Universiteit
Leiden

Master Computer Science

An Adaptive Re-evaluation Method for Evolution Strategy under Additive Noise

Name:	Catalin-Viorel Dinu
Student ID:	s3697037
Date:	21/06/2024
Specialisation:	Artificial Intelligence
1st supervisor:	Dr. Hao Wang
2nd supervisor:	Prof. Dr. Vedran Dunjko

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

AN ADAPTIVE RE-EVALUATION METHOD FOR EVOLUTION STRATEGY UNDER ADDITIVE NOISE

June 18, 2024

ABSTRACT

In the field of black-box optimization algorithms the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) is one of the most advanced optimization method. While mostly tested under ideal conditions, CMA-ES has shown promising robustness against realistic cost function evaluation which include noise. Previous works have proposed techniques to mitigate noise by population size adaptation. In contrast, we propose the use of a re-evaluation scheme to better approximate the true value. In particular, we study CMA-ES optimization under additive noise, and design an Adaptive Re-evaluation scheme to guarantee a better solution by re-evaluating existing candidates. Our method is based on maximizing a lower-bound of the expected improvement of the noiseless objective value given information about the noise level, the Lipschtz constant, and the hyper-parameters of CMA-ES. Furthermore, our AR-CMA-ES is capable of finding the analytical solution to the number of re-evaluations at a very small computational cost. We experimentally investigate the performance of the proposed re-evaluation method on a set of artificial test functions across various noise levels, optimization budgets, and the dimension of the search space. Compared to existing approaches to handle additive noise for CMA-ES, our method demonstrates significant advantages in terms of the precision to the optimum and the percentage of independent runs achieving a certain precision.

1 Introduction

Optimization problems are central in various scientific and engineering fields. Typically, these problems are analyzed under ideal conditions with the assumption of a noiseless environment. However, many real-world optimization problems involve noise, which can alter the true objective function, making the optimisation space less reliable. Various sources, such as measurement errors, environmental variability, or the inherent randomness of the system ([RKD17]) can cause noise to appear in the objective function. Therefore, several noise models were proposed in the literature, but we can broadly put them into two categories:

- **Additive Noise** ([R$^+$21],[DL15]): This model assumes that the noise added to the objective function value is independent of the function value itself. It can be expressed as $\tilde{f}(\vec{x}) = f(\vec{x}) + \sigma_\text{n}\mathcal{N}(0,1)$, where σ_n is the standard deviation of the noise.
- **Multiplicative Noise** ([UNS24]): In this model, the noise scales with the objective function value. It can be expressed as $\tilde{f}(\vec{x}) = f(\vec{x})(1 + \sigma_\text{n}\mathcal{N}(0,1))$ where σ_n is the standard deviation of the noise.

In the context of noisy optimisation, promising results show evolutionary algorithms (EAs), as their population-based approach provides robustness against noise ([Arn02, HNGK08, RKD17]). The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is recognized as a robust algorithm for black-box optimization, excelling in noiseless scenarios ([VAB$^+$18]). This algorithm stands as the backbone of several noise-resilient methods that fall into one of three general categories: population size adaptation, learning rate (step-size) adaptation or function re-evaluation for estimating the true function.

Population size adaptation ([LZC$^+$22],[HCPGM99],[NP98]) refers to dynamically modifying the population size such that the noise impact can be more evenly distributed via EA. Larger populations provide more candidates, which increases the likelihood that the selection mechanism will choose candidates whose fitnesses are near to the genuine value. This can result in a more accurate solution. Though successful, this approach requires more and more computational resources. Learning rate adaptation ([NAO23]) refers to adjusting the learning rate or step size in accordance with the noise level. Because they indicate a steady and progressive gain to the optimal value, ever-smaller steps can lower the sensitivity to noise. The most common method for dealing with noise involves re-evaluating the objective function ([AW93], [BM15], [HNGK08],[AW94]) for each candidate several times and averaging the results to obtain a more precise estimate of the underlying function. It helps by smoothing out any fluctuations caused by noise, providing a clearer direction for the searching process. This approach, while effective, is computationally expensive and inefficient unless it is properly managed.

We examine the additive Gaussian noise model in this work. We propose an alternative strategy based on adaptive sampling, where the optimal number of re-evaluations per point can be efficiently determined to minimize the impact of noisy evaluations. The primary objective of this study is to enhance the performance of CMA-ES in noisy optimization environments by introducing an **Adaptive Re-evaluation method** (AR-CMA-ES). We assessed AR-CMA-ES's performance using the benchmark functions at varying levels of additive Gaussian noise. According to experimental findings, the AR-CMA-ES works better than the comparison techniques at various additive noise levels, obtaining higher precision and a higher percentage of individual runs that meet a certain precision threshold.

Therefore, we list our contributions as follows:

- **Change the Center of Mass Updating Rule**: We propose a modification to the center of mass updating rule in CMA-ES to better handle noisy evaluations.
- **Optimize Re-evaluation Efficiency**: Our adaptive method is designed to optimize the efficiency of function re-evaluations, thereby enhancing the algorithm's effectiveness in the presence of additive noise.
- **Analytical Solution for the Number of Re-evaluations**: We provide an analytical solution for determining the number of re-evaluations needed for each candidate solution, which requires minimal additional computational resources.

2 Background

2.1 Problem description

We consider minimizing a single-objective, black-box, and differentiable function $\mathcal{L}\colon \mathbb{R}^d \to \mathbb{R}$. In this work, we only consider the additive Gaussian noise, as for the other types of noise similar analytical steps can be consider. We shall denote as the noisy function value by:

$$\tilde{\mathcal{L}}(\vec{x}) = \mathcal{L}(\vec{x}) + \sigma_{\mathrm{n}}\mathcal{N}(0,1) \tag{1}$$

We assume the gradient of $\mathcal{L}$ is Lipschitz continuous, i.e., $\|\nabla\mathcal{L}(\vec{x}) - \nabla\mathcal{L}(\vec{x}')\| \leq K \|\vec{x} - \vec{x}'\|$ for some $K < \infty$ and all $\vec{x}, \vec{x}' \in \mathbb{R}^d$.

The re-evaluation method estimates $\mathcal{L}(\vec{x})$ via the sample mean and is often employed for noise mitigation. Based on the Central Limit Theorem (C.L.T.), we have:

$$\sqrt{M}(\mathcal{L}(\vec{x}) - \widehat{\mathcal{L}(\vec{x})}) \xrightarrow{d} \sigma_{\mathrm{n}}\mathcal{N}(0,1) \tag{2}$$

where $\widehat{\mathcal{L}(\vec{x})} = M^{-1}\sum_{i=1}^{M} y_i$ is computed by sampling M independent and identically distributed (i.i.d.) samples y_i drawn from $\tilde{\mathcal{L}}(\vec{x})$.

It is important to determine the value of M properly: ideally, the task is to allocate a sufficiently large M to re-evaluate each point in a set $\{\vec{x}^i\}_i$ such that each pair of $\widehat{\mathcal{L}(\vec{x}^i)}$ and $\widehat{\mathcal{L}(\vec{x}^j)}$ can be separated with high probability, i.e., $\sigma_{\mathrm{n}}/\sqrt{M} \in o(|\mathcal{L}(\vec{x}^i) - \mathcal{L}(\vec{x}^j)|)$ which implies that noise's standard deviation, reduced by a factor of $\sqrt{M}$, is smaller than the difference between the true function values $\mathcal{L}(\vec{x}^{\,i})$ and $\mathcal{L}(\vec{x}^{\,i})$.

Considering $\{\vec{x}^i\}_i$ is contained within a compact subset of $\mathbb{R}^d$, we have the following two scenarios:

- when $\mathcal{L}$ exhibits a large Lipschitz constant locally, then we do not need to use a large value for M since $|\mathcal{L}(\vec{x}^i) - \mathcal{L}(\vec{x}^j)|$ is large;
- when the Lipschitz constant is small, the difference $|\mathcal{L}(\vec{x}^i) - \mathcal{L}(\vec{x}^j)|$ is small; requiring a much larger M to ensure that the noise does not obscure these differences.

The determination of M is a non-trivial task since we generally have no knowledge of the local Lipschitz constant of the black-box function, and the $\{\vec{x}^i\}_i$ is typically generated randomly in optimization algorithms.

2.2 CMA-ES

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [Han16] is a widely-applied variable-metric method for continuous single-objective black-box problems. Briefly, CMA-ES's search process maintains the so-called center of mass $\vec{m} \in \mathbb{R}^d$ to estimate the global minimum that is either sampled uniformly at random (u.a.r.) from the search domain or provided by the user. Afterwards, in each iteration, CMA-ES samples independent and identically distributed (i.i.d.) solutions $\{\vec{x}^{\,i}\}_{i=1}^{\lambda}$ from a multivariate Gaussian $\mathcal{N}(0, \sigma^2\mathbf{C})$:

$$\vec{x}^{\,i} = \vec{m} + \vec{z}^{\,i}, \quad \vec{z}^{\,i} \sim \sigma\mathbf{C}^{1/2}\mathcal{N}(0, \mathbf{I}), \quad i \in [1..\lambda],$$

where σ is the step size that controls the length of search steps, serving the same role as the learning rate in stochastic gradient descent methods. Both the step-size σ and the covariance matrix $\mathbf{C}$ are self-adapted in CMA-ES (see [Han16] for details). After evaluating those solutions on the objective function $\mathcal{L}$, CMA-ES ranks the mutations according to their objective values (ties are broken randomly), i.e., $\mathcal{L}(\vec{x}^{\,1:\lambda}) < \mathcal{L}(\vec{x}^{\,2:\lambda}) < \ldots < \mathcal{L}(\vec{x}^{\,\lambda:\lambda})$. Only the smallest $\mu < \lambda$ are selected to update the center of mass:

$$\vec{m} \leftarrow \vec{m} + \sum_{i=1}^{\mu} w_i \vec{z}^{\,i:\lambda}, \quad w_i = \log\lambda - \log i. \tag{3}$$

where $\vec{z}^{\,i:\lambda}$ represent the i-th mutation vector by value function. By default, CMA-ES uses monotonically decreasing weights with respect to the ranking of solutions. Subsequently, the internal parameters σ and $\mathbf{C}$ are also updated based on the selected μ solutions (see [Han16] for details).

2.2.1 3 Stage CMA-ES

A basic approach to managing noise in optimization involves fixing the number of re-evaluations (M) per candidate solution. However, this static method often shows limited performance due to its inability to adapt to varying noise levels. To address these limitations, the Three-Stage CMA-ES ([CMMS20],[BM15]) introduces a planned schedule that incrementally increases the number of function re-evaluations across three stages, enhancing robustness and efficiency.

In this method, function re-evaluations are proportionally allocated (10:3:1): 10 parts for the smallest M,3 parts for the middle M and 1 part for the highest M. For instance, with a budget of 1e7 function re-evaluations, M values might be set to $[100, 1000, 10000]$. This results in approximately 7,150 evaluations for M=100, 2,145 for M=1,000, and 715 for M=10000. Similarly, with a budget of 1e8 function re-evaluations could be 1000, 10000 and 100000, and for a budget of 1e9 it could be set to 10000,10000 and 1000000.

However, this planned behaviour can be problematic. The increase in function re-evaluation may occur either too early, leading to inefficient use of the allocated budget, or too late, potentially affecting the inner workings of CMA-ES, as noise can significantly impact the adaptation of step size and covariance matrix.

2.2.2 Uncertainty handling CMA-ES

Uncertainty handling CMA-ES (UH-CMA-ES)[HNGK09] proposes a method to measure uncertainty by recomputing the function value of some elements from the population and increasing the number of re-evaluations based on the changes that appeared in the ranking scheme.

