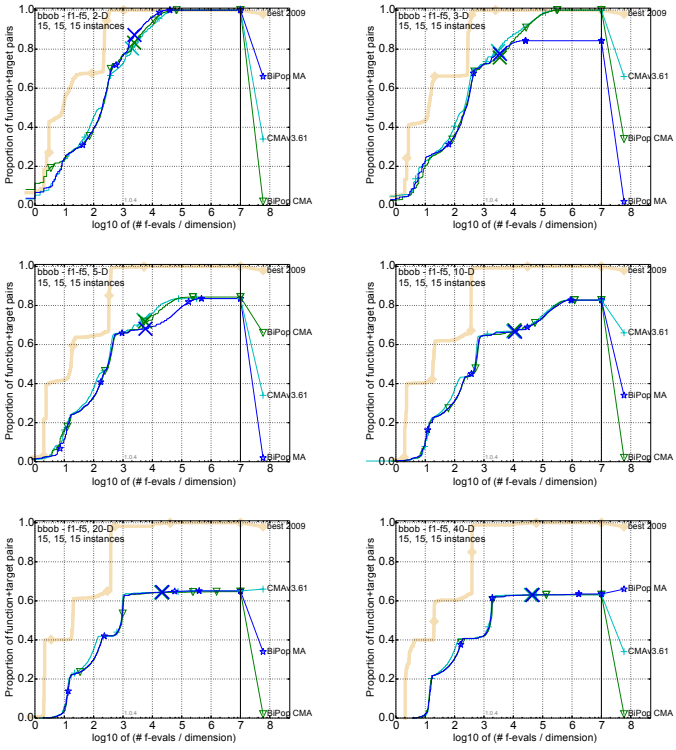

Fig. 14. Accumulated performance profiles of the BiPop-CMA(-ES), the BiPop-MA(-ES), and the production code CMAv3.61 on the test set of separable test functions.

Figure 15 provides the results for low or moderate condition test functions. There are virtually no differences between the performance of the BiPop-CMA-ES, the BiPop-MA-ES, and the CMAv3.61-ES production code. This statement holds also well for the subset of test functions with high condition,

[20]This "utopia" profile represents the best performance results taken from *all* algorithms benchmarked in the 2009 GECCO competition.

[21]Since these graphs are aggregated performance plots, this does not exclude that the strategies perform differently on individual instances. However, a close scrutiny of the single performance plots did not reveal markable differences.

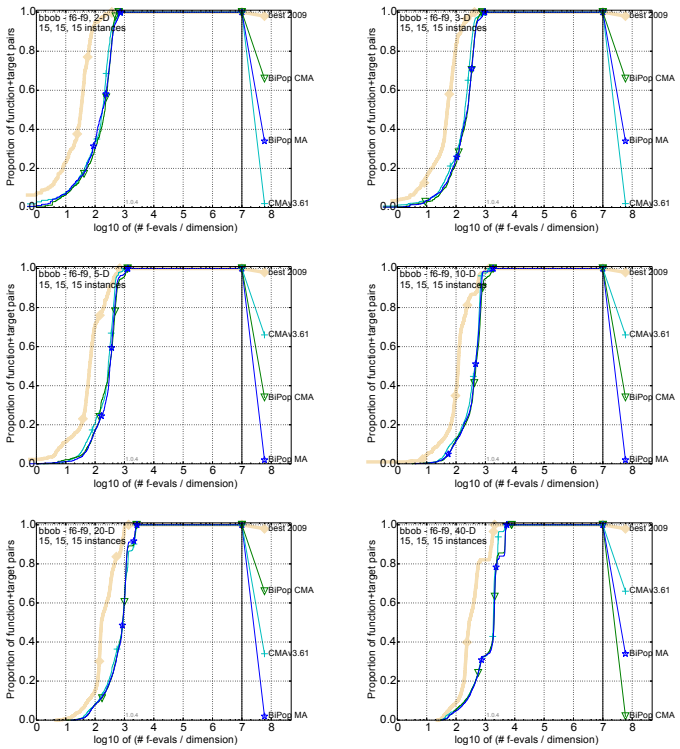

Fig. 15. Accumulated performance profiles of the BiPop-CMA(-ES), the BiPop-MA(-ES), and the production code CMAv3.61 on the test set low and moderate condition test functions.

