Alternative Step-Size Adaptation Rule for the Matrix Adaptation Evolution Strategy

Eryk Warchulski[a] and Jarosław Arabas[b]
Institute of Computer Science, Warsaw University of Technology, Warsaw, Poland

Keywords: Evolution Strategies, Step-Size Adaptation, Optimization.

Abstract: In this paper, we present a comparison of various step-size adaptation rules for the Matrix Adaptation Evolution Strategy (MA-ES) algorithm, which is a lightweight version of the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). In contrast to CMA-ES, MA-ES does not require to invoke numerically complex covariance matrix factorization. We take a step further in this direction and provide a quantitative assessment of alternative step-size rules to Cumulative Step Adaptation (CSA), which is considered to be a state-of-the-art method. Our study shows that generalized 1/5-th success rules like the Previous Population Midpoint Fitness rule (PPMF) or Population Success Rule (PSR) exhibit comparable or superior performance to the CSA rule on standard benchmark problems, including the CEC benchmark suites.

1 INTRODUCTION

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen et al., 2003) exhibits outstanding performance on various optimization problems and is considered a state-of-the-art method in the evolution strategy algorithms family. However, its performance is occupied by the high numerical complexity associated with matrix factorization. There have been attempts to decrease its complexity over the years, which led to the development of the Matrix Adaptation Evolution Strategy (MA-ES) (Beyer and Sendhoff, 2017). MA-ES takes radical steps to decrease the computational time and memory demands of CMA-ES by excluding the need for the covariance matrix factorization in the process of sampling the new points. In consequence, the evolution path is no longer needed. Such design decisions result in a lightweight version of CMA-ES with only slightly deteriorated performance. Still, the authors of MA-ES equipped introduced optimizer with the Cumulative Step-Size Adaptation (CSA) (Hansen and Ostermeier, 2001) rule to adapt the mutation strength. While it is a standard and highly performant method to control the mutation step, it exhibits certain limitations like sample distribution dependency, sensitivity to the population size, and complexity due to reliance on the evolution path (Ait Elhara et al., 2013), (Hansen, 2008).

We demonstrate that equipping MA-ES with simpler and less demanding rules to control mutation strength could be beneficial in terms of performance and simplicity. We provide an experimental study to compare CSA with other alternative step-size adaptation rules.

The paper is organized as follows. Section 2 outlines the MA-ES algorithm and compares it to the CMA-ES. In section 3, we introduce and describe step-size adaptation mechanisms considered in this paper. Section 4 contains the empirical evaluation of different step-size adaptation rules combined with the MA-ES algorithm. We perform experiments using the set of simple functions adopted from (Hansen et al., 2014) and (Krause et al., 2017), and additionally, we perform benchmarking using the CEC 2017 test suite (Awad et al., 2016). Section 5 concludes the paper.

2 MATRIX ADAPTATION STRATEGY

The MA-ES algorithm is outlined in Alg.1 while Alg. 2 shows the canonical version of CMA-ES. The consecutive steps of MA-ES resemble the classical version of CMA-ES (Hansen, 2023). The state of the algorithm consists of three parameters which are updated in every iteration t: the expectation vector $\mathbf{m}^t$,

[a] https://orcid.org/0000-0003-1416-7031
[b] https://orcid.org/0000-0002-5699-947X

Warchulski, E. and Arabas, J.
Alternative Step-Size Adaptation Rule for the Matrix Adaptation Evolution Strategy.
DOI: 10.5220/0013012800003837

In *Proceedings of the 16th International Joint Conference on Computational Intelligence (IJCCI 2024)*, pages 151-158
ISBN: 978-989-758-721-4; ISSN: 2184-3236

the matrix $\mathbf{M}^t$, and the evolution path vector $\mathbf{s}^t$ used for the step-size adaptation. After initialization of parameters (line 2), the algorithm samples in every iteration the set of λ difference vectors $\mathbf{d}^t$ (line 3), which are utilized to define points in the search space $\mathbf{x}^t$ (line 4). The sets of search points $\mathbf{X}^t_{1:\lambda}$ and difference vectors $\mathbf{D}^t_{1:\lambda}$ are sorted according to their fitness. The fraction of μ points with the best fitness and their corresponding difference vectors are used to update the expectation vector $\mathbf{m}^t$ (line 10), the evolution path vector $\mathbf{s}^t$ (line 11) and the matrix $\mathbf{M}^t$ (line 12). Similarly to CMA-ES, the step-size multiplier σ^t is updated according to the CSA rule (line 13).

The significant difference between MA-ES and CMA-ES is expressed in the form of matrix $\mathbf{M}^t$ and the absence of evolution path $\mathbf{p}^t$. In contrast to CMA-ES, the matrix $\mathbf{M}^t$ is updated by taking into account two summands: the outer product of the step-size evolution path $\mathbf{s}^t$ and the outer products of weighted difference vectors.

