Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

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

Catalin-Viorel Dinu, Yash J. Patel, Xavier Bonet-Monroig, Hao Wang

2025

Preprint:

Copyright ACM 2025. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in GECCO '25: Proceedings of the Genetic and Evolutionary Computation Conference Pages 710 - 718
https://doi.org/10.1145/3712256.3726352,
https://doi.org/10.1145/3712256.3726352.



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

Catalin-Viorel Dinu
LIACS, Leiden University
Leiden, The Netherlands
viorel.dinu00@gmail.com

Yash J. Patel
applied Quantum algorithms, Leiden University
LIACS, Leiden University
Leiden, The Netherlands
y.j.patel@liacs.leidenuniv.nl

Xavier Bonet-Monroig
Honda Research Institute Europe GmbH
Frankfurt, Germany
Instituut-Lorentz, Leiden University
Leiden, The Netherlands
xavier.bonet@honda-ri.de

Hao Wang
applied Quantum algorithms, Leiden University
LIACS, Leiden University
Leiden, The Netherlands
h.wang@liacs.leidenuniv.nl

ABSTRACT

The Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) is one of the most advanced algorithms in numerical black-box optimization. For noisy objective functions, several approaches were proposed to mitigate the noise, e.g., re-evaluations of the same solution or adapting the population size. In this paper, we devise a novel method to adaptively choose the optimal re-evaluation number for function values corrupted by additive Gaussian white noise. We derive a theoretical lower bound of the expected improvement achieved in one iteration of CMA-ES, given an estimation of the noise level and the Lipschitz constant of the function's gradient. Solving for the maximum of the lower bound, we obtain a simple expression of the optimal re-evaluation number. We experimentally compare our method to the state-of-the-art noise-handling methods for CMA-ES on a set of artificial test functions across various noise levels, optimization budgets, and dimensionality. Our method demonstrates significant advantages in terms of the probability of hitting near-optimal function values.

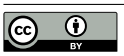
CCS CONCEPTS

• **Theory of computation** → *Random search heuristics*; • **Computing methodologies** → **Continuous space search**.

KEYWORDS

Evolutionary Strategy, Lipschitz Constant, Noisy Optimization, CMA-ES, Black-Box Optimization, Additive Gaussian Noise

ACM Reference Format:
Catalin-Viorel Dinu, Yash J. Patel, Xavier Bonet-Monroig, and Hao Wang. 2025. An Adaptive Re-evaluation Method for Evolution Strategy under Additive Noise. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO '25).* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3712256.3726352



This work is licensed under a Creative Commons Attribution 4.0 International License.

GECCO '25, July 14–18, 2025, Málaga, Spain

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1465-8/2025/07
https://doi.org/10.1145/3712256.3726352

1 INTRODUCTION

Optimization problems are central to various scientific and engineering fields [18, 21]. Typically, these problems are analyzed under ideal conditions, assuming a noiseless environment. However, many real-world optimization problems involve noise, which can distort the true objective function, making the optimization landscape less reliable. Noise in the objective function can arise from various sources, such as measurement errors, environmental variability, or the inherent randomness of the system [26]. In response, several noise models have been proposed in the literature, which can broadly be categorized into two types:

- **Additive Noise** [8, 27]: This model assumes that the noise added to the objective function value is independent of the function value itself. It is expressed as $\tilde{\mathcal{L}}(\vec{x}) = \mathcal{L}(\vec{x}) + \tau\mathcal{N}(0,1)$, where τ represents the standard deviation of the noise.
- **Multiplicative Noise** [31]: In this model, the noise scales with the objective function value. It is expressed as $\tilde{\mathcal{L}}(\vec{x}) = (1+\tau z)\mathcal{L}(\vec{x})$, where z can be a Gaussian or uniform random variable.

In noisy black-box optimization, evolutionary algorithms (EAs) have shown promising performance due to the intrinsic population dynamics, which improves the robustness against noise [3, 4, 15, 26]. The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [14] is the state-of-the-art algorithm among EAs [32]. To further improve the capabilities of CMA-ES over noisy evaluations, several noise-handling methods have been proposed, of which the most popular are:

- **Population size adaptation** [16, 19, 24] involves dynamically modifying the population size to mitigate the noise in function values. Using a larger population increases the probability of selecting candidate points whose fitness values are closer to the noiseless value.
- **Learning rate adaptation** [25] adjusts the learning rate according to the noise level. Smaller steps can reduce sensitivity

to noise, leading to steady and progressive gains toward the optimum.

- **Re-evaluating of the objective function** is the most common noise-handling method. for each candidate multiple times and then take the average to reduce the noise effect [1, 2, 5, 15, 29]. This approach helps smooth out artificial fluctuations in the function landscape induced by the noise.