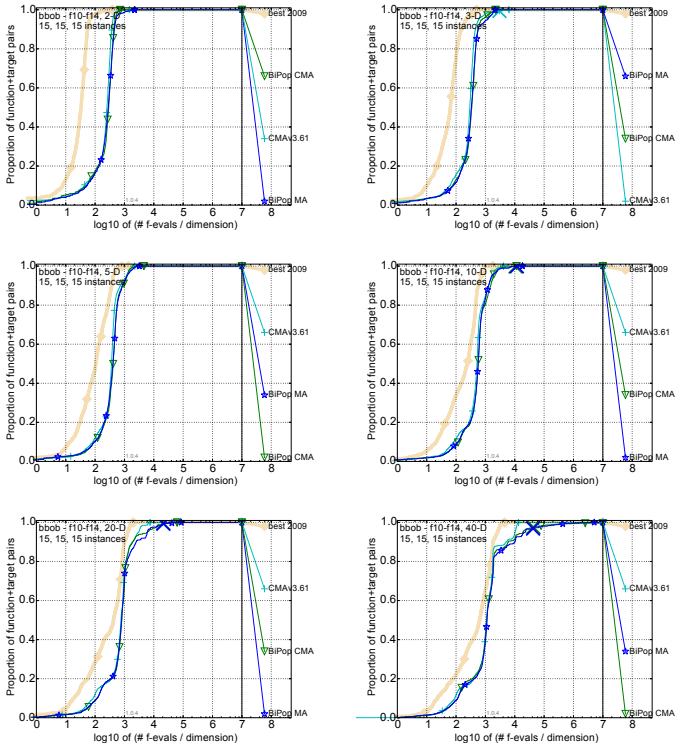

Fig. 16. Accumulated performance profiles of the BiPop-CMA(-ES), the BiPop-MA(-ES), and the production code CMAv3.61 on the test set of high condition test functions.

Fig. 16.

On the subset of multi-modal functions, Fig. 17, BiPop-CMA-ES and BiPop-MA-ES perform nearly equally well. The CMAv3.61-ES performs a bit better than BiPop-CMA-ES and BiPop-MA-ES. This is due to a certain performance advantage of the CMAv3.61-ES on Schaffer's F7 function f_{17} and f_{18}, which permanently reaches higher ECDF values (about 10% in all dimensions). The reasons for this performance advantage remains unclear.

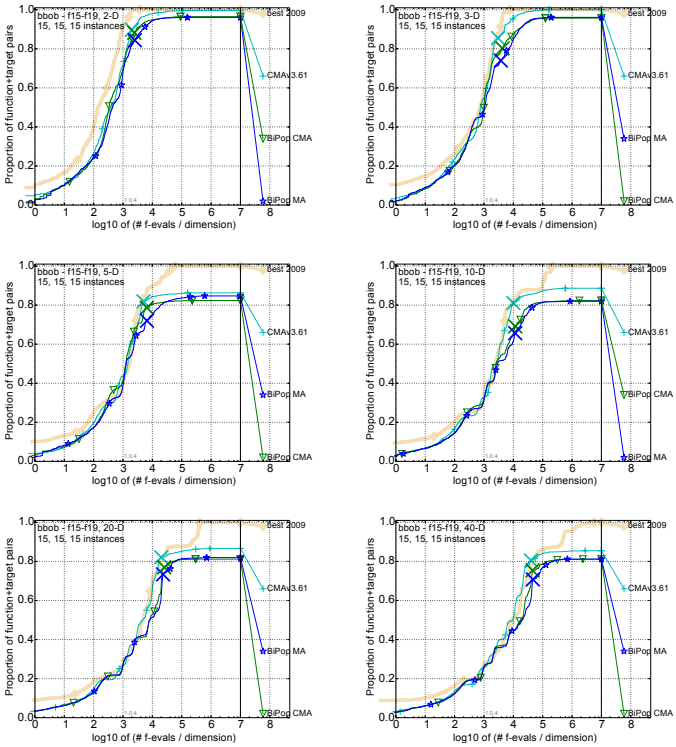

Fig. 17. Accumulated performance profiles of the BiPop-CMA(-ES), the BiPop-MA(-ES), and the production code CMAv3.61 on the test set of multi-modal test functions.