Due to limited space, we cannot rephrase the derivation of the matrix $\mathbf{M}^t$ from the covariance matrix $\mathbf{C}^t$ and removal of its evolution path $\mathbf{p}^t$. Interested readers are referred to the original work of Beyer (Beyer and Sendhoff, 2017).

Algorithm 1: Outline of MA-ES considered in the paper.

1: $t \leftarrow 1$
2: initialize$(\mathbf{m}^1, \sigma^1, \lambda, \ldots)$
3: $\mathbf{s}^1 \leftarrow \mathbf{0}$, $\mathbf{M}^1 \leftarrow \mathbf{I}_D$
4: **while** !stop **do**
5:   **for** $i = 1$ **to** λ **do**
6:     $\mathbf{d}^t_i \sim N(\mathbf{0}, \mathbf{M}^t)$
7:     $\mathbf{x}^t_i \leftarrow \mathbf{m}^t + \sigma^t \mathbf{d}^t_i$
8:   **end for**
9:   evaluate $(\mathbf{X}^t)$
10:   $\mathbf{m}^{t+1} \leftarrow \langle \mathbf{X}^t \rangle^\mu_w$ where $\langle \mathbf{X}^t \rangle^\mu_w = \sum_{i=1}^{\mu} w_i \mathbf{X}^t_{i:\mu}$
11:   $\mathbf{s}^{t+1} \leftarrow (1 - c_s)\mathbf{s}^t + \sqrt{\mu c_s(2 - c_s)} \cdot \langle \mathbf{D}^t \rangle^\mu_w$
12:   $\mathbf{M}^{t+1} \leftarrow \mathbf{M}^t[\mathbf{I}_n + \frac{c_1}{2}(\mathbf{M}^t_s - \mathbf{I}_n) + \frac{c_w}{2}(\mathbf{M}^t_D - \mathbf{I}_n)]$ where $\mathbf{M}^t_s = \mathbf{s}^t(\mathbf{s}^t)^T$, $\mathbf{M}^t_D = \langle \mathbf{D}^t(\mathbf{D}^t)^T \rangle^\mu_w$
13:   $\sigma^{t+1} \leftarrow \sigma^t \exp\left(\frac{c_s}{d_\sigma}\left(\frac{\|\mathbf{s}^{t+1}\|}{E\|N(\mathbf{0},\mathbf{I})\|} - 1\right)\right)$
14:   $t \leftarrow t + 1$
15: **end while**

Algorithm 2: Outline of classic CMA-ES.

1: $t \leftarrow 1$
2: initialize$(\mathbf{m}^1, \sigma^1, \mathbf{C}^1)$
3: $\mathbf{p}^1 \leftarrow \mathbf{0}$, $\mathbf{s}^1 \leftarrow \mathbf{0}$
4: **while** !stop **do**
5:   **for** $i = 1$ **to** λ **do**
6:     $\mathbf{d}^t_i \sim N(\mathbf{0}, \mathbf{C}^t)$
7:     $\mathbf{x}^t_i \leftarrow \mathbf{m}^t + \sigma^t \mathbf{d}^t_i$
8:   **end for**
9:   evaluate $(\mathbf{X}^t)$
10:   $\mathbf{m}^{t+1} \leftarrow \langle \mathbf{X}^t \rangle_w$
11:   $\mathbf{s}^{t+1} \leftarrow (1 - c_s)\mathbf{s}^t + \sqrt{\mu c_s(2 - c_s)} \cdot (\mathbf{C}^t)^{-\frac{1}{2}} \langle \mathbf{D}^t \rangle^\mu_w$
12:   $\mathbf{p}^{t+1} \leftarrow (1 - c_p)\mathbf{p}^t + \sqrt{\mu c_p(2 - c_p)} \cdot \langle \mathbf{D}^t \rangle^\mu_w$
13:   $\mathbf{C}^{t+1} \leftarrow (1 - c_1 - c_\mu)\mathbf{C}^t + c_1 \mathbf{C}^t_1 + c_\mu \mathbf{C}^t_\mu$ where $\mathbf{C}^t_\mu = \frac{1}{\mu_{\text{eff}}} \langle \mathbf{D}^t \rangle^\mu_w$, $\mu_{\text{eff}} = \sum_{i=1}^{\mu} (w_i)^2$, $\mathbf{C}^t_1 = \mathbf{p}^t(\mathbf{p}^t)^T$
14:   $\sigma^{t+1} \leftarrow \sigma^t \exp\left(\frac{c_s}{d_\sigma}\left(\frac{\|\mathbf{s}^{t+1}\|}{E\|N(\mathbf{0},\mathbf{I})\|} - 1\right)\right)$
15:   $t \leftarrow t + 1$
16: **end while**