In this work, we devise a novel method to determine the optimal re-evaluation number for each candidate point. For this, we consider objective functions with Lipschitz continuous gradient and an additive Gaussian noise model and derive a lower bound on the expected improvement of the function values at each iteration of CMA-ES. In turn, the analytical bound gives us a simple expression of the optimal number of re-evaluations for each candidate of the CMA-ES iteration. We implement this method into CMA-ES, an extension that we call **Adaptive Re-evaluation** method (AR-CMA-ES). To benchmark our method, we use a wide range of test functions with different levels of noise of the objectives. Our experimental results show that AR-CMA-ES outperforms existing noise-handling methods at all noise levels, achieving a much better accuracy-to-target across the test benchmarks. To summarize our contributions,

- we have derived a theoretical lower bound of the expected improvement of noiseless function values in one iteration of CMA-ES regardless of the objective function;
- we have chosen the optimal re-evaluation number by maximizing the efficiency metric, which is the expected improvement normalized by the re-evaluations;
- we have obtained a simple analytical expression for the optimal re-evaluation number and provide estimation procedures for the parameters required by the expression.

2 BACKGROUND

Problem formulation: We aim to minimize a single-objective, black-box, differentiable function $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$. In this study, we specifically address the scenario involving additive Gaussian noise, noting that similar analytical approaches can be applied to other types of noise. The noisy function value is represented as:

$$\tilde{\mathcal{L}}(\vec{x}) = \mathcal{L}(\vec{x}) + \tau\mathcal{N}(0,1).$$

We assume that the gradient of the function $\mathcal{L}$ is Lipschitz continuous, meaning $\exists K < \infty$ such that $\|\nabla\mathcal{L}(\vec{x}) - \nabla\mathcal{L}(\vec{x}')\|_2 \leq K\|\vec{x} - \vec{x}'\|_2$ for all $\vec{x}, \vec{x}' \in \mathbb{R}^d$.

The re-evaluation method estimated $\mathcal{L}(\vec{x})$ through the sample mean, is commonly used as a noise-mitigation method. According to the Central Limit Theorem (CLT), we have $\sqrt{M}(\mathcal{L}(\vec{x}) - \bar{\mathcal{L}}(\vec{x})) \xrightarrow{d} \tau\mathcal{N}(0,1)$, where $\bar{\mathcal{L}}(\vec{x}) = M^{-1}\sum_{i=1}^{M} y_i$ is computed by taking M independent and identically distributed (i.i.d.) samples y_i drawn from $\tilde{\mathcal{L}}(\vec{x})$.

Determining the appropriate value of M is crucial. Ideally, M should be large enough to ensure that the re-evaluated estimates $\bar{\mathcal{L}}(\vec{x}^i)$ and $\bar{\mathcal{L}}(\vec{x}^j)$ can be distinguished with high probability, i.e., $\tau/\sqrt{M} \ll |\mathcal{L}(\vec{x}^i) - \mathcal{L}(\vec{x}^j)|$. Considering that the set $\{\vec{x}^i\}_i$ is contained within a compact subset of $\mathbb{R}^d$, we encounter the following two scenarios:

- When $\mathcal{L}$ exhibits a large local Lipschitz constant, a smaller M is sufficient since $|\mathcal{L}(\vec{x}^i) - \mathcal{L}(\vec{x}^j)|$ is relatively large.

Algorithm 1: AR-CMA-ES. Our modifications to the standard CMA-ES are highlighted.