For the multi-modal test functions with weak global structure, Fig. 18, one can observe performance differences between BiPop-CMA-ES and BiPop-MA-ES with a performance advantage of the former. Having a look into the single function ECDF graphs (not shown here), however, does not reveal a clear picture. There are also cases where BiPop-CMA-ES and BiPop-MA-ES perform equally well. Even more astonishing, the production code CMAv3.61 performs in most of the cases significantly inferior to the BiPop-MA-ES and the BiPop-CMA-ES.

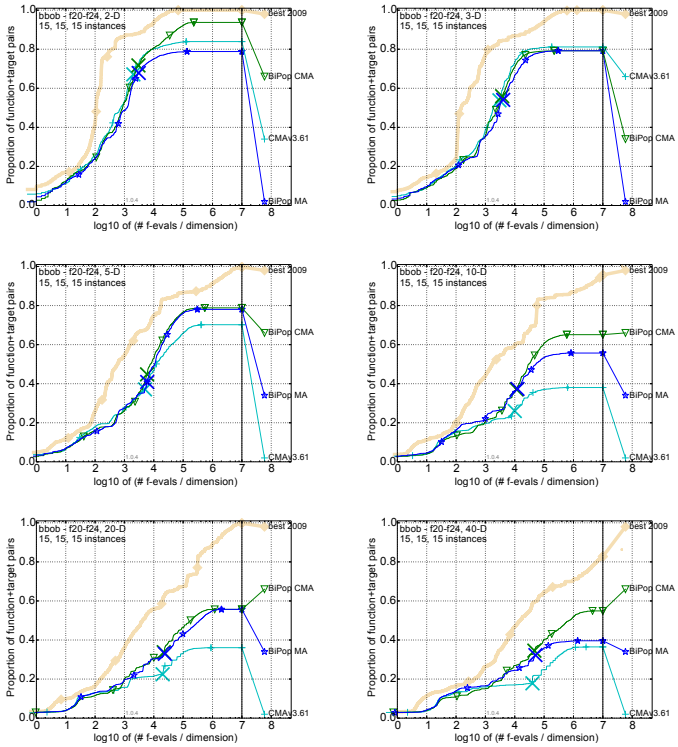

Fig. 18. Accumulated performance profiles of the BiPop-CMA(-ES), the BiPop-MA(-ES), and the production code CMAv3.61 on the test set of multi-modal test functions with weak global structure.

C. BBOB COCO ERT-Graphs

Alternatively, one can evaluate and compare the performance of the algorithms using the expected runtime (ERT) graphs (as for the definition of ERT, cp. Eq. (37)). Figure 19 shows the $\mathrm{ERT}(N)$ functions for the precision target $f_{\mathrm{target}} = \min[f] + 10^{-8}$. As one can see on most of the test functions, BiPop-CMA-ES and BiPop-MA-ES perform very similarly. As already observed in the ECDF graphs, the CMAv3.61-ES production code performs better on some of the functions, especially on the Sphere, Sharp ride, Bent cigar and to a certain extend on Rastrigin. It seems that CMAv3.61-ES has been "optimized" for the Sphere and Sharp ride. Interestingly, BiPop-CMA-ES and BiPop-MA-ES excel on Gallagher for higher dimensions. NB, there are some test functions where no ES version was able to get sufficiently close to the global optimizer: (Skew) Rastrigin, Schaffer F7, Griewank-Rosenbrock, Schwefel, Katsuuras and Lunacek bi-Rastrigin. However, except of Bent cigar and Sharp ridge all three BiPop-ES versions performed rather similarly. Thus, justifying the statement that one can switch to the MA-ES also from viewpoint of ERT.