3 STEP-SIZE ADAPTATION

The control of mutation strength in evolution strategies proved to be a crucial mechanism for the convergence of evolution strategy optimizers (Hansen and Auger, 2014). In this study, we want to take a step further toward a less numerically demanding CMA-ES-based method by assessing lightweight alternative step-size adaptation methods to the CSA. Although the CSA derived in (Beyer and Sendhoff, 2017) does not require computing inversion of $\sqrt{\mathbf{C}}$ as in the classic CMA-ES (line 11 in Algorithm 2), it still involves vector operations and inherits all drawbacks of the CSA pointed in (Beyer and Arnold, 2003), and (Hansen, 2008). Therefore, we will only consider methods based on the Rechenberg's 1/5-th success rule (Rechenberg, 1994) or techniques derived from line-search methods (Salomon, 1998). Naturally, we must exclude from our study methods like xNES-SA (Glasmachers et al., 2010) or its derivations proposed in (Krause et al., 2017) which are based on matrix computations.

3.1 Cumulative Step-Size Adaptation

The core mechanism of CSA (Hansen and Ostermeier, 2001) focuses on the norm of evolution path $\mathbf{s}^t$ which accumulates over iterations the values of expectation vectors $\mathbf{m}^t$ scaled by $\langle \mathbf{D}^t \rangle^\mu_w$ — see Algorithm 3.

Algorithm 3: Cumulative Step-Size Adaptation (CSA).

$$\mathbf{s}^{t+1} \leftarrow (1-c_s)\mathbf{s}^t + c_s\sqrt{\mu_{\text{eff}}}\langle \mathbf{D}^t \rangle_w^{\mu}$$

$$\sigma^{t+1} \leftarrow \sigma^t \exp\left(\frac{c_s}{d_\sigma}\left(\frac{\|\mathbf{s}^{t+1}\|}{E\|N(\mathbf{0},\mathbf{I})\|} - 1\right)\right)$$

The philosophy behind CSA is based on the following two phenomena. The first phenomenon is related to the distance between the expectation vector $\mathbf{m}^t$ and the optimum of the fitness function. If they are close, then the optimum is surrounded by the population of points. Then, the selection will prefer shorter difference vectors $\mathbf{d}_i^t$, reducing the length of $\mathbf{s}^t$. Thus, it will also effect in reducing the value of σ^t. The further the expectation vector $\mathbf{m}^t$ from the optimum, the longer difference vectors will be selected, which will result in the opposite effect, i.e., values of $\mathbf{s}^s$ and of σ^t will be increased. The second phenomenon is the correlation between consecutive evolution paths $\mathbf{s}^t$. If the correlation is positive, then the values of σ^t will be increased. Otherwise, a negative correlation will lead to decreased values of σ^t.

3.2 Previous Population Midpoint Fitness

Previous Population Midpoint Fitness (PPMF) (Warchulski and Arabas, 2021) was introduced as an attempt to generalize the Rechenberg's 1/5-th success rule for $(\mu/\mu_w,\lambda)$-ES algorithms. The PPMF is depicted on Algorithm 4.

Algorithm 4: Previous Population Midpoint Fitness (PPMF).

$$\overline{\mathbf{m}}^{t-1} \leftarrow \frac{1}{\lambda}\sum_{i=1}^{\lambda}\mathbf{X}_i^{t-1}$$

evaluate $(\overline{\mathbf{m}}^{t-1})$

$$p_s^t \leftarrow \left|\{i : q(\mathbf{X}_i^t) < q(\overline{\mathbf{m}}^{t-1})\}\right|/\lambda$$

$$\sigma^{t+1} \leftarrow \sigma^t \exp\left(\frac{1}{d_\sigma}\cdot\frac{p_s^t - p_t}{1-p_t}\right)$$

The method is inspired by the step-size adaptation mechanism employed in $(1+1)$-CMA-ES (Arnold and Hansen, 2010) and utilizes the observations made in (Arabas and Biedrzycki, 2017) about the positive impact of the midpoint on the evolution algorithms performance. In each iteration, PPMF estimates the success probability p_s^t by calculating the ratio of points from the current generation with better fitness value than the arithmetic midpoint from the previous generation $\overline{\mathbf{m}}^{t-1}$. The step-size is adapted in an exponential fashion. The method is controlled via damping factor d_σ and target probability p_t parameters.

3.3 Median Success Rule

Another attempt to generalize the 1/5-th success rule for $(\mu/\mu_w,\lambda)$-ES is the Median Success Rule (MSR) (Ait Elhara et al., 2013) — see Algorithm 5.