Procedure: AR-CMA-ES($\tilde{\mathcal{L}}$, B, λ, $\vec{x}_L, \vec{x}_U$);
Input: a noisy objective function $\tilde{\mathcal{L}}$, population size λ, evaluation budget B, $[\vec{x}_L, \vec{x}_U] \subseteq \mathbb{R}^d$;
$\sigma \leftarrow 0.1 \times \|\vec{x}_U - \vec{x}_L\|_\infty$; ▷ step size
$\mathbf{C} \leftarrow \mathbf{I}$; ▷ covariance matrix
$M \leftarrow 1, \vec{g} \leftarrow 0$;
Sample $\vec{m}$ u.a.r. in $[\vec{x}_L, \vec{x}_U]$;
Estimate the noise level τ;
repeat
  for $i \in [1..\lambda]$ **do**
    $\vec{x}^i \leftarrow \vec{m} + \mathbf{C}^{1/2}\vec{\varepsilon}^i$, $\vec{\varepsilon}^i \sim \sigma\mathcal{N}(0, \mathbf{I})$;
    $\bar{\mathcal{L}}(\vec{x}^i) \leftarrow \sum_{i=1}^{M} \tilde{\mathcal{L}}(\vec{x}^i)/M$;
    $\Delta\bar{\mathcal{L}}(\vec{x}^i) \leftarrow \bar{\mathcal{L}}(\vec{m}) - \bar{\mathcal{L}}(\vec{x}^i)$;
  $A \leftarrow -\min\{\Delta\bar{\mathcal{L}}(\vec{x}^i)\}_i$;
  $B \leftarrow B - \lambda M$;
  $w_i \leftarrow \frac{\Delta\bar{\mathcal{L}}^i + A}{\sum_{k=1}^{\lambda}\Delta\bar{\mathcal{L}}^k + \lambda A}$; ▷ Eq. (4)
  $\vec{m} \leftarrow \vec{m} + \sum_{i=1}^{\lambda} w_i \mathbf{C}^{1/2}\vec{\varepsilon}^i$;
  $s_{\max} \leftarrow$ the largest eigenvalue of $\mathbf{C}$;
  Estimate the Lipschitz constant K of $\nabla\mathcal{L}$;
  $\vec{g} \leftarrow (1-\alpha)\vec{g} - \frac{\alpha}{\lambda\sigma^2}\sum_{i=1}^{\lambda}(\Delta\bar{\mathcal{L}}^i + A)\vec{\varepsilon}^i$;
  $a \leftarrow \frac{dKs_{\max}\tau^2}{4\lambda}$; ▷ Eq. (15)
  $b \leftarrow (A - \frac{\sigma^2(\lambda+d+1)Ks_{\max}}{4\lambda})\|\vec{g}\|_2^2 - \frac{A^2 dKs_{\max}}{4\lambda}$;
  $M \leftarrow (1-\beta)M + 2a/b$; ▷ Eq. (16)
  // See [14]
  Update $\mathbf{C}$ and σ with $\{w_i\}_i$ and $\{\vec{\varepsilon}^i\}_i$;
until $B \leq 0$;
Output: $\vec{m}$

- When the Lipschitz constant is small, the difference $|\mathcal{L}(\vec{x}^i) - \mathcal{L}(\vec{x}^j)|$ is also small, requiring a much larger M to ensure that the noise does not obscure these differences.

CMA-ES. The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [14] is a widely used black-box optimization algorithm for continuous, single-objective problems [5, 15, 20, 28]. CMA-ES maintains a "center of mass" $\vec{m} \in \mathbb{R}^d$, which estimates the global minimum. In each iteration CMA-ES generates independent and identically distributed (i.i.d.) candidate solutions $\{\vec{x}^i\}_{i=1}^{\lambda}$ from a multivariate Gaussian:

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

where $\vec{\varepsilon}^i$ is referred to as the i-th mutation vector, σ is the step size that scales the mutation vector, and $\mathbf{C}$ is the covariance matrix. Both σ and $\mathbf{C}$ are self-adapted within CMA-ES [14]. CMA-ES ranks the candidates based on their objective values (with ties broken randomly) as follows: $\mathcal{L}(\vec{x}^{1:\lambda}) < \mathcal{L}(\vec{x}^{2:\lambda}) < \ldots < \mathcal{L}(\vec{x}^{\lambda:\lambda})$.

The center of mass is then updated using weighted recombination of the top-μ candidates ($\mu < \lambda$):

$$\vec{m} \leftarrow \vec{m} + \vec{z},\ \vec{z} = \sum_{i=1}^{\mu} w_i \mathbf{C}^{1/2}\vec{\varepsilon}^{\,i:\lambda},\ \sum_{i=1}^{\mu} w_i = 1. \tag{2}$$

Here, $\vec{\varepsilon}^{\,i:\lambda}$ is the mutation vector that generates $\vec{x}^{\,i:\lambda}$. By default, CMA-ES uses a monotonically decreasing function w.r.t. the ranking of these candidates for assigning the weight w_i.

For a noisy objective $\bar{\mathcal{L}}$, it is common to re-evaluate each candidate point $\vec{x}$ over M trials and provide CMA-ES with the average function value $\bar{\mathcal{L}}$. Based on CLT, we approximately have $\bar{\mathcal{L}}(\vec{x}) \sim \mathcal{L}(\vec{x}) + \mathcal{N}(0, \tau^2/M)$, when M is large. There are several proposals to extend CMA-ES to minimize noisy functions (see Sec. 4).