Therefore, UH-CMA-ES implies choosing a sub-population of size $\lambda_{\text{re-es}}$ that is re-estimated. As the mutation vectors are independently sampled, first $\lambda_{\text{re-es}}$ can be chosen. Using the re-estimated values, two lists are composed as follows:

$$L_1 = \Big((\vec{x}^1, \tilde{\mathcal{L}}^1), ..., (\vec{x}^\lambda, \tilde{\mathcal{L}}^\lambda), (\vec{x}^1, \tilde{\mathcal{L}}^1), ..., (\vec{x}^\lambda, \tilde{\mathcal{L}}^\lambda)\Big)$$
$$L_2 = \Big((\vec{x}^1, \tilde{\mathcal{L}}^1), ..., (\vec{x}^\lambda, \tilde{\mathcal{L}}^\lambda), (\vec{x}^1, \tilde{\mathcal{L}}^1_{\text{re-es}}), ..., (\vec{x}^\lambda, \tilde{\mathcal{L}}^\lambda_{\text{re-es}})\Big)$$

where $\tilde{\mathcal{L}}^i_{\text{re-es}} = \begin{cases} \text{the re-estimated value, if } i \leq \lambda_{\text{re-es}} \\ \tilde{\mathcal{L}}^i\text{, otherwise} \end{cases}$. The method sorts the two lists and computes the difference in ranking between the initial values and the re-estimated one:

$$\Delta_i = |rank_{L_1}(\vec{x}^i) - rank^{\text{re-es}}_{L_2}(\vec{x}^i)|$$

The uncertainty level is defined as:

$$s = \frac{1}{\lambda_{\text{re-es}}} \sum_{i=1}^{\lambda_{\text{re-es}}} \Big(2\Delta_i - \Delta_\theta^{\lim}(rank_{L_2}^{\text{re-es}}(\vec{x}^i) - \mathbf{1}_{\tilde{\mathcal{L}}^i_{\text{re-es}} > \tilde{\mathcal{L}}^i}) - \Delta_\theta^{\lim}(rank_{L_1}(\vec{x}^i) - \mathbf{1}_{\tilde{\mathcal{L}}^i_{\text{re-es}} < \tilde{\mathcal{L}}^i})) \Big)$$

where $\Delta_\theta^{\lim}(n)$ denotes the θ×50 percentile of the possible rank changes and it is introduced to limit the impact of minor ranking changes, which might be insignificant

If the computed uncertainty level s is positive, it indicates significant uncertainty in the current evaluations. In response, the number of re-evaluations M is adaptively increased:

$$M \leftarrow \alpha M$$

where α is a scaling factor greater than one.

However, this approach does not consider the form of the noise, potentially resulting in sub-optimal estimates of the number of re-evaluations (M). Moreover, it implies a computation overhead due to the necessity of re-estimating the function value of $\lambda_{\text{re-es}}$ candidates.

3 Adaptive Re-evaluation Method (AR-CMA-ES)

As previously discussed, our method serves as an extension of the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), with the primary objective of dynamically estimating the optimal number of function re-evaluations (denoted as M) required for each iteration. We begin our investigation by generalizing the update mechanism of the search position in the CMA-ES algorithm as:

$$\vec{z} = \sum_{i=1}^{\lambda} w_i \mathbf{C}^{1/2} \vec{\varepsilon}^i, \quad w_i = h\left(\Delta\tilde{\mathcal{L}}^i\right), \quad \vec{\varepsilon}^i \sim \sigma\mathcal{N}(0, \mathbf{I}), \tag{4}$$

where $\Delta\tilde{\mathcal{L}}^i = \tilde{\mathcal{L}}(\vec{m}) - \tilde{\mathcal{L}}(\vec{m} + \mathbf{C}^{1/2}\vec{\varepsilon}^i)$ is the change of the noisy cost function and $h : \mathbb{R} \to \mathbb{R}$ is a strictly increasing weight function, and $\mathbf{C}$ is the covariance matrix. We refer to $\vec{\varepsilon}^i$ as the i-th mutation vector.

Instead of using the default weight function (see Eq. (3)), we consider the proportional weight function for the ease of analysis, which assigns a positive weight in proportion to the loss value of each mutation:

$$h\left(\Delta\tilde{\mathcal{L}}^i\right) = \frac{\Delta\tilde{\mathcal{L}}^i + A}{\sum_{k=1}^{\lambda} \Delta\tilde{\mathcal{L}}^k + \lambda A}, \tag{5}$$

where A is chosen as the smallest possible value that ensures all weights remain positive with high probability.

Consider the first-order Taylor expansion:

$$\Delta\tilde{\mathcal{L}}^i = -\left\langle \nabla\mathcal{L}(\vec{m}), \mathbf{C}^{1/2}\vec{\varepsilon}^i \right\rangle + \mathcal{O}\left(\left\|\mathbf{C}^{1/2}\vec{\varepsilon}^i\right\|_2^2\right) + \delta^{\,i} \tag{6}$$

$$= -\langle \vec{g}, \vec{\varepsilon}^i \rangle + R\left\|\vec{\varepsilon}^i\right\|_2^2 + \delta^{\,i} \tag{7}$$

where $\delta^i \sim \mathcal{N}\left(0, \frac{\eta^i}{M}\right)$ is the evaluation noise which is independent of the mutation $\vec{\varepsilon}^i$, $\vec{g} = \mathbf{C}^{1/2}\mathcal{L}(\vec{m})$ and $R \in \mathbb{R}$. When the step-size σ is small, we have $\Delta\tilde{\mathcal{L}}^i \sim (\sigma\left\|\mathbf{C}^{1/2}\nabla\mathcal{L}(\vec{m})\right\|_2 + \sqrt{\eta^i/M})\mathcal{N}(0,1)$. Choosing $A \geq c\left(\sigma\left\|\mathbf{C}^{1/2}\nabla\mathcal{L}(\vec{m})\right\|_2 + \sqrt{\eta^i/M}\right)$ will ensure $\Pr(\Delta\tilde{\mathcal{L}}^i + A \leq 0) \leq 1 - \Phi(c)$ (e.g., $c = 3$ gives ca. 0.15% chance of realizing negative weights). Also, since A is a probabilistic upper bound of $\Delta\tilde{\mathcal{L}}^i$, we can relax the denominator of Eq. (5) to $2\lambda A$, which leads to a modified search direction:

$$\vec{z}\,' = \frac{1}{2\lambda A} \sum_{i=1}^{\lambda} (\Delta\tilde{\mathcal{L}}^i + A)\mathbf{C}^{1/2}\vec{\varepsilon}^i, \quad \vec{\varepsilon}^i \sim \sigma\mathcal{N}(0, \mathbf{I}). \tag{8}$$

Although the above expression is easier to analyze, we have to show that $\vec{z}\,'$ does not differ too much from $\vec{z}$:

$$\vec{z}\,' = \frac{\sum_{k=1}^{\lambda} \Delta\tilde{\mathcal{L}}^k + \lambda A}{2\lambda A}\vec{z} = \left(\frac{1}{2} + \frac{1}{2c}\mathcal{N}(0,1)\right)\vec{z}. \tag{9}$$

The probability that $\vec{z}'$ inverts $\vec{z}$ is $1 - \Phi(c)$ which is pretty tiny for $c \geq 3$. Hence, we can safely take Eq. (8) for the following analysis.

Our goal is to estimate the number of function re-evaluations (M) required by maximizing their efficiency. This approach strikes a balance by enhancing the expected improvement gained in each generation while minimizing the number of function re-evaluations.The *efficiency* (similar to [GLD$^+$21]) metric is defined as follows :

$$\gamma = \frac{\mathbb{E}\left[\mathcal{L}(\vec{m}) - \mathcal{L}(\vec{m} + \vec{z}')\right]}{M}. \tag{10}$$

Computing the exact form of the efficiency is hard, we can instead focus on finding a lower bound that can be easily maximized. This approach simplifies the task while still guiding the optimization process effectively. Firstly, we consider a change of basis of $\mathbb{R}^d$, i.e., $\forall i \in [1..d], \vec{e}_i' = \mathbf{C}^{1/2}\vec{e}_i$. Note that $\Delta\tilde{\mathcal{L}}$ is not affected by the change of basis. In the new basis, the update vector is:

$$\vec{v}' = \mathbf{C}^{-1/2}\vec{z}' = \frac{1}{2\lambda A}\sum_{i=1}^{\lambda} \underbrace{(\Delta\tilde{\mathcal{L}}^i + A)\vec{\varepsilon}^i}_{\vec{v}^{\,i}} \tag{11}$$

Using the first-order Taylor expansion, we proceed to calculate the first moment and the second non-central moment for each individual component of $\vec{v}^{\,i}$(see Appendix A.2). For $k \in [1..d]$, we have:

$$\mathbb{E}\left[v_k^i\right] = -g_k\sigma^2 \tag{12}$$

$$\mathbb{E}\left[(v_k^i)^2\right] = \frac{\eta_i\sigma^2}{M} + (\|\vec{g}\|_2^2 + 2g_k^2)\sigma^4 + A^2\sigma^2 \tag{13}$$

where v_k^i and g_k are the k-th component of $\vec{v}^{\,i}$ and $\vec{g}$, respectively.

Considering the Quadratic Upper Bound on the loss function (see Theorem A.1), we can compute a lower bound on the expected improvement gained by updating the centre of mass. This analysis is detailed in Appendix A.3. Consequently, we establish the following lower bound:

$$\begin{aligned}
\mathbb{E}\left[\tilde{\mathcal{L}}(\vec{m}) - \tilde{\mathcal{L}}(\vec{m} + \vec{z}')\right] &= \mathbb{E}\left[\tilde{\mathcal{L}}(\vec{m}) - \tilde{\mathcal{L}}(\vec{m} + \mathbf{C}^{1/2}\vec{v}')\right] && (14)\\
&\geq \mathbb{E}\left[-\langle\vec{g}, \vec{v}'\rangle - \frac{K}{2}\left\|\mathbf{C}^{1/2}\vec{v}'\right\|_2^2\right] && (15)\\
&- -\mathbb{E}\left[\langle\vec{g}, \vec{v}'\rangle\right] - \frac{K}{2}\mathbb{E}\left[\left\|\mathbf{C}^{1/2}\vec{v}'\right\|_2^2\right] && (16)\\
&\geq -\mathbb{E}\left[\langle\vec{g}, \vec{v}'\rangle\right] - \frac{K\lambda_d}{2}\mathbb{E}\left[\|\vec{v}'\|_2^2\right] && (17)\\
&= \frac{\sigma^2}{2A}\|\vec{g}\|_2^2 - \frac{\sigma^4(\lambda + d + 1)K\lambda_d}{8\lambda A^2}\|\vec{g}\|_2^2 - \frac{dK\lambda_d\sigma^2}{8\lambda} - \frac{1}{M}\frac{\sigma^2 dK\lambda_d}{8\lambda^2 A^2}\sum_{i=1}^{\lambda}\eta_i && (18)
\end{aligned}$$

where λ_d is the biggest eigenvalue of $\mathbf{C}$ and K is the Lipschitz constant of $\nabla\tilde{\mathcal{L}}^i$. Let us comment on the steps in the previous computation

- (14) to (15): follows the Quadratic Upper Bound (see Theorem A.1)
- (15) to (16): follows the Linearity of expectation
- (16) to (17): we have a consistent norm with $\left\|\mathbf{C}^{1/2}\vec{v}'\right\|_2 \leq \left\|\mathbf{C}^{1/2}\right\|_2 \|\vec{v}\|_2$, and $\left\|\mathbf{C}^{1/2}\right\|_2 = \sqrt{\lambda_d}$
- (17) to (18): step-by-step solution on the 2 elements can be found in Appendix A.3

Considering the lower bound on expected improvement (eq. (18)), we can establish a lower bound on efficiency:

$$\begin{aligned}
\gamma &\geq -\frac{1}{M^2}\frac{\sigma^2}{2A^2}\left(\frac{dK\lambda_d}{4\lambda^2}\sum_{i=1}^{\lambda}\eta_i\right) + \frac{1}{M}\frac{\sigma^2}{2A^2}\left(A\|\vec{g}\|_2^2 - \frac{\sigma^2(\lambda + d + 1)K\lambda_d}{4\lambda}\|\vec{g}\|_2^2 - \frac{A^2 dK\lambda_d}{4\lambda}\right) && (19)\\
&= -\frac{1}{M^2}\frac{\sigma^2}{2A^2}a + \frac{1}{M}\frac{\sigma^2}{2A^2}b && (20)
\end{aligned}$$