Algorithm 5: Median Success Rule (MSR).

$$K_s \leftarrow \left|\{i : q(\mathbf{X}_i^t) < q(\mathbf{X}_{k:\lambda-1})\}\right|/\lambda$$

$$z^t \leftarrow \frac{2}{\lambda}\cdot\left(K_s - \frac{\lambda+1}{2}\right)$$

$$p_s^{t+1} \leftarrow (1-c_\sigma)p_s^t + c_\sigma \cdot z^t$$

$$\sigma^{t+1} \leftarrow \sigma^t \exp\left(\frac{p_s}{d_\sigma}\right)$$

The core idea of MSR is to compare the current generation's fitness values to the chosen k-th percentile of fitness values from the previous generation. The method estimates the success probability p_s, which is used to calculate normalizing statistic z^t. The step-size is adapted exponentially according to the smoothed value of z^t and damping factor d_σ. Parameters of MSR are the percentile k, the smoothing factor c_σ, and the dumping factor d_σ.

3.4 Population Success Rule

The Population Success Rule (PSR), which was introduced in (Loshchilov, 2015), is presented as Algorithm 6.

Algorithm 6: Population Success Rule (PSR).

$r^t, r^{t-1} \leftarrow$ ranks of q^t, q^{t-1} in $q^t \cup q^{t-1}$

$$z_{\text{psr}}^t \leftarrow \frac{\sum_{i=1}^{\lambda} r^{t-1}(i) - r^t(i)}{\lambda^2} - z^\star$$

$$z^{t+1} \leftarrow (1-c_\sigma)z^t + c_\sigma z_{\text{psr}}^t$$

$$\sigma^{t+1} \leftarrow \sigma^t \exp\left(\frac{z^{t+1}}{d_\sigma}\right)$$

The PSR is derived from the MSR with the assumption that the base method may be enhanced if the success probability p_s^t is calculated by taking into account fitness values from the previous and current generations. In each iteration, the PSR constructs rank r^t and r^{t-1} of points from current and previous generations using the set of mixed fitness values $q^t \cup q^{t-1}$. Then, ranks are used to calculate enhanced z^t statistics. Further steps are the same as in the MSR method. The PSR is controlled by target probability $z^\star$, damping factor d_σ, and exponential smoothing factor c_σ.

3.5 Two-Point Adaptation

The Two-point Adaptation (TPA) rule was introduced in (Hansen, 2008). In contrast to CSA or generalized versions of the 1/5-th success rule, TPA relies solely on the optimizer runtime trajectory and does not assume an internal model of optimality regarding the step-size values. Algorithm 7 outlines the TPA.

Algorithm 7: Two-point adaptation (TPA).

1: $q_+ \leftarrow q\left(\mathbf{m}^t + \alpha'\sigma^t \langle \mathbf{D}^t \rangle_w^{\mu}\right)$
2: $q_- \leftarrow q\left(\mathbf{m}^t - \alpha'\sigma^t \langle \mathbf{D}^t \rangle_w^{\mu}\right)$
3: $\alpha_{\text{act}} \leftarrow \mathbb{I}_{q_- < q_+}\{-\alpha + \beta < 0\} + \mathbb{I}_{q_- \geq q_+}\{\alpha > 0\}$
4: $\alpha_s \leftarrow (1 - c_\alpha)\alpha_s + c_\alpha \alpha_{\text{act}}$
5: $\sigma^{t+1} \leftarrow \sigma^t \exp(\alpha_s)$

The method requires computing fitness of two additional points, i.e., q_- and q_+. These values are used to calculate the α_s parameter employed to update step-size σ^t values exponentially. The TPA relies on the following parameters: smoothing factor c_σ, test width coefficient α', changing factor α, and update bias β.

4 NUMERICAL VALIDATION

We performed two types of numerical experiments to assess the overall dynamic of MA-ES equipped with different step-size adaptation mechanisms.

The convergence rates were evaluated by running each optimizer on a set of basic optimization problems with different properties. As a well-conditioned problem, we used the sphere function. For ill-conditioned problems, we selected cigar and ellipsoid functions. The ellipsoid function was parameterized by the condition coefficient k. We also included the Rosenbrock function as an example of a non-convex problem, which is challenging for solvers to optimize due to the narrow valley. Equations 1-4 present the set of considered functions. For each problem and optimizer, we ran 50 runs independently and recorded the best-so-far value for the following dimensions: $D = 10, 50, 100, 200$. Each optimizer was terminated after $100 \cdot D$ iterations and was started in point $\mathbf{x}_0 = [100, \ldots, 100]^D$. The results from experiments were averaged and presented as convergence curves.