3 ADAPTIVE RE-EVALUATION (AR-CMA-ES)

We summarize our method in Alg. 1, where our modifications to the standard CMA-ES are highlighted. Our method extends the CMA-ES algorithm to handle additive noise, with the primary objective of dynamically estimating the optimal number of function re-evaluations required for each candidate. We first modify Eq. (2) to consider all mutation vectors:

$$\vec{z} = \sum_{i=1}^{\lambda} w_i \mathbf{C}^{1/2}\vec{\varepsilon}^{\,i}, \quad \vec{\varepsilon}^{\,i} \sim \sigma\mathcal{N}(0, \mathbf{I}), \tag{3}$$

where the recombination weights are determined from the noisy function values. The rationale for this consideration is that taking an average over a larger set helps reduce the impact of noises on the weight.

Modify the search direction $\vec{z}$: Instead of using the default weighting scheme, we consider the proportional weights for ease of analysis, a method commonly applied in evolutionary algorithms [10]. This approach assigns a positive weight proportional to the loss value of each mutation

$$w_i = \frac{\Delta\bar{\mathcal{L}}^i + A}{\sum_{k=1}^{\lambda} \Delta\bar{\mathcal{L}}^k + \lambda A}, \tag{4}$$

where $\Delta\bar{\mathcal{L}}^i = \bar{\mathcal{L}}(\vec{m}) - \bar{\mathcal{L}}(\vec{m} + \mathbf{C}^{1/2}\vec{\varepsilon}^{\,i})$ represents the change in the noisy objective function value, and A is chosen as the smallest possible value that ensures all weights remain positive with high probability. Considering the first-order Taylor expansion of $\bar{\mathcal{L}}(\vec{m} + \mathbf{C}^{1/2}\vec{\varepsilon}^{\,i})$, then,

$$\begin{aligned}\Delta\bar{\mathcal{L}}^i &= -\left\langle \nabla\mathcal{L}(\vec{m}), \mathbf{C}^{1/2}\vec{\varepsilon}^{\,i}\right\rangle + \mathcal{O}\left(\left\|\mathbf{C}^{1/2}\vec{\varepsilon}^{\,i}\right\|_2^2\right) + \delta^i \\ &= -\langle \vec{g}, \vec{\varepsilon}^{\,i}\rangle + R\left\|\vec{\varepsilon}^{\,i}\right\|_2^2 + \delta^i, \text{ for some } R \in \mathbb{R},\end{aligned} \tag{5}$$

with $\vec{g} = \mathbf{C}^{1/2}\nabla\mathcal{L}(\vec{m})$ and $\delta^i \sim \mathcal{N}\left(0, \tau^2/M\right)$ being i.i.d. noise in function value, and independent of $\vec{\varepsilon}^{\,i}$.

When the step-size σ is small, we have $\Delta\bar{\mathcal{L}}^i \sim \sigma\|\vec{g}\|_2 + \mathcal{N}(0, \tau^2/M)$. Choosing $A \geq c\tau/\sqrt{M} - \sigma\|\vec{g}\|_2$ will ensure $\Pr(\Delta\bar{\mathcal{L}}^i + A \leq 0) \leq \Phi(-c)$ (e.g., $c = 3$ gives ca. 0.15% chance of realizing negative weights). Also, since A is a probabilistic upper bound of $\Delta\bar{\mathcal{L}}^i$, we can relax the denominator of Eq. (4) to $2\lambda A$, which leads to a modified search direction:

$$\vec{z}\,' = \frac{1}{2\lambda A}\sum_{i=1}^{\lambda}(\Delta\bar{\mathcal{L}}^i + A)\mathbf{C}^{1/2}\vec{\varepsilon}^{\,i}. \tag{6}$$

The modified search direction $\vec{z}\,'$ is easier to analyze and keeps the direction of $\vec{z}$ with high probability:

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

The probability that $\vec{z}\,'$ inverts $\vec{z}$ is $1 - \Phi(c)$ which is negligible for $c \geq 3$. Also, notice that $\mathbb{E}(\|\vec{z}'\| \mid \vec{z}) = \frac{1}{2}\|z\|$. It suffices to halve CMA-ES's parameter in the step-size adaptation to ensure our modification does not affect other dynamics thereof. Hence, we can safely use the modified search direction $\vec{z}\,'$ in the following analysis.

Efficiency in noisy optimization. For a search algorithm, it is natural to maximize the expected improvement induced by the random search direction $\vec{z}\,'$, i.e., $\mathbb{E}(\mathcal{L}(\vec{m}) - \mathcal{L}(\vec{m} + \vec{z}\,'))$. Since $\vec{z}\,'$ is determined from the noisy function values, the more function re-evaluations (M) we use, the more likely $\vec{z}\,'$ would be a descending direction. Hence, in the noisy scenario, it is more sensible to maximize the expected improvement while minimizing M, resulting in an *efficiency metric* (similar to the one proposed in [12])

$$\gamma = \frac{\mathbb{E}\left[\mathcal{L}(\vec{m}) - \mathcal{L}(\vec{m} + \vec{z}\,')\right]}{M}. \tag{7}$$

Given an arbitrary black-box function, it is challenging to compute the exact form γ. Instead, we seek a lower bound of it and then determine the optimal value of M by maximizing the lower bound.