So we have a quadratic equation, where if **b** is positive, we have a finite M that maximises γ (we know the coefficient of $\frac{1}{M^2}$ is always positive):

$$M^* = \frac{2a}{b} \tag{21}$$

We employ an exponential smoothing technique on the estimated function re-evaluations to ensure stability:

$$M \leftarrow (1-\beta)M + \beta M^*, \quad \beta \in [0,1]$$

Estimating the evaluation noise Given that each individual evaluation noise follows approximately the same distribution as the theoretical noise $\frac{1}{\sqrt{M}}\mathcal{N}(0, \sigma_{\text{n}})$, we consider the following estimator:

$$\sum_{i=1}^{\lambda} \eta_i \approx \lambda \sigma_{\text{n}}^2 \tag{22}$$

Estimating the loss gradient We observe that eq. (12) indicates the mutation vectors are unbiased estimators of the loss gradient:

$$\vec{g} = -\frac{\mathbb{E}\left[\vec{v}^{\,i}\right]}{\sigma}. \tag{23}$$

We construct an estimator of this expected value by aggregating all mutation vectors, i.e.,

$$\vec{g}^* = -\frac{1}{\lambda\sigma}\sum_{i=1}^{\lambda} \vec{v}^{\,i}. \tag{24}$$

Furthermore, we can exponentially smooth such an estimation across iterations of CMA-ES to get a more robust value.

$$\vec{g} \leftarrow (1-\alpha)\vec{g} + \alpha\vec{g}^*, \quad \alpha \in [0,1] \tag{25}$$

4 Experiments

Experiments This study conducts a comprehensive comparative analysis of our method (AR-CMA-ES) with 2 other optimization algorithms: UH-CMA-ES and 3-Stage CMA-ES. The evaluation aims to assess the robustness and performance of these algorithms across a diverse set of functions, including uni-modal and multi-modal functions, as well as dimension-separable and non-separable functions (see Table 1). Additionally, the analysis considers varying levels of noise, represented by different variances of the noise term $\sigma_{\text{n}}^2 \in [1, 10, 100]$.

Table 1: Benchmark functions used in the experiments with their respective search space ($d \rightarrow$ the dimension of the search space, $lb \rightarrow$ lower bound and $ub \rightarrow$ upper bound)

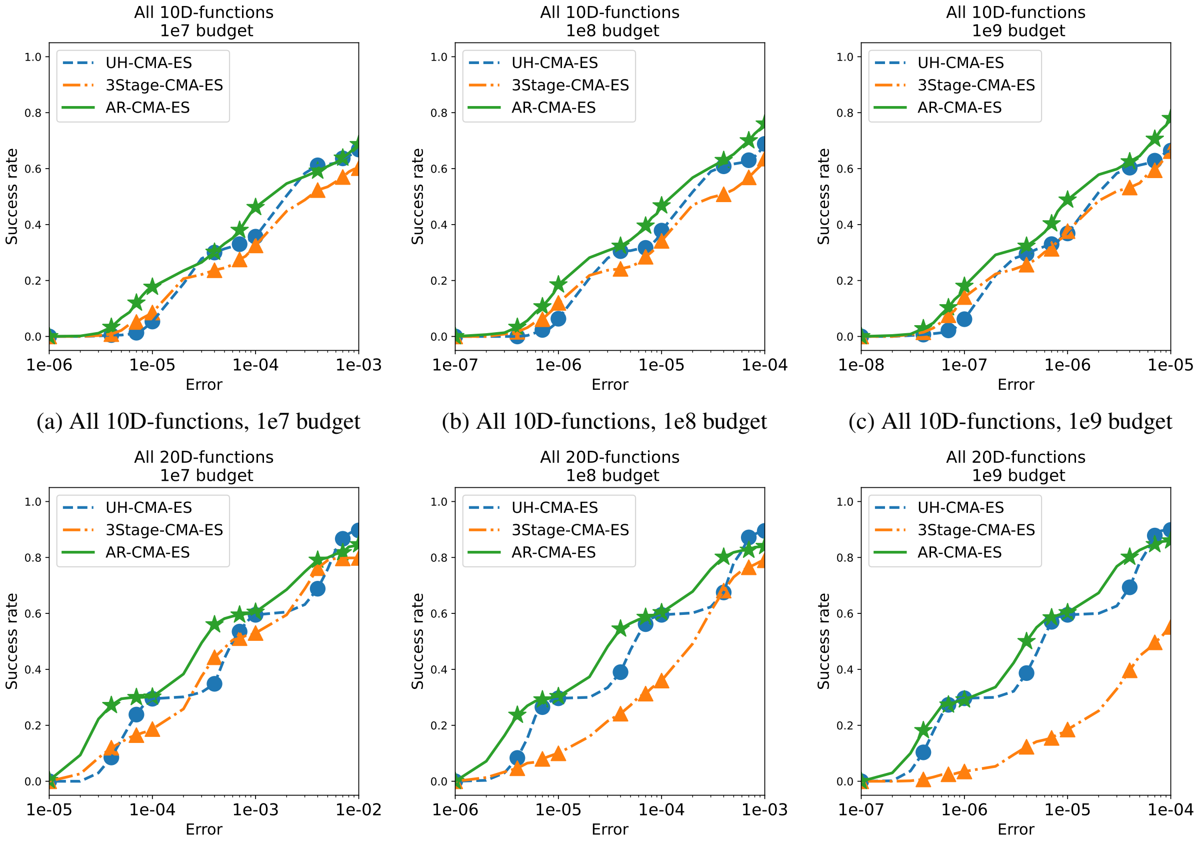
Name	$\mathcal{L}(\vec{x})$	Search Space
Sphere	$\sum_{i=1}^{d} x_i^2$	$[-5,5]^d$
Ellipsoid	$\sum_{i=1}^{d} 100^{\frac{i-1}{d-1}} x_i^2$	$[-5,5]^d$
Rotated Ellipsoid	$\sum_{i=1}^{d} 100^{\frac{d-i}{d-1}} x_i^2$	$[-5,5]^d$
Hyper-Ellipsoid	$\sum_{i=1}^{d} i x_i^2$	$[-5,5]^d$
Rotated Hyper-Ellipsoid	$\sum_{i=1}^{d} (d-i+1) x_i^2$	$[-5,5]^d$
Rastrigin	$10d + \sum_{i=1}^{d}\left[x_i^2 - 10\cos(2\pi x_i)\right]$	$[-5,5]^d$
Trid	$\sum_{i=1}^{d}(x_i-1)^2 - \sum_{i=2}^{d} x_i x_{i-1}$	$[-d^2, d^2]^d$
Cosine Mixture	$-0.1\sum_{i=1}^{d}\cos(5\pi x_i) + \sum_{i=1}^{d} x_i^2$	$[-1,1]^d$
Bohachevsky	$\sum_{i=1}^{d-1}\left[x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7\right]$	$[-15,15]^d$
Schwefel02	$\sum_{i=1}^{d}\left(\sum_{j=1}^{i} x_i\right)^2$	$[-10,10]^d$

To establish a fair comparison, all algorithms are initialized with consistent starting parameters: $\lambda = 100$, $\mu = 50$ and $\sigma_0 = 0.1 * (ub - lb)$. Our method differs from the others by incorporating two specific parameters: the exponential smoothing factors. We extensively tested various combinations and determined that setting α and β to 0.1 yielded the most consistent and reliable results. Also, different from the analytical results, we choose the probabilistic upper bound as the smallest improvement observed ($A \leftarrow \min_{i=1}^{\lambda} \Delta\tilde{\mathcal{L}}^i$). Furthermore, each function's global Lipschitz gradient constant (K) was computed analytically to ensure precise evaluation. We tested our algorithm across different budgets of function evaluations, with a limit set on the number of function re-evaluations per individual in a generation at 1% of the total evaluation budget.

This study aims to provide valuable insights into our algorithm's strengths and weaknesses by analyzing it across different function types and noise levels. Thus, it will aid in selecting the most suitable approach for various optimization tasks.

Results Figure 1 presents the empirical cumulative distribution function (ECDF) for different dimensions (10, 20) and budgets (1e7, 1e8, 1e9). We computed the mean ECDF across all functions and noise levels. As we increase the budget and the function dimension, and hence the complexity, AR-CMA-ES shows a substantial performance improvement compared to 3-Stage CMA-ES. With more complex problems, 3-Stage CMA-ES lacks the ability to adapt effectively, as there is no assurance that the increased number of function re-evaluations is necessary. Moreover, we observe a slight but consistent improvement with AR-CMA-ES compared to UH-CMA-ES. For smaller and smaller errors to the optimum, AR-CMA-ES achieves a higher percentage of solutions that meet those thresholds compared to UH-CMA-ES, indicating better performance in reaching closer to the optimal solutions.

In Appendix B, Figure 6 and Figure 7 also present the mean ECDF across different dimensions (10, 20), budgets (1e7, 1e8, 1e9), and noise levels (1, 10, 100). As the noise level increases, there is a slight decrease in performance. This behavior is due to the overestimation, as the number of function re-evaluations is linearly dependent on the noise.

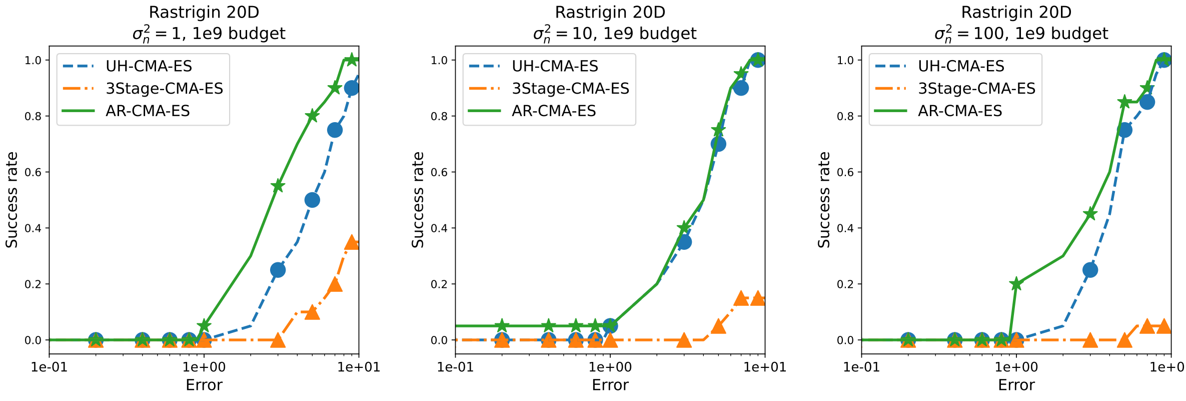


(a) All 10D-functions, 1e7 budget
(b) All 10D-functions, 1e8 budget
(c) All 10D-functions, 1e9 budget
(d) All 20D-functions, 1e7 budget
(e) All 20D-functions, 1e8 budget
(f) All 20D-functions, 1e9 budget

Figure 1: The combined Empirical Cumulative Distribution (ECDF) of all benchmark functions (each computed from 20 independent runs) across dimensions 10D and 20D, and varying budgets (1e7, 1e8, 1e9).

For closer analysis, we consider two specific functions: the Rastrigin function and the Trid function. The Rastrigin function is highly multimodal, presenting numerous local optima, while the Trid function is a non-separable function across dimensions, making it a challenging problem for optimization algorithms.

Figure 2 shows the empirical cumulative distribution function (ECDF) for the 20D Rastrigin function with a budget of 1e9 function evaluations and different noise levels ($\sigma_{\text{n}}^2 \in [1, 10, 100]$). The Rastrigin function is highly multimodal, so optimization can easily get stuck in local minima. However, we observe a significant improvement with AR-CMA-ES compared to 3-Stage CMA-ES and a slight improvement over UH-CMA-ES. From Figure 3 showing the convergence plot, it is clear that the improvement gained by our method is due to the higher number of functions that succeed in avoiding local minima.