The second type of experiment was the benchmarking using the standard CEC'2017 suite of problems. We investigated the performance of MA-ES, coupled with considered step-size adaptation, on different classes of optimization functions with different

Optimization problems used to assess the convergence dynamics of different MA-ES variants: (1) Sphere (2) Ellipsoid (3) Cigar (4) Rosenbrock.

$$q(\mathbf{x}) = \sum_{i=1}^{D} x_i^2 \tag{1}$$

$$q(\mathbf{x}) = \sum_{i=1}^{D} k^{\frac{i-1}{D-1}} x_i^2 \tag{2}$$

$$q(\mathbf{x}) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2 \tag{3}$$

$$q(\mathbf{x}) = \sum_{i=1}^{D-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2\right] \tag{4}$$

difficulty and dimension numbers. The CEC'2017 benchmark suite is split into four classes of optimization problems: unimodal (F1-F3), multimodal (F4-F10), hybrid functions (F11-F20), and composition functions (F21-F30). Hybrid and complex functions are the product of composing selected unimodal and multimodal functions in two different manners. To resemble real-world problems, the hybrid functions divide the decision variables into components with different properties. In contrast, the composition functions merge the properties of selected component (unimodal or multimodal) functions. Additionally, each function is geometrically transformed by shifting, rotating, or scaling. We conducted numerical experiments on CEC'2017 following the rules specified in (Awad et al., 2016). The only difference between the official specification and our setup was that we did not exclude the sum of different powers functions from the benchmark set. The specification authors removed this function after the competitions, although most software libraries implement or re-implement CEC functions with the mentioned function. The results of the experiments are demonstrated as ECDF plots (Hansen, 2018) aggregated over each problem class.

4.1 Parameter Setup

Each MA-ES variant shares the same standard values for general parameters like the population size λ, or matrix update coefficients c_1, c_w suggested in (Beyer and Sendhoff, 2017). The initial value for step-size σ^0 was set to 1.

Specific parameters of step-size adaptation rules were set to the values recommended by their authors. We treat each adaptation rule as a drop-in replacement, so we perform no parameter tuning. Contrary to the opinion expressed in (Krause et al., 2017), we do not believe that experiments on non-tunned methods may lead to biases and unfair comparisons. Each considered step-size adaptation mechanism was introduced with the recommended parameter values. We assume that the authors investigated their methods and recommended values shall ensure satisfactory performance on various optimization problems.

The parameters for step-size adaptation mechanisms used to conduct numerical experiments are listed below:

1. CSA: following the (Beyer and Sendhoff, 2017), we set the damping factor to $d_\sigma = 1 + c_s + 2\max(0, \sqrt{\frac{\mu_{\text{eff}}-1}{D+1}} - 1)$ where $c_s = \frac{\mu_{\text{eff}}+2}{\mu_{\text{eff}}+D+5}$
2. TPA: following the (Hansen, 2008), we set $\alpha' = 0.5$, $\alpha = 0.5$, $\beta = 0$, and $c_\sigma = 0.3$
3. MSR: following the (Ait Elhara et al., 2013), we set $k = 0.3\lambda$, $c_\sigma = 0.3$, and damping factor to $d_\sigma = \frac{2D-2}{D}$
4. PPMF: following the (Warchulski and Arabas, 2021), we set $d_\sigma = 0.2$, and $p_t = 0.1$
5. PSR: following the (Loshchilov, 2015), we set $z^\star = 0.25$, $c_\sigma = 0.3$, $d_\sigma = 1$.

All performed experiments can be reproduced by using source code and containerized environment available in the repository https://github.com/ewarchul/maes-2024.

4.2 Convergence Dynamics

The convergence curves obtained for the sphere function are presented in Fig. 1. The results show that alternative methods exhibit faster convergence for almost every dimension than CSA. For dimension $D = 10$, the PPMF is the slowest method, but its dynamic differs for greater dimensions where it, together with MSR, outperforms other methods. Such behavior may indicate too aggressive step-size reduction, leading to premature convergence on multimodal functions. On both ellipsoid functions and the cigar function depicted respectively in Fig. 5, Fig. 2, and Fig. 4 similar effects can be observed. The 1/5-th success rule variants and TPA rapidly reduce step-size, but the gap between them and CSA decreases with the growing dimension. The performance of each MA-ES variant is similar on the Rosenbrock function in Fig 3. In 10 dimensions, the PPMF reveals the worst convergence, but with increasing dimensions - the gap between variants diminishes.