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 $\Delta\bar{\mathcal{L}}$ is not affected by the change of basis. In the new coordinate system, the search direction is:

$$\vec{v}\,' = \mathbf{C}^{-1/2}\vec{z}\,' = \frac{1}{2\lambda A}\sum_{i=1}^{\lambda}\underbrace{(\Delta\bar{\mathcal{L}}^i + A)\vec{\varepsilon}^{\,i}}_{\vec{v}^{\,i}} \tag{8}$$

We can obtain the first moment and the second moment of the component of $\vec{v}^{\,i}$ (see Appendix B for the details). For $k \in [1..d]$ we have:

$$\mathbb{E}\left[v_k^i\right] = -g_k\sigma^2 \tag{9}$$

$$\mathbb{E}\left[(v_k^i)^2\right] = \frac{\tau^2\sigma^2}{M} + \left(\|\vec{g}\|_2^2 + 2g_k^2\right)\sigma^4 + A^2\sigma^2 \tag{10}$$

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

Using the above statistical property of $\vec{v}\,'$ and quadratic upper bound of the loss function (see Thm. 1), we bound from below the expected improvement (see Appendix C for the derivation):

$$\begin{aligned}&\mathbb{E}\left[\mathcal{L}(\vec{m}) - \mathcal{L}(\vec{m} + \vec{z}\,')\right] \\ &= \mathbb{E}\left[\mathcal{L}(\vec{m}) - \mathcal{L}(\vec{m} + \mathbf{C}^{1/2}\vec{v}\,')\right] &(11)\\ &\geq \mathbb{E}\left[-\left\langle \vec{g}, \vec{v}\,'\right\rangle - \frac{K}{2}\left\|\mathbf{C}^{1/2}\vec{v}\,'\right\|_2^2\right] &(12)\\ &\geq -\mathbb{E}\left\langle \vec{g}, \vec{v}\,'\right\rangle - \frac{Ks_{\max}}{2}\mathbb{E}\left\|\vec{v}\,'\right\|_2^2 &(13)\\ &= \frac{\sigma^2}{2A}\|\vec{g}\|_2^2 - \frac{\sigma^4(\lambda + d + 1)Ks_{\max}}{8\lambda A^2}\|\vec{g}\|_2^2 - \frac{dKs_{\max}\sigma^2}{8\lambda} \\ &\quad - \frac{1}{M}\frac{\sigma^2 dKs_{\max}\tau^2}{8\lambda A^2} &(14)\end{aligned}$$

where $s_{\max}$ is the largest eigenvalue of $\mathbf{C}$ and K is the Lipschitz constant of $\nabla\mathcal{L}$.

REMARK. *From Eq.* (11) *to* (12)*, we use the quadratic upper bound of real-analytic functions (see Theorem 1). From Eq.* (12) *to* (13)*, we take the fact that* $\left\|\mathrm{C}^{1/2}\vec{v}\,'\right\|_2 \leq \left\|\mathrm{C}^{1/2}\right\|_2 \|\vec{v}\|_2$*, and* $\left\|\mathrm{C}^{1/2}\right\|_2 = \sqrt{s_{max}}$*. The derivation of Eq.* (14) *from* (13) *requires Eq.* (9) *and Eq.* (10)*.*

Consequently, we obtain a lower bound on the efficiency:

$$\gamma \geq \frac{\sigma^2}{2A^2}\left(aM^{-2} + bM^{-1}\right), \tag{15}$$

where,

$$a = \frac{dKs_{\max}\tau^2}{4\lambda},$$
$$b = \left(A - \frac{\sigma^2(\lambda + d + 1)Ks_{\max}}{4\lambda}\right)\|\vec{g}\|_2^2 - \frac{A^2 dKs_{\max}}{4\lambda}.$$