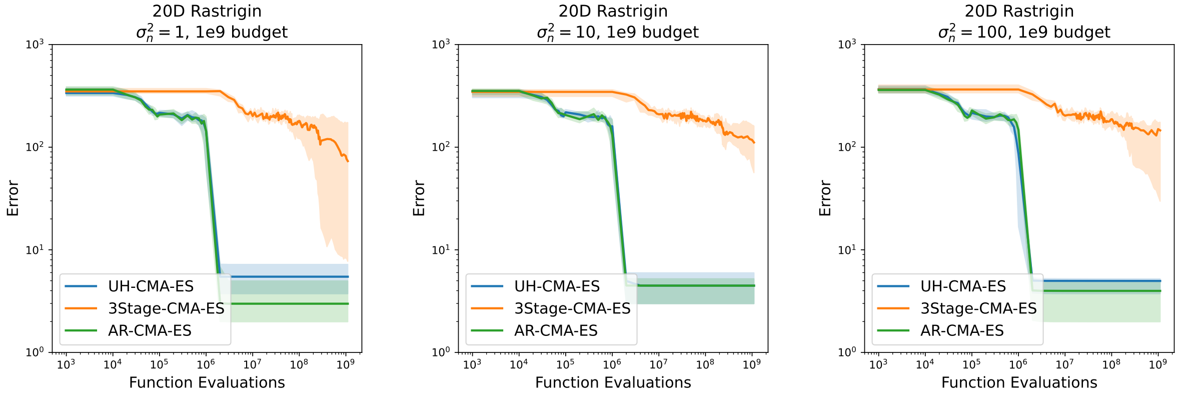


(a) 20D Rastrigin - $\sigma_{\text{n}}^2 = 1$

(b) 20D Rastrigin - $\sigma_{\text{n}}^2 = 10$

(c) 20D Rastrigin - $\sigma_{\text{n}}^2 = 100$

Figure 2: Empirical Cumulative Distribution Function (ECDF) for the 20D Rastrigin function with a budget of 1e9 function evaluations under different noise levels ($\sigma_{\text{n}}^2 \in [1, 10, 100]$).



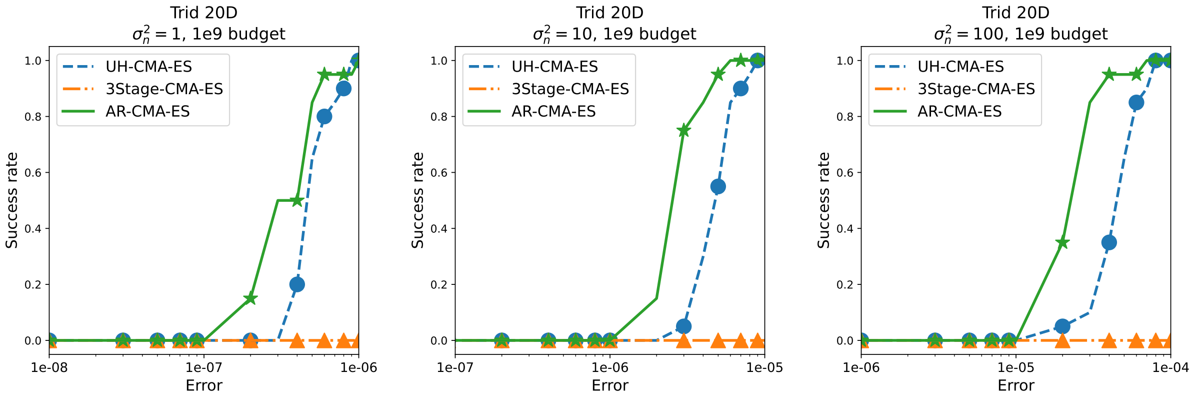
(a) 20D Rastrigin - $\sigma_{\text{n}}^2 = 1$

(b) 20D Rastrigin - $\sigma_{\text{n}}^2 = 10$

(c) 20D Rastrigin - $\sigma_{\text{n}}^2 = 100$

Figure 3: Convergence plot for the 20D Rastrigin function with a budget of 1e9 function evaluations under different noise levels ($\sigma_{\text{n}}^2 \in [1, 10, 100]$).

Figure 4 shows the empirical cumulative distribution function (ECDF) for the 20D Trid function with a budget of 1e9 function evaluations and different noise levels ($\sigma_{\text{n}}^2 \in [1, 10, 100]$). As discussed, the Trid function is non-separable across dimensions (the minimum cannot be found by optimizing each dimension separately). We observe that AR-CMA-ES achieves substantial improvement compared to 3-Stage CMA-ES and a noticeable improvement over UH-CMA-ES. s shown in Figure 3, 3-Stage CMA-ES did not achieve certain error thresholds, and the comparison includes only the convergence curves of the two adaptive methods. We can see that our method outperforms UH-CMA-ES across all noise levels, and even shows a slight increase in performance as the noise level increases.

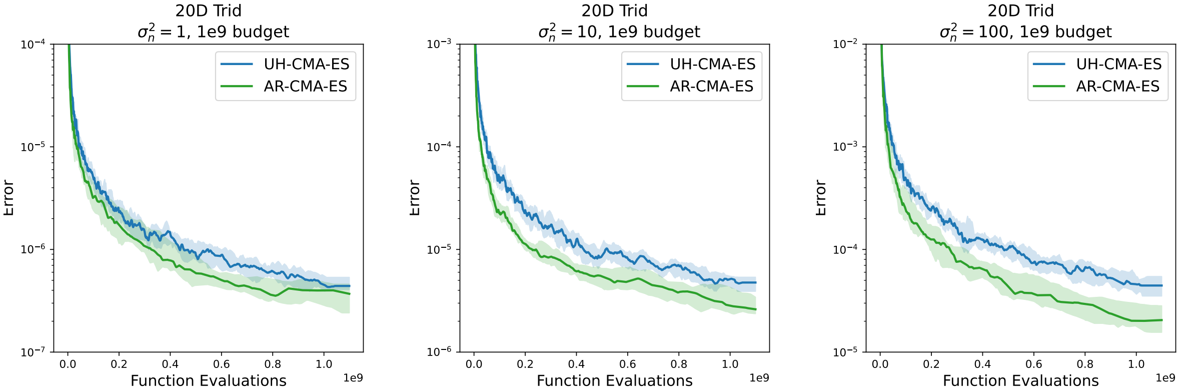


(a) 20D Trid - $\sigma_{\rm n}^2 = 1$

(b) 20D Trid - $\sigma_{\rm n}^2 = 10$

(c) 20D Trid - $\sigma_{\rm n}^2 = 100$

Figure 4: Empirical Cumulative Distribution Function (ECDF) for the 20D Trid function with a budget of 1e9 function evaluations under different noise levels ($\sigma_{\rm n}^2 \in [1, 10, 100]$).



(a) 20D Trid - $\sigma_{\rm n}^2 = 1$

(b) 20D Trid - $\sigma_{\rm n}^2 = 10$

(c) 20D Trid - $\sigma_{\rm n}^2 = 100$

Figure 5: Convergence plot for the 20D Trid function with a budget of 1e9 function evaluations (FE) under different noise levels ($\sigma_{\rm n}^2 \in [1, 10, 100]$).

5 Conclusion

In this paper, we introduced AR-CMA-ES, an enhancement of the CMA-ES algorithm designed to improve robustness against additive noise in optimization tasks. Our method incorporates a novel updating rule for the center of mass, which adapts based on actual performance improvements.

We derived a lower bound on the efficiency of function re-evaluations and developed an analytical solution for determining the optimal number of function re-evaluations. This adaptive strategy optimizes performance and ensures efficient resource allocation without significant computational overhead, leading to better optimization in noisy environments. AR-CMA-ES substantially outperforms 3Stage-CMA-ES and demonstrates a slight but consistent advantage over UH-CMA-ES. While AR-CMA-ES demonstrates significant improvements in handling additive noise and optimizing performance, it has several limitations that warrant consideration.

- **Assumptions on Noise Characteristics**: AR-CMA-ES is designed with a focus on additive noise. If the noise characteristics deviate from this assumption, such as in cases with multiplicative noise or other forms of complex noise patterns, the algorithm's performance might degrade. Further research is needed to extend the method to handle a broader range of noise types effectively.
- **Impact of Noise Variance**: The number of function re-evaluations in AR-CMA-ES is linearly dependent on the noise variance. As the noise level increases, this dependency can lead to a reduction in performance. The

increased need for re-evaluations in high-noise environments can strain computational resources and affect overall optimization efficiency.

- **Potential Benefits of Local Estimators for Lipschitz Constants**: Function re-evaluation in AR-CMA-ES is linearly dependent on the Lipschitz constant of the gradient. Utilizing local estimators for Lipschitz constants could reduce the probability of overestimating the number of function re-evaluations in AR-CMA-ES. By providing more accurate estimates tailored to specific regions of the optimization landscape, local estimators may help mitigate the impact of high global Lipschitz constants on algorithm performance.
- **Limited Empirical Validation**: While AR-CMA-ES demonstrates performance improvements on synthetic benchmark functions, its effectiveness on real-world problems remains to be fully explored. The empirical validation primarily focuses on synthetic functions that adhere to the assumptions about the function and noise type. Further experimentation is needed to evaluate the method's performance on functions that naturally conform to these assumptions. Examples include quantum loss functions, which are prevalent in quantum computing optimization tasks. Extending the empirical validation to encompass a broader range of real-world problems will provide deeper insights into the method's applicability and effectiveness in practical scenarios.

Despite these limitations, AR-CMA-ES offers a promising approach for improving the robustness of evolutionary algorithms in noisy optimization environments. Future work will focus on addressing these challenges and further refining the method to enhance its applicability and performance across a broader range of optimization tasks.

References

[Arn02] Dirk V Arnold. *Noisy optimization with evolution strategies*, volume 8. Springer Science & Business Media, 2002.

[AW93] Akiko N Aizawa and Benjamin W Wah. Dynamic control of genetic algorithms in a noisy environment. In *Proceedings of the fifth international conference on genetic algorithms*, volume 2, page 1, 1993.

[AW94] Akiko N. Aizawa and Benjamin W. Wah. Scheduling of genetic algorithms in a noisy environment. *Evolutionary Computation*, 2(2):97–122, 1994.

[BM15] Xavier Bonet Monroig. Optimization of quantum algorithms for near-term quantum computers. *Phys. Rev. A*, 2:042303, 2015.

[CMMS20] Chris Cade, Lana Mineh, Ashley Montanaro, and Stasja Stanisic. Strategies for solving the fermi-hubbard model on near-term quantum computers. *Phys. Rev. B*, 102:235122, Dec 2020.

[DL15] Duc-Cuong Dang and Per Kristian Lehre. Efficient optimisation of noisy fitness functions with population-based evolutionary algorithms. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, pages 62–68, 2015.

[GLD$^+$21] Andi Gu, Angus Lowe, Pavel A Dub, Patrick J Coles, and Andrew Arrasmith. Adaptive shot allocation for fast convergence in variational quantum algorithms. *arXiv preprint arXiv:2108.10434*, 2021.

[Han16] Nikolaus Hansen. The CMA evolution strategy: A tutorial. *CoRR*, abs/1604.00772, 2016.

[HCPGM99] George Harik, Erick Cantú-Paz, David E Goldberg, and Brad L Miller. The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Evolutionary computation*, 7(3):231–253, 1999.

[HNGK08] Nikolaus Hansen, André SP Niederberger, Lino Guzzella, and Petros Koumoutsakos. A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion. *IEEE Transactions on Evolutionary Computation*, 13(1):180–197, 2008.

[HNGK09] Nikolaus Hansen, AndrÉ S. P. Niederberger, Lino Guzzella, and Petros Koumoutsakos. A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion. *IEEE Transactions on Evolutionary Computation*, 13(1):180–197, 2009.

[LZC$^+$22] Zhenhua Li, Shuo Zhang, Xinye Cai, Qingfu Zhang, Xiaomin Zhu, Zhun Fan, and Xiuyi Jia. Noisy optimization by evolution strategies with online population size learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(9):5816–5828, 2022.

[NAO23] Masahiro Nomura, Youhei Akimoto, and Isao Ono. Cma-es with learning rate adaptation: Can cma-es with default population size solve multimodal and noisy problems? In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 839–847, 2023.

[NP98] V. Nissen and J. Propach. On the robustness of population-based versus point-based optimization in the presence of noise. *IEEE Transactions on Evolutionary Computation*, 2(3):107–119, 1998.