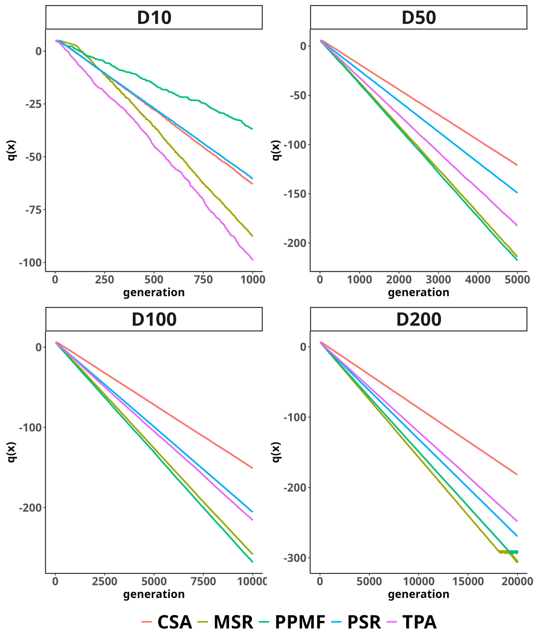

Figure 1: Semi-log convergence plot of the best-so-far solution for MA-ES coupled with different step-size adaptation rules on the sphere function in 10, 50, 100, and 200 dimensions.

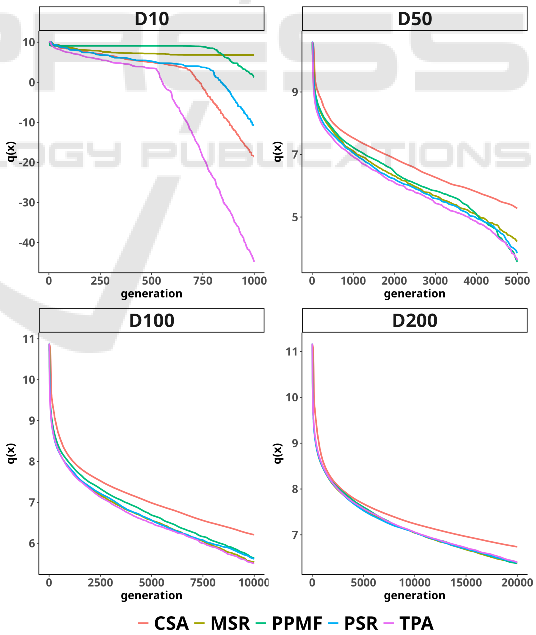

Figure 2: Semi-log convergence plot of the best-so-far solution for MA-ES coupled with different step-size adaptation rules on the ellipsoid function with condition coefficient $k = 10$ in 10, 50, 100, and 200 dimensions.

4.3 CEC'2017 Benchmarking

According to the benchmark results for problems in 10 dimensions depicted in Fig. 6, on unimodal prob-

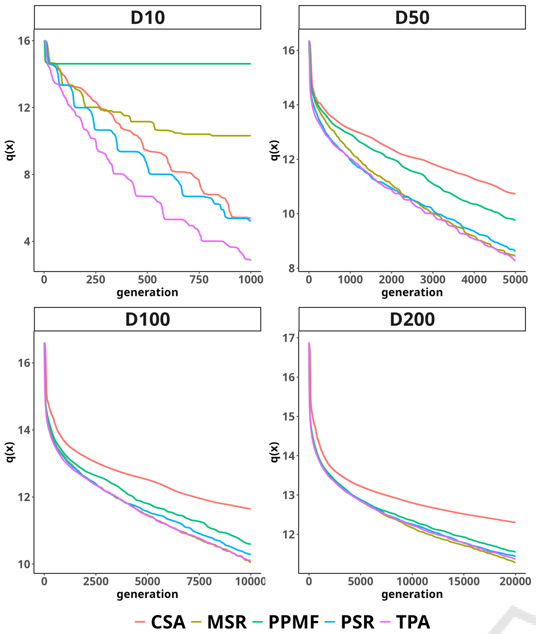

Figure 3: Semi-log convergence plot of the best-so-far solution for MA-ES coupled with different step-size adaptation rules on the ellipsoid function with condition coefficient $k = 100$ in 10, 50, 100, and 200 dimensions.

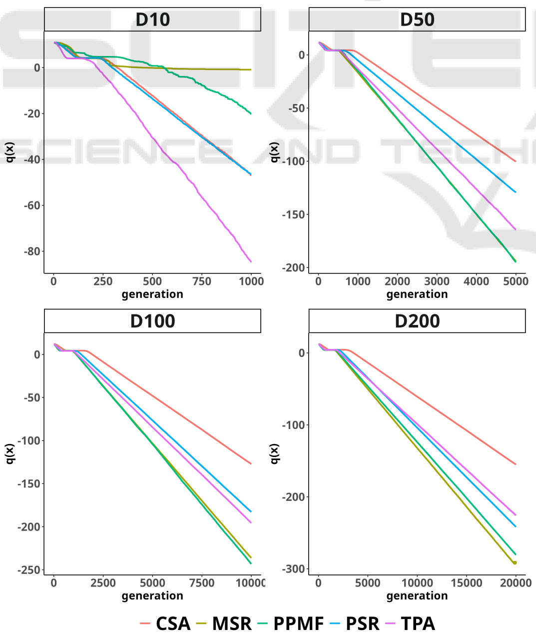

Figure 4: Semi-log convergence plot of the best-so-far solution for MA-ES coupled with different step-size adaptation rules on the cigar function in 10, 50, 100, and 200 dimensions.