Eq. (15) is quadratic function of M^{-1}. Obviously, $a > 0$. If term $b > 0$, then there is a unique maximizer thereof in $[0, \infty)$:

$$M^* = -\frac{2a}{b}. \tag{16}$$

REMARK. *In practice, we notice that the optimal value calculated in Eq.* (16) *is prone to numerical instability. Therefore, we apply exponential smoothing to* M^* *in each iteration:*

$$M_t = (1-\beta)M_{t-1} + \beta M^*,\ \beta \in (0,1).$$

The initial value M_0 *should be small and specified by the user. We take* M_t *re-evaluations for each candidate solution in iteration* t*. Whenever* $b < 0$*,* M^* *is negative, we simply ignore* M^**, pause the above smoothing operation, and use the* M_t *from the last iteration for the re-evaluation.*

We further validate the theoretical lower bound of γ on the 10-dimensional noisy sphere function with $\tau = 1, \lambda = 20$: we measure, for a range of different re-evaluation number M, the empirical improvement over M from 50 independent simulations of the mutation at iteration 100 (or any other iterations in the convergent phase). We show the result in Fig. 1, which numerically validates the correctness of the lower bound and, more importantly, shows that the lower bound curve resembles the trend of the empirical one. As a result, the optimal re-evaluations M^* (green star) upper-bounds the optimum estimated from the empirical curve (red star).
There are a few unknown parameters needed in the lower bound (Eq. (15)). We discuss how to estimate those as follows.

Estimate the Lipschitz constant for $\nabla\mathcal{L}$*:* Lipschitz constant estimation (Lipschitz learning algorithms) is an active research topic [11, 17, 30], and we have no intention of developing new estimation methods in this work. Instead, we propose a feasible solution based on fitting a local Gaussian process regression to the population. For black-box problems, we employ a similar estimation method as in [11]: we fit a local Gaussian process model to the population $\{(\vec{x}^{\,i}, \bar{\mathcal{L}}(\vec{x}^{\,i}))\}_{i=1}^{\lambda}$, specified by zero prior mean function and Gaussian kernel with white noise to handle the noisy function value: $k(\vec{x},\vec{x}') = \exp(-\theta\|\vec{x}-\vec{x}'\|^2) + \tau/\sqrt{M}\mathbb{1}_{\{\vec{x}'\}}(\vec{x})$. Let $H(\vec{x})$ be the Hessian matrix of the posterior mean function at point $\vec{x}$ and C denote the convex hull of $\{\vec{x}^{\,i}\}_i$, we can show $\widehat{K} = \max_{\vec{x}\in C}\|H(\vec{x})\|_2$ is a valid estimate to Lipschitz constant of $\nabla\mathcal{L}$ restricted to C: Applying the mean-value theorem, we have

$$\left\|\nabla\mathcal{L}(\vec{x}) - \nabla\mathcal{L}(\vec{x}')\right\|_2 = \left\|H(\vec{z})(\vec{x}-\vec{x}')\right\| \leq \|H(\vec{z})\|_2\left\|\vec{x}-\vec{x}'\right\|_2,$$

where $H(\vec{z})$ is the Hessian at $\vec{z}$ and $\vec{z} = (1-t)\vec{x} + t\vec{x}'$, for some $t \in (0,1)$. We have, for all $\vec{x},\vec{x}' \in C, \|\nabla\mathcal{L}(\vec{x}) - \nabla\mathcal{L}(\vec{x}')\|_2 \leq \max_{\vec{z}\in C}\|H(\vec{z})\|_2\|\vec{x}-\vec{x}'\|_2$, suggesting $\max_{\vec{z}\in C}\|H(\vec{z})\|_2$ can serve as the Lipschitz constant estimation. To efficiently compute $\widehat{K}$, we approximately solve the above maximization problem by sampling $100d$ points u.a.r. in C.

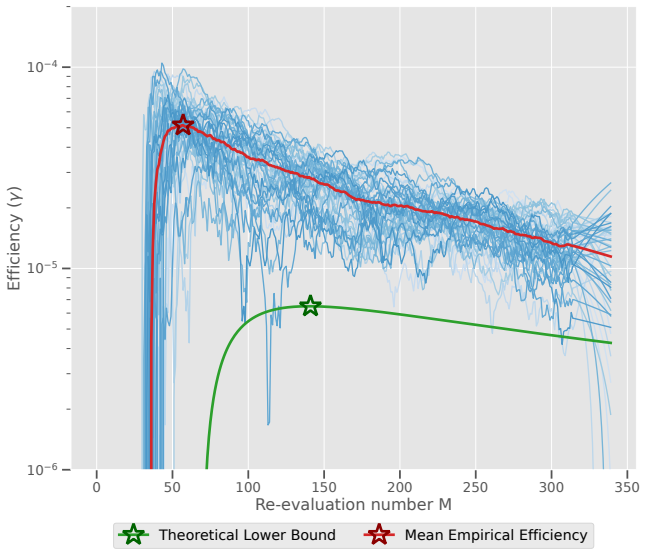


Figure 1: On the sphere function, the theoretical lower bound of the efficiency (green curve) and the empirical efficiency curve (red curve), i.e., the empirical improvement over the re-evaluation number, estimated from 50 independent simulations of the mutation of CMA-ES at iteration 100. Each simulated result is shown in light blue curves. We depict, in the star symbol, the maximum of both empirical and theoretical curves.