[Pol63] Boris T Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.

[R+21] Jonathan E Rowe et al. Evolutionary algorithms for solving unconstrained, constrained and multi-objective noisy combinatorial optimisation problems. *arXiv preprint arXiv:2110.02288*, 2021.

[RKD17] Pratyusha Rakshit, Amit Konar, and Swagatam Das. Noisy evolutionary optimization algorithms–a comprehensive survey. *Swarm and Evolutionary Computation*, 33:18–45, 2017.

[UNS24] Kento Uchida, Kenta Nishihara, and Shinichi Shirakawa. Cma-es with adaptive reevaluation for multiplicative noise. *arXiv preprint arXiv:2405.11471*, 2024.

[VAB+18] Konstantinos Varelas, Anne Auger, Dimo Brockhoff, Nikolaus Hansen, Ouassim Ait ElHara, Yann Semet, Rami Kassab, and Frédéric Barbaresco. A comparative study of large-scale variants of cma-es. In *Parallel Problem Solving from Nature–PPSN XV: 15th International Conference, Coimbra, Portugal, September 8–12, 2018, Proceedings, Part I 15*, pages 3–15. Springer, 2018.

A Appendix: Theorems and step-by-step solutions

A.1 Quadratic Upper Bound

Theorem Considering a differentiable function $f(x)$ with a Lipschitz continuous gradient ∇f, we have the following upper bound on f

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{K}{2} \|y - x\|_2^2, \quad \forall x, y \tag{26}$$

where K is the Lipschitz constant of the gradient ∇f (proof on [Pol63]).

The functions discussed in this paper adhere to the properties required by the previous theorem. Therefore we have:

$$\tilde{\mathcal{L}}(\vec{m} + \mathbf{C}^{1/2}\vec{v}\,') \leq \tilde{\mathcal{L}}(\vec{m}) + \langle \nabla \mathcal{L}(\vec{m}), \mathbf{C}^{1/2}\vec{v}\,' \rangle + \frac{K}{2} \left\| \mathbf{C}^{1/2}\vec{v}\,' \right\|_2^2 = \tag{27}$$

$$\tilde{\mathcal{L}}(\vec{m}) - \tilde{\mathcal{L}}(\vec{m} + \mathbf{C}^{1/2}\vec{v}\,') \geq -\langle \vec{g}, \vec{v}\,' \rangle - \frac{K}{2} \left\| \mathbf{C}^{1/2}\vec{v}\,' \right\|_2^2 \tag{28}$$

where $\vec{g} = \mathbf{C}^{1/2}\nabla\mathcal{L}(\vec{m})$.

A.2 Individual components of $\vec{v}^{\,i}$

We have $\vec{v}^{\,i} = \Delta\tilde{\mathcal{L}}^i\vec{\varepsilon}^i$, we can define for each $k \in [1..d]$ the indicvidual component as:

$$v_k^i = (\Delta\tilde{\mathcal{L}}^i + A)\varepsilon_k^i = \tag{29}$$

$$= \Big[-\left\langle \vec{g}, \vec{\varepsilon}^i \right\rangle + \delta^i - R \left\| \vec{\varepsilon}^i \right\|_2^2 + A \Big] \varepsilon_k^i \tag{30}$$

where $\varepsilon_k^i \sim \sigma\mathcal{N}(0,1)$, $\delta_i \sim \mathcal{N}(0, \frac{\eta_i}{M})$ and δ_i is independent of ε_k^i.

We have the first moment of each individual component:

$$\mathbb{E}\left[v_k^i\right] = \mathbb{E}\left[-\langle \vec{g}, \vec{\varepsilon}^i \rangle \varepsilon_k^i + \delta^i\varepsilon_k^i - R\left\|\vec{\varepsilon}^i\right\|_2^2 \varepsilon_k^i + A\varepsilon_k^i \right] = \tag{31}$$

$$= -\underbrace{\mathbb{E}\left[\langle \vec{g}, \vec{\varepsilon}^i \rangle \varepsilon_k^i\right]}_{A_1} + \underbrace{\mathbb{E}\left[\delta^i\varepsilon_k^i\right]}_{A_2} - R\underbrace{\mathbb{E}\left[\left\|\vec{\varepsilon}^i\right\|_2^2 \varepsilon_k^i\right]}_{A_3} + A\underbrace{\mathbb{E}\left[\varepsilon_k^i\right]}_{A_4=0} \tag{32}$$

We will compute each element (A_1, A_2, A_3) separately:

$$A_1 = \mathbb{E}\left[\langle \vec{g}, \vec{\varepsilon}^i \rangle \varepsilon_k^i\right] \overset{\text{(a)}}{=} \mathbb{E}\left[\sum_{j=1}^{d} g_j \varepsilon_j^i \varepsilon_k^i\right] \overset{\text{(b)}}{=} \sum_{j=1}^{d} g_j \mathbb{E}\left[\varepsilon_j^i \varepsilon_k^i\right] = \tag{33}$$

$$\overset{(c)}{=} g_k \mathbb{E}\left[(\varepsilon_k^i)^2\right] + \sum_{j\neq k}^{d} g_j \mathbb{E}\left[\varepsilon_j^i\right]\mathbb{E}\left[\varepsilon_k^i\right] \overset{(d)}{=} g_k\sigma^2 \tag{34}$$

Some comments on the steps in the previous computation:

- (a): follows definition of inner product
- (b): follows the Linearity of expectation
- (c): ε_j^i and ε_k^i are independent $\forall k \neq j$
- (d): $\mathbb{E}\left[\varepsilon_k^i\right] = 0$ and $\mathbb{E}\left[(\varepsilon_k^i)^2\right] = \sigma^2, \forall k$

$$A_2 = \mathbb{E}\left[\delta^i \varepsilon_k^i\right] \overset{(a)}{=} \mathbb{E}\left[\delta^i\right]\mathbb{E}\left[\varepsilon_k^i\right] \overset{(b)}{=} 0 \tag{35}$$

Some comments on the steps in the previous computation:

- (a): δ^i and ε_k^i are independent $\forall k$
- (b): $\mathbb{E}\left[\delta^i\right] = 0$ and $\mathbb{E}\left[\varepsilon_k^i\right] = 0, \forall k$

$$A_3 = \mathbb{E}\left[\left\|\vec{\varepsilon}^i\right\|_2^2 \varepsilon_k^i\right] \overset{(a)}{=} \mathbb{E}\left[\sum_{j=1}^{d} (\varepsilon_j^i)^2 \varepsilon_k^i\right] \overset{(b)}{=} \sum_{j=1}^{d} \mathbb{E}\left[(\varepsilon_j^i)^2 \varepsilon_k^i\right] = \tag{36}$$

$$\overset{(c)}{=} \mathbb{E}\left[(\varepsilon_k^i)^3\right] + \sum_{j\neq i}^{d} \mathbb{E}\left[(\varepsilon_j^i)^2\right]\mathbb{E}\left[\varepsilon_k^i\right] \overset{(d)}{=} 0 \tag{37}$$

Some comments on the steps in the previous computation:

- (a): follows definition of norm
- (b): follows the Linearity of expectation
- (c): ε_j^i and ε_k^i are independent $\forall k \neq j$
- (d): $\mathbb{E}\left[\varepsilon_k^i\right] = 0$, $\mathbb{E}\left[(\varepsilon_k^i)^2\right] = \sigma^2$ and $\mathbb{E}\left[(\varepsilon_k^i)^3\right] = 0, \forall k$

So, considering equations (34),(35),(37), we have the following final form for **the first moment** of v_k^i

$$\mathbb{E}\left[v_k^i\right] = -g_k\sigma^2 \tag{38}$$

We also can compute the second non-central moment of each individual component:

$$\mathbb{E}\left[(v_k^i)^2\right] = \mathbb{E}\left[(\Delta\tilde{\mathcal{L}}^i + A)^2 (\varepsilon_k^i)^2\right] \tag{39}$$

$$= \mathbb{E}\left[\left(-\langle\vec{g}, \vec{\varepsilon}^i\rangle + \delta^i + R\left\|\vec{\varepsilon}^i\right\|_2^2 + A\right)^2 (\varepsilon_k^i)^2\right] = \tag{40}$$

$$= \underbrace{\mathbb{E}\left[\langle\vec{g}, \vec{\varepsilon}^i\rangle^2 (\varepsilon_k^i)^2\right]}_{B_1} + \underbrace{\mathbb{E}\left[(\delta^i)^2 (\varepsilon_k^i)^2\right]}_{B_2} + R^2 \underbrace{\mathbb{E}\left[\left\|\vec{\varepsilon}^i\right\|_2^4 (\varepsilon_k^i)^2\right]}_{B_3} + A^2 \underbrace{\mathbb{E}\left[(\varepsilon_k^i)^2\right]}_{B_4=\sigma^2} = \tag{41}$$

$$-2\underbrace{\mathbb{E}\left[\langle\vec{g}, \vec{\varepsilon}^i\rangle\delta^i\right]}_{B_5} - 2R \underbrace{\mathbb{E}\left[\langle\vec{g}, \vec{\varepsilon}^i\rangle \left\|\vec{\varepsilon}^i\right\|_2^2 (\varepsilon_k^i)^2\right]}_{B_6} + 2R \underbrace{\mathbb{E}\left[\delta^i \left\|\vec{\varepsilon}^i\right\|_2^2 (\varepsilon_k^i)^2\right]}_{B_7} = \tag{42}$$

$$-2A \underbrace{\mathbb{E}\left[\langle\vec{g}, \vec{\varepsilon}^i\rangle (\varepsilon_k^i)^2\right]}_{B_8} + 2A \underbrace{\mathbb{E}\left[\delta^i (\varepsilon_k^i)^2\right]}_{B_9} + 2AR \underbrace{\mathbb{E}\left[\left\|\vec{\varepsilon}^i\right\|_2^2 (\varepsilon_k^i)^2\right]}_{B_{10}} \tag{43}$$

We will compute each component ($B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9, B_{10}$) separately:

$$B_1 = \mathbb{E}\left[\langle\vec{g}, \vec{\varepsilon}^i\rangle^2 (\varepsilon_k^i)^2\right] \overset{(a)}{=} \mathbb{E}\left[\sum_{j=1}^{d}\sum_{l=1}^{d} g_j g_l \varepsilon_j^i \varepsilon_l^i (\varepsilon_k^i)^2\right] \overset{(b)}{=} \sum_{j=1}^{d}\sum_{l=1}^{d} g_j g_l \mathbb{E}\left[\varepsilon_j^i \varepsilon_l^i (\varepsilon_k^i)^2\right] = \tag{44}$$

$$\overset{\text{(c)}}{=}\sum_{j\neq k}^{d}\sum_{l\neq j,k}^{d} g_j g_l \mathbb{E}\left[\varepsilon^i_j\right]\mathbb{E}\left[\varepsilon^i_l\right]\mathbb{E}\left[(\varepsilon^i_k)^2\right] + 2\sum_{j\neq k}^{d} g_j g_k \mathbb{E}\left[\varepsilon^i_j\right]\mathbb{E}\left[(\varepsilon^i_k)^3\right] + \sum_{j\neq k}^{d} g_j^2 \mathbb{E}\left[(\varepsilon^i_j)^2\right]\mathbb{E}\left[(\varepsilon^i_k)^2\right] + g_k^2\mathbb{E}\left[(\varepsilon^i_k)^4\right] = \tag{45}$$

$$\overset{\text{(d)}}{=}\sum_{j\neq k}^{d} g_j^2\sigma^4 + 3g_k^2\sigma^4 \overset{\text{(e)}}{=} (\|\vec{g}\|_2^2 + 2g_k^2)\sigma^4 \tag{46}$$

Some comments on the steps in the previous computation:

- (a): follows definition of inner product
- (b): follows the Linearity of expectation
- (c): ε^i_j and ε^i_k are independent $\forall k \neq j$
- (d): $\mathbb{E}\left[\varepsilon^i_k\right] = 0, \mathbb{E}\left[(\varepsilon^i_k)^2\right] = \sigma^2, \mathbb{E}\left[(\varepsilon^i_k)^3\right] = 0$ and $\mathbb{E}\left[(\varepsilon^i_k)^4\right] = 3\sigma^4, \forall k$
- (e): follows definition of norm

$$B_2 = \mathbb{E}\left[(\delta^i)^2(\varepsilon^i_k)^2\right] \overset{\text{(a)}}{=} \mathbb{E}\left[(\delta^i)^2\right]\mathbb{E}\left[(\varepsilon^i_k)^2\right] \overset{\text{(b)}}{=} \frac{\eta_i}{M}\sigma^2 \tag{47}$$

Some comments on the steps in the previous computation:

- (a): δ^i and ε^i_k are independent $\forall k$
- (b): $\mathbb{E}\left[(\delta^i)^2\right] = \frac{\eta_i}{M}$ and $\mathbb{E}\left[(\varepsilon^i_k)^2\right] = \sigma^2$

$$B_3 = \mathbb{E}\left[\left\|\vec{\varepsilon}^i\right\|_2^4 (\varepsilon^i_k)^2\right] \overset{\text{(a)}}{=} \mathbb{E}\left[\sum_{j=1}^{d}\sum_{l=1}^{d}(\varepsilon^i_j)^2(\varepsilon^i_l)^2(\varepsilon^i_k)^2\right] \overset{\text{(b)}}{=} \sum_{j=1}^{d}\sum_{l=1}^{d}\mathbb{E}\left[(\varepsilon^i_j)^2(\varepsilon^i_l)^2(\varepsilon^i_k)^2\right] = \tag{48}$$

$$\overset{\text{(c)}}{=}\sum_{j\neq k}^{d}\sum_{l\neq j,k}^{d}\mathbb{E}\left[(\varepsilon^i_j)^2\right]\mathbb{E}\left[(\varepsilon^i_l)^2\right]\mathbb{E}\left[(\varepsilon^i_k)^2\right] + 2\sum_{j\neq k}^{d}\mathbb{E}\left[(\varepsilon^i_j)^2\right]\mathbb{E}\left[(\varepsilon^i_k)^4\right] + \sum_{j\neq k}^{d}\mathbb{E}\left[(\varepsilon^i_j)^4\right]\mathbb{E}\left[(\varepsilon^i_k)^2\right] + \mathbb{E}\left[(\varepsilon^i_k)^6\right] = \tag{49}$$

$$\overset{\text{(d)}}{=}\sum_{j\neq k}^{d}\sum_{l\neq j,k}^{d}\sigma^6 + 2\sum_{j\neq k}^{d}3\sigma^6 + \sum_{j\neq k}^{d}3\sigma^6 + 15\sigma^6 = (d-1)(d-2)\sigma^6 + 6(d-1)\sigma^6 + 3(d-1)\sigma^6 + 15\sigma^6 \tag{50}$$

$$=(d^2 + 6d + 8)\sigma^6 \tag{51}$$

Some comments on the steps in the previous computation:

- (a): follows definition of norm
- (b): follows the Linearity of expectation
- (c): ε^i_j and ε^i_k are independent $\forall k \neq j$
- (d): $\mathbb{E}\left[(\varepsilon^i_k)^2\right] = \sigma^2$, $\mathbb{E}\left[(\varepsilon^i_k)^4\right] = 3\sigma^4$ and $\mathbb{E}\left[(\varepsilon^i_k)^5\right] = 15\sigma^6, \forall k$

$$B_5 = \mathbb{E}\left[\langle\vec{g}, \vec{\varepsilon}^i\rangle\delta^i\right] \overset{\text{(a)}}{=} \mathbb{E}\left[\langle\vec{g}, \vec{\varepsilon}^i\rangle\right]\mathbb{E}\left[\delta^i\right] \overset{\text{(b)}}{=} 0 \tag{52}$$

Some comments on the steps in the previous computation:

- (a): δ^i and ε^i_k are independent $\forall k$
- (b): $\mathbb{E}\left[\delta^i\right] = 0$

$$B_6 = \mathbb{E}\left[\langle\vec{g}, \vec{\varepsilon}^i\rangle\left\|\vec{\varepsilon}^i\right\|_2^2(\varepsilon^i_k)^2\right] \overset{\text{(a)}}{=} \mathbb{E}\left[\sum_{j=1}^{d}\sum_{l=1}^{d}g_j\varepsilon^i_j(\varepsilon^i_l)^2(\varepsilon^i_k)^2\right] \overset{\text{(b)}}{=} \sum_{j=1}^{d}\sum_{l=1}^{d}g_j\mathbb{E}\left[\varepsilon^i_j(\varepsilon^i_l)^2(\varepsilon^i_k)^2\right] = \tag{53}$$

$$\overset{\text{(c)}}{=}\sum_{j\neq k}^{d}\sum_{l\neq j,k}^{d} g_j\mathbb{E}\left[\varepsilon_j^i\right]\mathbb{E}\left[(\varepsilon_l^i)^2\right]\mathbb{E}\left[(\varepsilon_k^i)^2\right]+\sum_{j\neq k}^{d} g_j\mathbb{E}\left[\varepsilon_j^i\right]\mathbb{E}\left[(\varepsilon_k^i)^4\right]+ \tag{54}$$

$$+\sum_{j\neq k}^{d} g_j\mathbb{E}\left[(\varepsilon_j^i)^2\right]\mathbb{E}\left[(\varepsilon_k^i)^3\right]+\sum_{j\neq k}^{d} g_j\mathbb{E}\left[(\varepsilon_j^i)^3\right]\mathbb{E}\left[(\varepsilon_k^i)^2\right]+g_k\mathbb{E}\left[(\varepsilon_k^i)^5\right]= \tag{55}$$

$$\overset{\text{(d)}}{=}0 \tag{56}$$

Some comments on the steps in the previous computation:

- (a): follows definition of norm and inner product
- (b): follows the Linearity of expectation
- (c): ε_j^i and ε_k^i are independent $\forall k \neq j$
- (d): $\mathbb{E}\left[(\varepsilon_k^i)^2\right]=\sigma^2$, $\mathbb{E}\left[(\varepsilon_k^i)^4\right]=3\sigma^4$ and $\mathbb{E}\left[(\varepsilon_k^i)^5\right]=15\sigma^6, \forall k$

$$B_7 =\mathbb{E}\left[\delta^i\left\|\vec{\varepsilon}^i\right\|_2^2(\varepsilon_k^i)^2\right]\overset{\text{(a)}}{=}\mathbb{E}\left[\delta^i\right]\mathbb{E}\left[\left\|\vec{\varepsilon}^i\right\|_2^2(\varepsilon_k^i)^2\right]\overset{\text{(b)}}{=}0 \tag{57}$$

Some comments on the steps in the previous computation:

- (a): δ^i and ε_k^i are independent $\forall k$
- (b): $\mathbb{E}\left[\delta^i\right]=0$

$$B_8=\mathbb{E}\left[\langle\vec{g},\vec{\varepsilon}^i\rangle(\varepsilon_k^i)^2\right]\overset{\text{(a)}}{=}\mathbb{E}\left[\sum_{j=1}^{d} g_j\varepsilon_j^i(\varepsilon_k^i)^2\right]\overset{\text{(b)}}{=}\sum_{j=1}^{d} g_j\mathbb{E}\left[\varepsilon_j^i(\varepsilon_k^i)^2\right] \tag{58}$$

$$\overset{\text{(c)}}{=} g_j\mathbb{E}\left[(\varepsilon_k^i)^3\right]+\sum_{j\neq k}^{d} g_j\mathbb{E}\left[\varepsilon_j^i\right]\mathbb{E}\left[(\varepsilon_k^i)^2\right]\overset{\text{(d)}}{=}0 \tag{59}$$

Some comments on the steps in the previous computation:

- (a): follows definition of inner product
- (b): follows the Linearity of expectation
- (c):ε_j^i and ε_k^i are independent $\forall k \neq j$
- (d): $\mathbb{E}\left[(\varepsilon_k^i)\right]=0$, $\mathbb{E}\left[(\varepsilon_k^i)^2\right]=\sigma^2$ and $\mathbb{E}\left[(\varepsilon_k^i)^3\right]=0, \forall k$

$$B_9=\mathbb{E}\left[\delta^i(\varepsilon_k^i)^2\right]\overset{\text{(a)}}{=}\mathbb{E}\left[\delta^i\right]\mathbb{E}\left[(\varepsilon_k^i)^2\right]\overset{\text{(b)}}{=}0 \tag{60}$$

Some comments on the steps in the previous computation:

- (a): δ^i and ε_k^i are independent $\forall k$
- (b): $\mathbb{E}\left[\delta^i\right]=0$

$$B_{10}=\mathbb{E}\left[\left\|\vec{\varepsilon}^i\right\|_2^2(\varepsilon_k^i)^2\right]\overset{\text{(a)}}{=}\mathbb{E}\left[\sum_{j=1}^{d}(\varepsilon_j^i)^2(\varepsilon_k^i)^2\right]\overset{\text{(b)}}{=}\sum_{j=1}^{d}\mathbb{E}\left[(\varepsilon_j^i)^2(\varepsilon_k^i)^2\right]= \tag{61}$$

$$\overset{\text{(c)}}{=}\mathbb{E}\left[(\varepsilon_k^i)^4\right]+\sum_{j\neq k}^{d}\mathbb{E}\left[(\varepsilon_j^i)^2(\varepsilon_k^i)^2\right]\overset{\text{(d)}}{=}3\sigma^4+\sum_{j\neq k}^{d}\sigma^4=(d+2)\sigma^4 \tag{62}$$

Some comments on the steps in the previous computation:

- (a): follows definition of norm
- (b): follows the Linearity of expectation
- (c): ε_j^i and ε_k^i are independent $\forall k \neq j$
- (d): $\mathbb{E}\left[(\varepsilon_k^i)^2\right] = \sigma^2$ and $\mathbb{E}\left[(\varepsilon_k^i)^4\right] = 3\sigma^4, \forall k$

So, using equations (46),(47),(51),(52),(56),(57),(59),(60) and (62), we have the following final form for **the second non-central moment** of v_k^i:

$$\mathbb{E}\left[(v_k^i)^2\right] = \frac{\eta_i\sigma^2}{M} + (\|\vec{g}\|^2 + 2g_k^2)\sigma^4 + A^2\sigma^2 + R^2(d^2+6d+8)\sigma^6 + 2AR(d+2)\sigma^4 \tag{63}$$

Considering the approximation where we remove R-terms, we have:

$$\mathbb{E}\left[(v_k^i)^2\right] \approx \frac{\eta_i\sigma^2}{M} + (\|\vec{g}\|^2 + 2g_k^2)\sigma^4 + A^2\sigma^2 \tag{64}$$

A.3 Expected Lower Bound

We have the following lower bound:

$$\mathbb{E}\left(\tilde{\mathcal{L}}(\vec{\theta}) - \tilde{\mathcal{L}}(\vec{\theta}+\vec{z})\right) \geq -\underbrace{\mathbb{E}\left[\langle\vec{g},\vec{v}\,'\rangle\right]}_{C_1} - \frac{K\lambda_d}{2}\underbrace{\mathbb{E}\left[\|\vec{v}\,'\|_2^2\right]}_{C_2} \tag{65}$$

We will compute each component (C_1, C_2) separately:

$$C_1 = \mathbb{E}\left[\langle\vec{g},\vec{v}\,'\rangle\right] \overset{\text{(a)}}{=} \mathbb{E}\left[\langle\vec{g}, \frac{1}{2\lambda A}\sum_{i=1}^{\lambda}\vec{v}^{\,i}\rangle\right] \overset{\text{(b)}}{=} \frac{1}{2\lambda A}\sum_{i=1}^{\lambda}\mathbb{E}\left[\langle\vec{g},\vec{v}^{\,i}\rangle\right] = \tag{66}$$

$$\overset{\text{(c)}}{=} \frac{1}{2\lambda A}\sum_{i=1}^{\lambda}\mathbb{E}\left[\sum_{k=1}^{d} g_k v_k^i\right] \overset{\text{(d)}}{=} \frac{1}{2\lambda A}\sum_{i=1}^{\lambda}\sum_{k=1}^{d} g_k\mathbb{E}\left[v_k^i\right] \overset{\text{(e)}}{=} \frac{1}{2\lambda A}\sum_{i=1}^{\lambda}\sum_{k=1}^{d} -g_k^2\sigma^2 = \tag{67}$$

$$= -\frac{1}{2\lambda A}\lambda\|\vec{g}\|_2^2\sigma^2 = -\frac{\sigma^2}{2A}\|\vec{g}\|_2^2 \tag{68}$$