lems, the MA-ES coupled with CSA outperformed each alternative method. However, the performance on more complex functions is comparable between

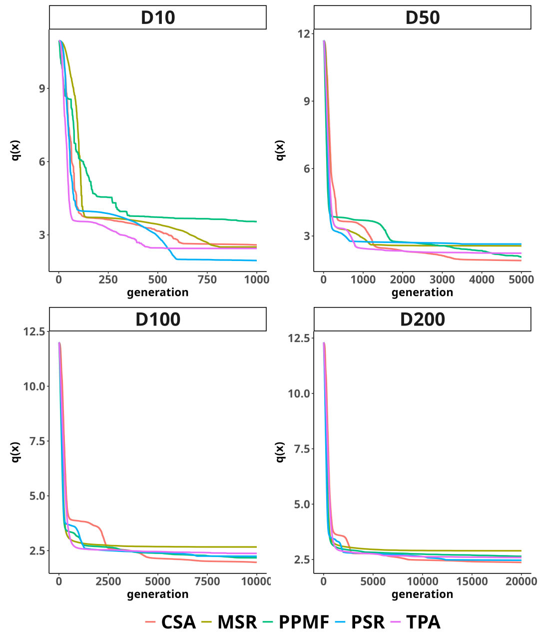

Figure 5: Semi-log convergence plot of the best-so-far solution for MA-ES coupled with different step-size adaptation rules on the Rosenbrock function in 10, 50, 100, and 200 dimensions.

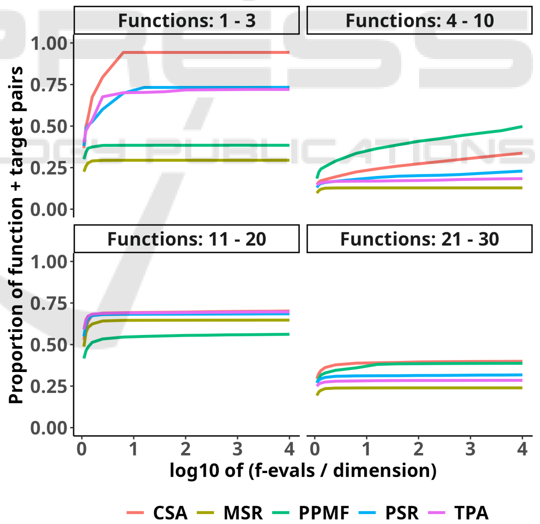

Figure 6: Results for MA-ES coupled with different step-size adaptation rules on CEC'2017 in 10 dimensions.

each method. On multimodal functions, the PPMF achieved better performance than other variants. The same effect can be observed for problems in 30 dimensions presented in Fig. 7, i.e., CSA is superior only on unimodal functions. In 50 dimensions shown in Fig. 8, the performance gap on basic multimodal and composition functions between PPMF and other methods increases in favor of PPMF. The method with

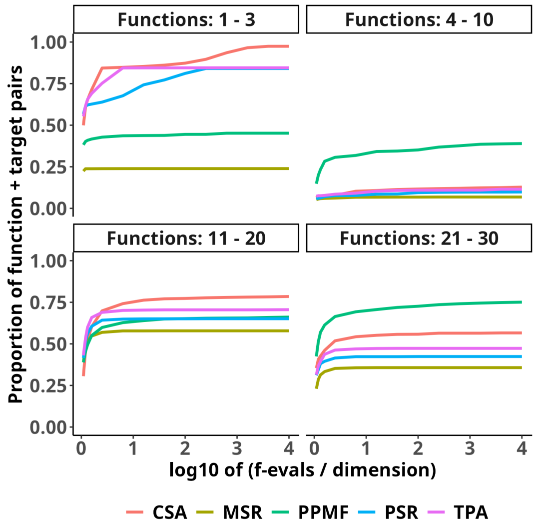

Figure 7: Results for MA-ES coupled with different step-size adaptation rules on CEC'2017 in 30 dimensions.