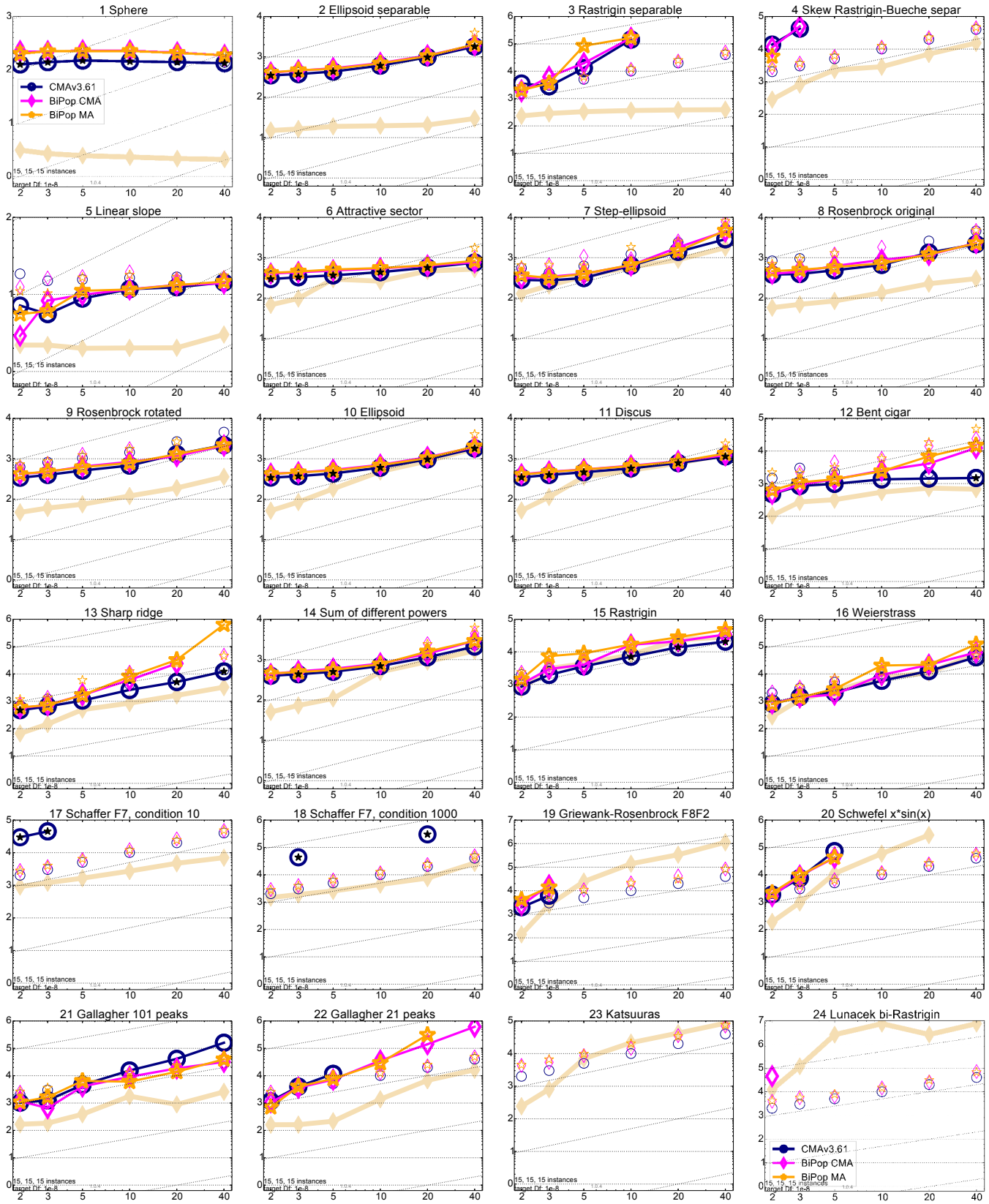

Fig. 19. Vertical axes: $\log_{10}$ of the expected number of function evaluations (ERT) divided by dimension N needed to get closer as 10^{-8} to the optimum; horizontal axes: search space dimension N. Slanted grid lines indicate quadratic scaling with N. Symbols used: CMAv3.61-ES: ○, BiPop-CMA-ES: ◇, BiPop-MA-ES: ★. Light symbols give the maximum number of function evaluations from the longest trial divided by N. Black stars indicate a statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions. [This caption has been adapted from the GECCO BBOB competition LaTeX template.]