Some comments on the steps in the previous computation:

- (a): follows the definition of $\vec{v}\,'$
- (b): follows the Linearity of inner product and Linearity of expectation
- (c): follows the definition of inner product
- (d): follows the Linearity of expectation
- (e): follows equation 38

$$C_2 = \mathbb{E}\left[\|\vec{v}\,'\|_2^2\right] \overset{\text{(a)}}{=} \mathbb{E}\left[\langle\vec{v}\,',\vec{v}\,'\rangle\right] \overset{\text{(b)}}{=} \mathbb{E}\left[\langle\frac{1}{2\lambda A}\sum_{i=1}^{\lambda}\vec{v}^{\,i}, \frac{1}{2\lambda A}\sum_{j=1}^{\lambda}\vec{v}^{\,j}\rangle\right] = \tag{69}$$

$$\overset{\text{(c)}}{=} \frac{1}{4\lambda^2A^2}\sum_{i=1}^{\lambda}\sum_{j=1}^{\lambda}\mathbb{E}\left[\langle\vec{v}^{\,i},\vec{v}^{\,j}\rangle\right] \overset{\text{(d)}}{=} \frac{1}{4\lambda^2A^2}\sum_{i=1}^{\lambda}\sum_{j=1}^{\lambda}\mathbb{E}\left[\sum_{k=1}^{d} v_k^i v_k^j\right] \overset{\text{(e)}}{=} \frac{1}{4\lambda^2A^2}\sum_{i=1}^{\lambda}\sum_{j=1}^{\lambda}\sum_{k=1}^{d}\mathbb{E}\left[v_k^i v_k^j\right] = \tag{70}$$

$$\overset{\text{(f)}}{=} \frac{1}{4\lambda^2A^2}\sum_{i=1}^{\lambda}\sum_{j\neq i}^{\lambda}\sum_{k=1}^{d}\mathbb{E}\left[v_k^i\right]\mathbb{E}\left[v_k^j\right] + \frac{1}{4\lambda^2A^2}\sum_{i=1}^{\lambda}\sum_{k=1}^{d}\mathbb{E}\left[(v_k^i)^2\right] = \tag{71}$$

$$\overset{\text{(g)}}{=} \frac{1}{4\lambda^2A^2}\sum_{i=1}^{\lambda}\sum_{j\neq i}^{\lambda}\sum_{k=1}^{d} g_k^2\sigma^4 + \frac{1}{4\lambda^2A^2}\sum_{i=1}^{\lambda}\sum_{k=1}^{d}\left(\frac{\eta_i\sigma^2}{M} + (\|\vec{g}\|^2 + 2g_k^2)\sigma^4 + A^2\sigma^2\right) = \tag{72}$$

$$= \frac{\lambda(\lambda-1)\sigma^4}{4\lambda^2 A^2}\left\|\vec{g}\right\|_2^2 + \frac{1}{4\lambda^2 A^2}\sum_{i=1}^{\lambda}\sum_{k=1}^{d}\frac{\eta_i\sigma^2}{M} + \frac{1}{4\lambda^2 A^2}\sum_{i=1}^{\lambda}\sum_{k=1}^{d}(\left\|\vec{g}\right\|^2 + 2g_k^2)\sigma^4 + \frac{1}{4\lambda^2 A^2}\sum_{i=1}^{\lambda}\sum_{k=1}^{d}A^2\sigma^2 = \tag{73}$$

$$= \frac{(\lambda-1)\sigma^4}{4\lambda A^2}\left\|\vec{g}\right\|_2^2 + \frac{1}{M}\frac{\sigma^2 d}{4\lambda^2 A^2}\sum_{i=1}^{\lambda}\eta_i + \frac{\lambda(d+2)\sigma^4}{4\lambda^2 A^2}\left\|\vec{g}\right\|_2^2 + \frac{A^2 d\sigma^2}{4\lambda A^2} = \tag{74}$$

$$= \frac{1}{M}\frac{\sigma^2 d}{4\lambda^2 A^2}\sum_{i=1}^{\lambda}\eta_i + \frac{(\lambda+d+1)\sigma^4}{4\lambda A^2}\left\|\vec{g}\right\|_2^2 + \frac{A^2 d\sigma^2}{4\lambda A^2} \tag{75}$$

Some comments on the steps in the previous computation:

- (a): follows $\left\|\vec{x}\right\|_2^2 = \langle\vec{x},\vec{x}\rangle$
- (b): follows the definition of $\vec{v}\,'$
- (c): follows the Linearity of inner product and Linearity of expectation
- (d): follows the definition of inner product
- (e): follows the Linearity of expectation
- (f): v_k^i and v_k^j are independent $\forall i \neq j$
- (g): follows equation 38 and 64

Considering equations (68) and (75), we have the following final form for the expected lower bound on the improvement gained by updating the centre of mass $\vec{m}$:

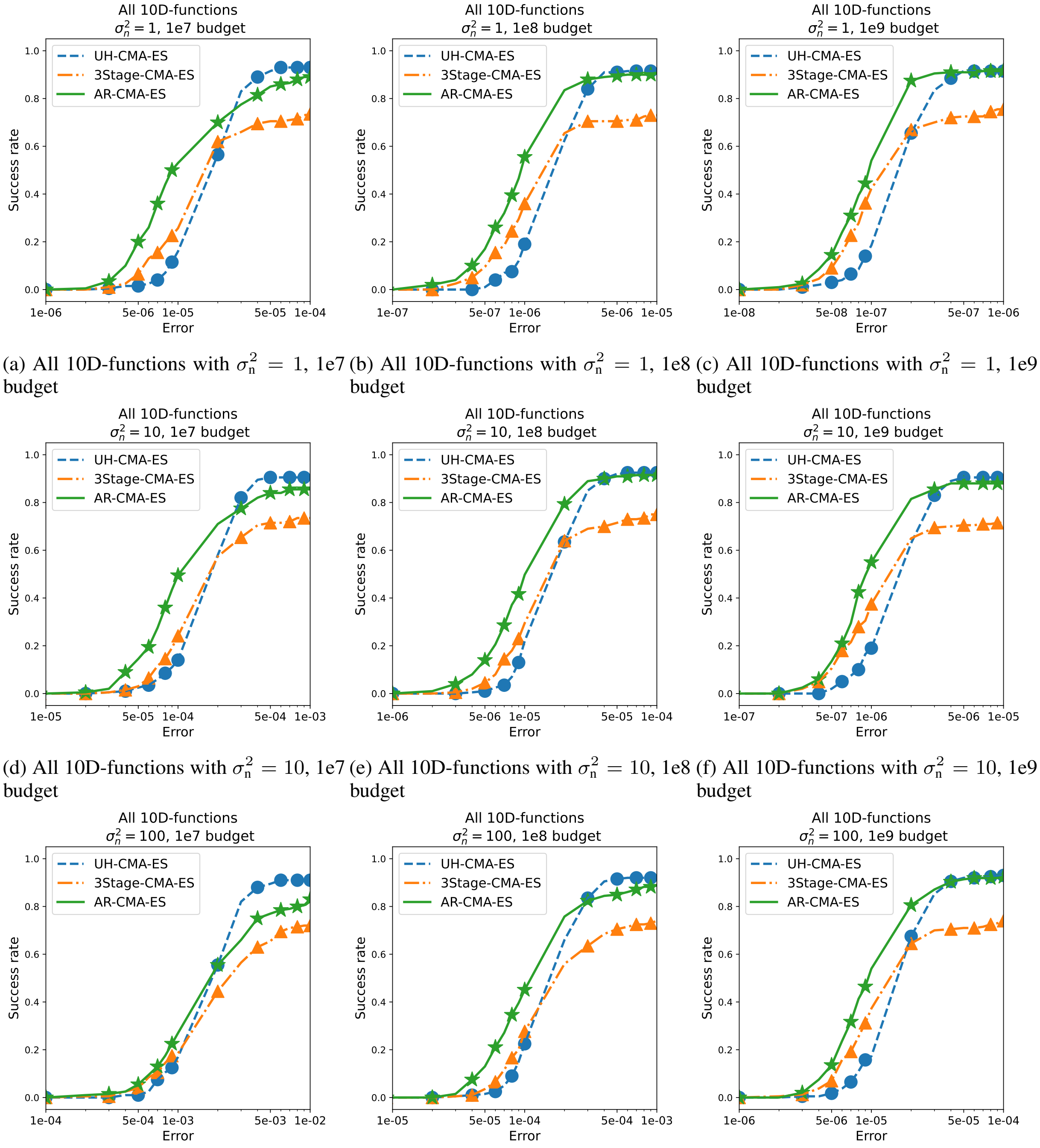
$$\mathbb{E}\left(\tilde{\mathcal{L}}(\vec{m}) - \tilde{\mathcal{L}}(\vec{m}+\vec{z})\right) \geq \tag{76}$$

$$\geq -\mathbb{E}\left[\langle\vec{g},\vec{v}\,'\rangle\right] - \frac{K\lambda_d}{2}\mathbb{E}\left[\left\|\vec{v}\,'\right\|_2^2\right] = \tag{77}$$

$$= \frac{\sigma^2}{2B}\left\|\vec{g}\right\|^2 - \frac{1}{M}\frac{K\lambda_d\sigma^2 d}{8\lambda^2 A^2}\sum_{i=1}^{\lambda}\eta_i - \frac{K\lambda_d(\lambda+d+1)\sigma^4}{8\lambda A^2}\left\|\vec{g}\right\|_2^2 - \frac{dK\lambda_d\sigma^2}{8\lambda} = \tag{78}$$

$$= \frac{\sigma^2}{2A}\left\|\vec{g}\right\|_2^2 - \frac{\sigma^4(\lambda+d+1)K\lambda_d}{8\lambda A^2}\left\|\vec{g}\right\|_2^2 - \frac{dK\lambda_d\sigma^2}{8\lambda} - \frac{1}{M}\frac{\sigma^2 dK\lambda_d}{8\lambda^2 A^2}\sum_{i=1}^{\lambda}\eta_i \tag{79}$$