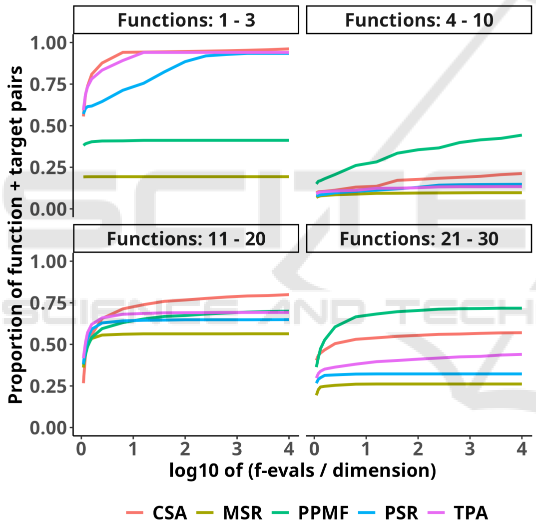

Figure 8: Results for MA-ES coupled with different step-size adaptation rules on CEC'2017 in 50 dimensions.

the worst performance for each dimension is MSR, which may indicate sensitivity to the parameters setup or recommended values not being generic enough to combine well with MA-ES. The TPA and PSR variants reveal almost identical performance, which was slightly worse than CSA.

5 CLOSING REMARKS

We demonstrate that the MA-ES algorithm can be accompanied with different cumulative step-size adaptation techniques that involve different mechanisms than CSA. For specific optimization problems, MA-ES coupled with the alternative methods performs more efficiently than CSA. Among the analyzed step-size adaptation methods, PPMF is quite competitive, particularly in higher dimensions and when the optimization problem is highly multimodal.

REFERENCES

Ait Elhara, O., Auger, A., and Hansen, N. (2013). A median success rule for non-elitist evolution strategies: study of feasibility. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, page 415–422, New York, NY, USA. Association for Computing Machinery.

Arabas, J. and Biedrzycki, R. (2017). Improving evolutionary algorithms in a continuous domain by monitoring the population midpoint. *IEEE Transactions on Evolutionary Computation*, 21(5):807–812.

Arnold, D. V. and Hansen, N. (2010). Active covariance matrix adaptation for the (1+1)-cma-es. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, page 385–392, New York, NY, USA. Association for Computing Machinery.

Awad, N. H., Ali, M., Liang, J., Qu, B., and Suganthan, P. N. (2016). Problem definitions and evaluation criteria for the CEC 2017 special session and competition on real-parameter optimization. Technical report, Nanyang Technol. Univ., Singapore and Jordan Univ. Sci. Technol. and Zhengzhou Univ., China.

Beyer, H.-G. and Arnold, D. (2003). Qualms regarding the optimality of cumulative path length control in csa/cma-evolution strategies. *Evolutionary computation*, 11:19–28.

Beyer, H.-G. and Sendhoff, B. (2017). Simplify your covariance matrix adaptation evolution strategy. *IEEE Transactions on Evolutionary Computation*, 21(5):746–759.

Glasmachers, T., Schaul, T., Yi, S., Wierstra, D., and Schmidhuber, J. (2010). Exponential natural evolution strategies. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, page 393–400, New York, NY, USA. Association for Computing Machinery.

Hansen, N. (2008). Cma-es with two-point step-size adaptation.

Hansen, N. (2018). A practical guide to experimentation. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '18, page 432–447, New York, NY, USA. Association for Computing Machinery.

Hansen, N. (2023). The cma evolution strategy: A tutorial.

Hansen, N., Atamna, A., and Auger, A. (2014). How to assess step-size adaptation mechanisms in randomised search. In Bartz-Beielstein, T., Branke, J., Filipič, B., and Smith, J., editors, *Parallel Problem Solving from*

Nature – PPSN XIII, pages 60–69, Cham. Springer International Publishing.

Hansen, N. and Auger, A. (2014). *Principled Design of Continuous Stochastic Search: From Theory to Practice*, pages 145–180. Springer Berlin Heidelberg, Berlin, Heidelberg.

Hansen, N., Müller, S. D., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary Computation*, 11(1):1–18.

Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.*, 9(2):159–195.

Krause, O., Glasmachers, T., and Igel, C. (2017). Qualitative and quantitative assessment of step size adaptation rules. In *Proceedings of the 14th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*, FOGA '17, page 139–148, New York, NY, USA. Association for Computing Machinery.

Loshchilov, I. (2015). Lm-cma: An alternative to l-bfgs for large-scale black box optimization. *Evolutionary computation*, 25.

Rechenberg, I. (1994). Evolutionsstrategie. *Arbeitstagung der Freien Akademie Berlin,*.

Salomon, R. (1998). Evolutionary algorithms and gradient search: similarities and differences. *IEEE Transactions on Evolutionary Computation*, 2(2):45–55.

Warchulski, E. and Arabas, J. (2021). A new step-size adaptation rule for cma-es based on the population midpoint fitness. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 825–831.