Estimate the noise τ*:* Since we assume homogeneous additive noise, it suffices to calculate the unbiased sample standard deviation $\hat{s}(M)$ of the function value at a randomly chosen point for various values of M before invoking CMA-ES. Using the relationship $\mathbb{E}(\hat{s}(M)) = \tau/\sqrt{M}$, a simple curve-fitting of $\hat{s}(M)$ can provide a robust estimate for τ.

Estimate $\vec{g}$*:* In Eq. (9) implies that the mutation vectors are unbiased estimators of the gradient: $\vec{g} = -\mathbb{E}(\vec{v}^{\,i})/\sigma^2$ for $i \in [1..\lambda]$. We further reduce the variance of this estimator by averaging over all candidates, i.e., $\vec{g}^* = -\lambda^{-1}\sum_{i=1}^{\lambda}\vec{v}^{\,i}/\sigma^2$. Taking Eq. (10), we have the variance of the estimate: $\mathrm{Var}(g_k) = (2/\lambda - 1)g_k^2 + \|\vec{g}\|^2/\lambda + (\tau^2/M + A^2)/\lambda\sigma^2, k \in [1..d]$. Hence, the variance is small either the population size is large or $\|\vec{g}\|$ is small, which happens when CMA-ES approaches a local minimum ($\vec{g} = 0$). For the sake of numerical stability, we exponentially smooth $\vec{g}^*$ values in the past: $\vec{g} \leftarrow (1-\alpha)\vec{g} + \alpha\vec{g}^*, \alpha \in (0,1)$.

Time complexity: Our method incurs small time complexity in addition to the standard CMA-ES: Eq. (16) only involves a constant number of arithmetic operations; the largest eigenvalue $s_{\max}$ of C is provided internally by the standard CMA-ES. It takes $O(\lambda)$ time to estimate $\vec{g}$ and takes $O(1)$ to estimate the noise level τ since the latter is only executed once. The Lipschitz estimation takes $O(\lambda^3)$ time to fit the Gaussian process and $O(\lambda d^2)$ to compute $\|H(\vec{x})\|_1$ (the Hessian of the posterior mean function). Since, in practice, the population size is small - typically $\lambda \in \Theta(\log d)$, the actual CPU time used in Lipschitz estimation is marginal.

4 RELATED WORKS

Three-Stage CMA-ES:. The authors in [7] propose a static schedule that divides the optimization process into three distinct stages, with the number of re-evaluations increasing ten-fold at each stage. For example, with a budget of 10^7 function re-evaluations, the method allocates $M_1 = 100, M_2 = 1\,000, M_3 = 10\,000$, and keeps a fixed ratio of 10:3:1 among the total function evaluations in three stages. Such a setup results in evaluations of approximately 7 150, 2 145, and 715 candidates at each stage, respectively. Despite its simplicity, this method has been shown to work well on quantum chemistry problems [5, 7]. However, this method may not be as effective for other problems, as the fixed number of re-evaluations might either fall short or be excessive, potentially slowing down the convergence rate of CMA-ES.

Uncertainty handling CMA-ES:. The Uncertainty handling CMA-ES(UH-CMA-ES) introduced in ref. [15] presents an adaptive strategy that increases the re-evaluation number M if significant ranking changes occur for some candidates when their noisy function values are recomputed with the current M. Specifically, after evaluating each point in the population $\{\vec{x}^{\,i}\}_i$ with M re-evaluations, a random sub-population is selected to re-estimate the function values. The entire population is then reordered based on these updated noisy values, and the ranking changes for each $\vec{x}^{\,i}$ are compared before and after re-estimation. UH-CMA-ES aggregates these rank changes across all candidates to determine whether M should be adjusted. If the indicator is positive, M is increased multiplicatively; otherwise, it stays the same.

Population Size Adaptation CMA-ES:. The population size adaptation in CMA-ES builds on the Information Geometric Optimization (IGO) framework [22], where population size is treated as the number of Monte Carlo samples used to estimate the natural gradient. This approach reduces gradient variability in multimodal and noisy functions compared to unimodal ones.