B Appendix: All experiments



(a) All 10D-functions with $\sigma_\text{n}^2 = 1$, 1e7 budget

(b) All 10D-functions with $\sigma_\text{n}^2 = 1$, 1e8 budget

(c) All 10D-functions with $\sigma_\text{n}^2 = 1$, 1e9 budget

(d) All 10D-functions with $\sigma_\text{n}^2 = 10$, 1e7 budget

(e) All 10D-functions with $\sigma_\text{n}^2 = 10$, 1e8 budget

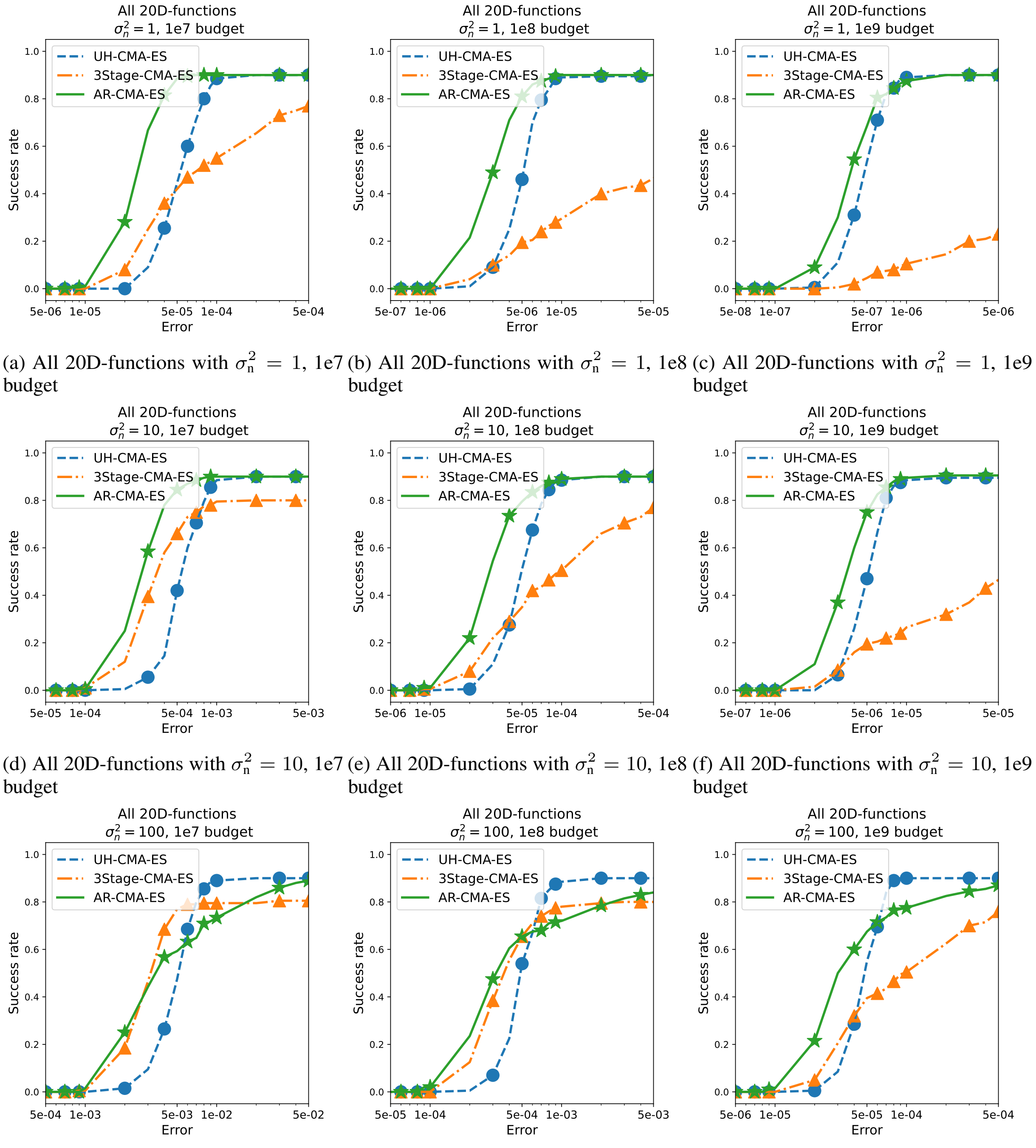
(f) All 10D-functions with $\sigma_\text{n}^2 = 10$, 1e9 budget

(g) All 10D-functions with $\sigma_\text{n}^2 = 100$, 1e7 budget

(h) All 10D-functions with $\sigma_\text{n}^2 = 100$, 1e8 budget

(i) All 10D-functions with $\sigma_\text{n}^2 = 100$, 1e9 budget

Figure 6: The combined Empirical Cumulative Distribution (ECDF) of all 10D benchmark functions across different noise levels ($\sigma_\text{n}^2 \in [1, 10, 100]$) and varying budgets (1e7,1e8, 1e9).



(a) All 20D-functions with $\sigma_n^2 = 1$, 1e7 budget

(b) All 20D-functions with $\sigma_n^2 = 1$, 1e8 budget

(c) All 20D-functions with $\sigma_n^2 = 1$, 1e9 budget

(d) All 20D-functions with $\sigma_n^2 = 10$, 1e7 budget

(e) All 20D-functions with $\sigma_n^2 = 10$, 1e8 budget

(f) All 20D-functions with $\sigma_n^2 = 10$, 1e9 budget

(g) All 20D-functions with $\sigma_n^2 = 100$, 1e7 budget

(h) All 20D-functions with $\sigma_n^2 = 100$, 1e8 budget

(i) All 20D-functions with $\sigma_n^2 = 100$, 1e9 budget

Figure 7: The combined Empirical Cumulative Distribution (ECDF) of all 10D benchmark functions across different noise levels ($\sigma_n^2 \in [1, 10, 100]$) and varying budgets (1e7,1e8, 1e9).

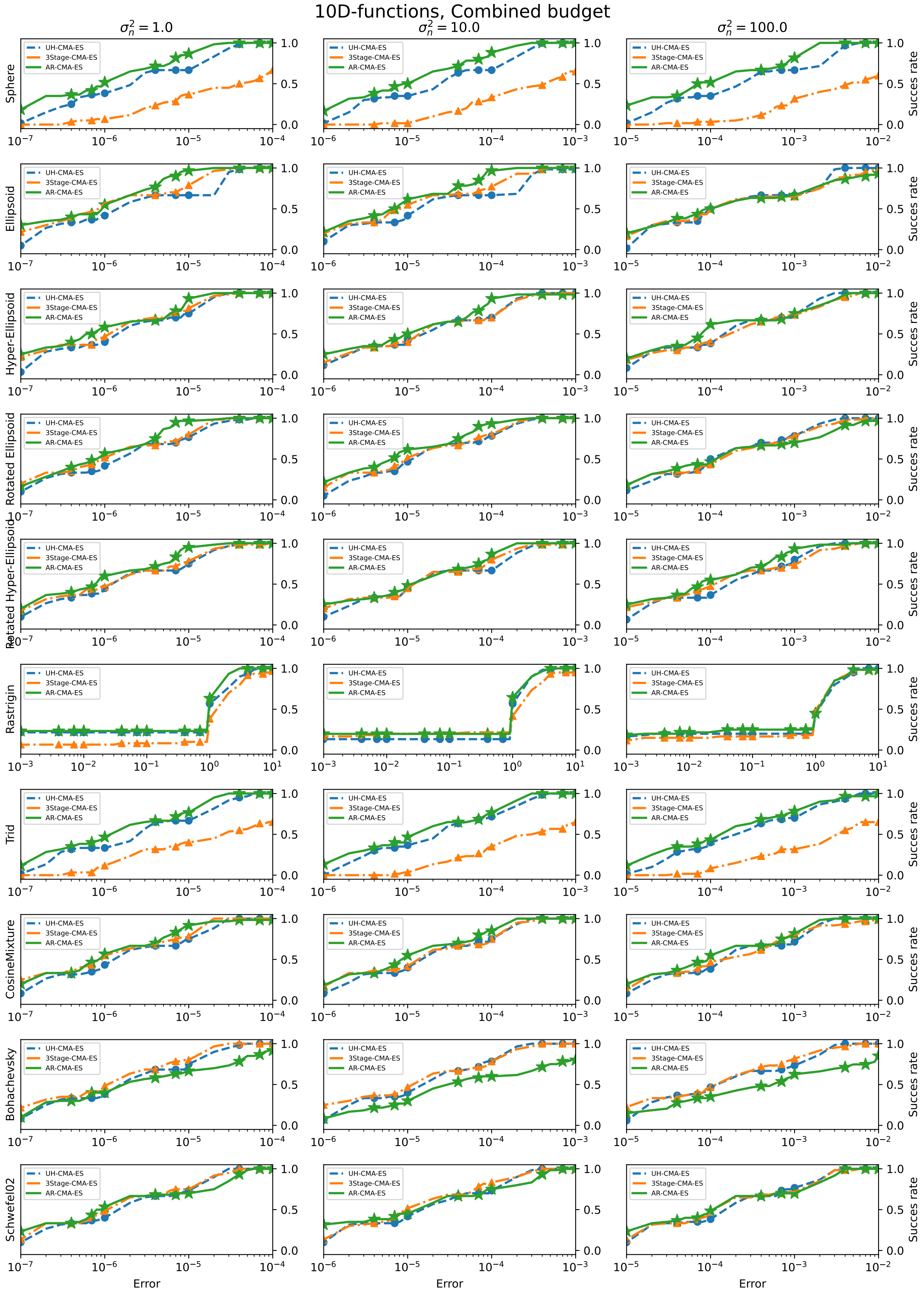


Figure 8: Empirical Cumulative Distribution Function (ECDF) for each 10D function across different noise levels ($\sigma_\mathrm{n}^2 \in [1, 10, 100]$).

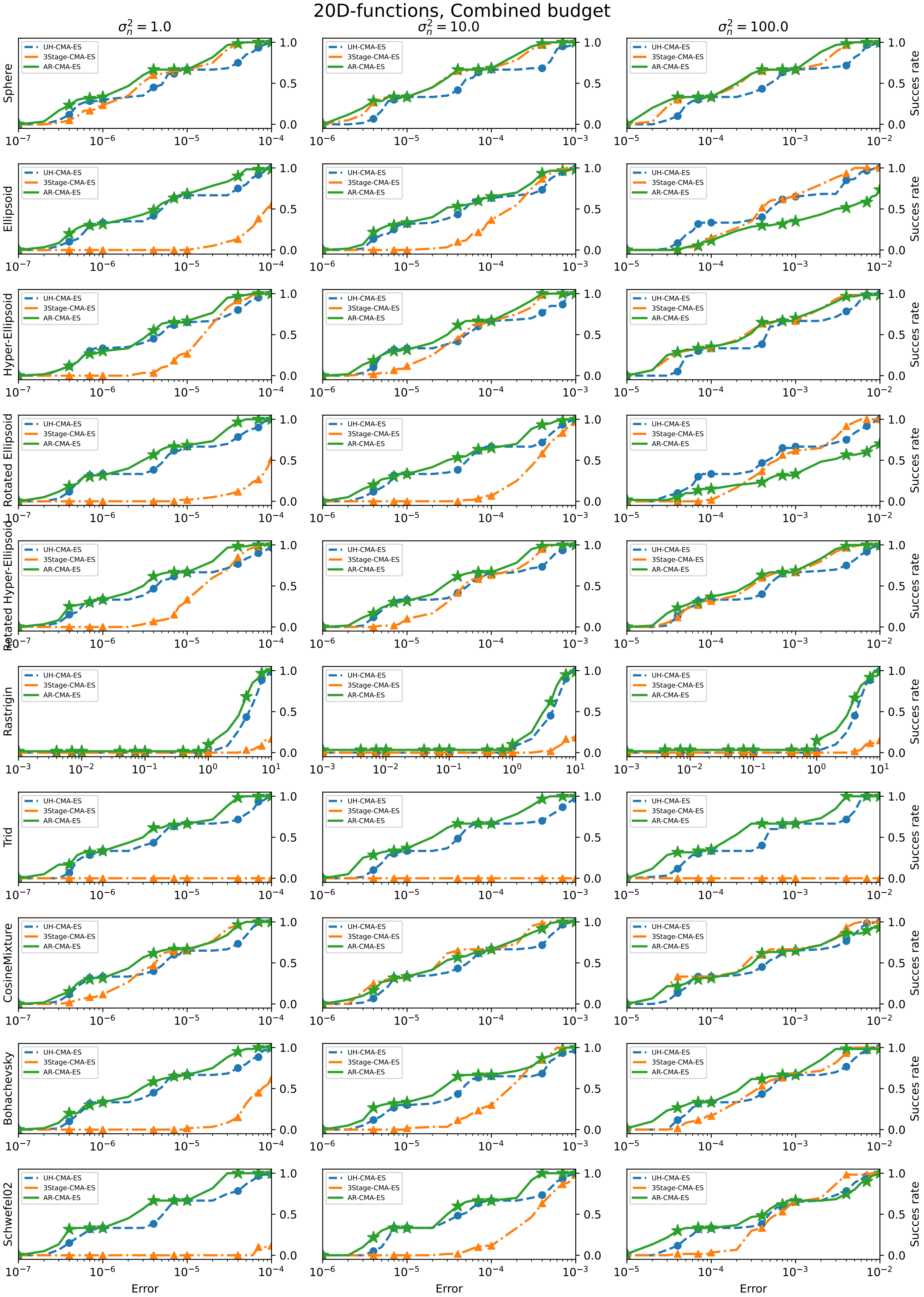


Figure 9: Empirical Cumulative Distribution Function (ECDF) for each 20D function across different noise levels ($\sigma_n^2 \in [1, 10, 100]$).