Applied to CMA-ES, population size adjusts based on the length of the evolution path [23]. A normalization factor ensures accurate parameter update assessment, while step-size adjustment maintains stability during population changes. When updates lack precision, population size increases to enhance exploration; when precision is sufficient, it decreases to focus on exploitation. This dynamic balance between exploration and exploitation improves performance in noisy and multimodal environments.

Learning Rate Adaptation CMA-ES:. The so-called Learning Rate Adaptation CMA-Es (LRA-CMA-ES) presented in ref. [25] introduces a dynamic adjustment of the learning rates (η_m^t and η_{C}^t) on a per-iteration basis. Effectively, such adaptation translates into tuning the updates Δ_m^t and Δ_{C}^t. As such, the updating rules of the center of mass and the covariance matrix are $m^{t+1} = m^t + \eta_m^t \Delta_m^t$ and $\mathrm{C}^{t+1} = \mathrm{C}^t + \eta_{\mathrm{C}}^t \Delta_{\mathrm{C}}^t$. It estimates the signal-to-noise ratio as the fraction between the expected value of the updating vector and its variance. The adaptive learning rate mechanism seeks to maintain a constant signal-to-noise ratio (SNR) provided as a hyperparameter. Thus, when the empirical SNR is higher than the provided constant, the learning rate is increased, and when it is lower, the learning rate is reduced.

5 EXPERIMENTS

Experiments setup: We make an empirical comparison of AR-CMA-ES against the most advanced methods: UH-CMA-ES, Three-Stage CMA-ES, PSA-CMA-ES, and LRA-CMA-ES. We thoroughly re-implement them by integrating their original source code with the modular CMA-ES [9] framework, also considering the details in the original publication to the best of our ability[1].

For the objective functions, we choose ten standard artificial test functions (see Table 3 in the Appendix D for their definition). These test functions encompass a wide range of landscapes, such as unimodal/multi-modal landscapes and dimension-separable and non-separable properties, which are considered difficult for numerical optimization. To gather statistically relevant data, we will execute 20 independent runs for each test function. Additionally, we add artificial noise in three levels: $\tau^2 \in \{1, 10, 100\}$. To make the comparison as fair as possible, we use the same population size of CMA-ES, $\mu = 50, \lambda = 100$, for all methods; the initial step size is set to $\sigma_0 = 0.1 \times \|\vec{x}_U - \vec{x}_L\|_\infty$, where $[\vec{x}_L, \vec{x}_U] \subset \mathbb{R}^d$ is the search space (see Table 3 in the appendix for the search space of each function). For the methods we compare, we leave their remaining hyperparameter settings unchanged from the original publication.

To determine the coefficients α and β used in exponential smoothing for our method, we extensively test various combinations of them, which results in setting $\alpha = 0.1$ and $\beta = 0.1$. For the value of A in Eq. (4), we choose the smallest measured $\Delta\bar{\mathcal{L}}$ value among all candidates in each iteration.

Finally, we test all methods with different budgets of function evaluations, where we recap the re-evaluation number per candidate at 1% of the total budget.

Results: First, since we modified CMA-ES's default recombination weights in Eq. 4, we verify the performance of AR-CMA-ES against CMA-ES in the noiseless ($\tau = 0$) and noisy scenarios ($\tau = 1$). In Table 1, we see that in the noiseless cases, AR-CMA-ES is quite comparable to CMA-ES while under the additive noise ($\tau = 1$), AR-CMA-ES significantly improves CMA-ES. For the noise-free case, we used a fixed budget of 10^4 function re-evaluations for each function for both AR-CMA-ES and CMA-ES. An exception was made for the Trid function, which required a larger budget of 10^5 re-evaluations to be successfully solved. For the noisy case, we allocated a significantly larger budget of 10^9 function re-evaluations. To ensure a fair comparison, we also allowed standard CMA-ES to re-evaluate each candidate for 10^4, as it will eventually reach a plateau for any input given candidate no of re-evaluations.

[1]The source code can be accessed at
https://anonymous.4open.science/r/ShotFrugal-7CD4

Table 1: Numerical verification of AR-CMA-ES against CMA-ES for $d = 20$ on noiseless test functions $\tau^2 = 0$ and functions with low noise levels $\tau^2 = 1$. For each algorithm, we report the mean and standard error of the final noiseless precision achieved over 20 runs. Different evaluation budgets were used for noisy and noiseless settings–please refer to the Results section for further details.

Problem	AR-CMA-ES ($\tau^2 = 0$)	CMA-ES ($\tau^2 = 0$)	AR-CMA-ES ($\tau^2 = 1$)	CMA-ES ($\tau^2 = 1$)
Sphere	$1.24 \times 10^{-5} \pm 6.86 \times 10^{-6}$	$1.80 \times 10^{-6} \pm 1.28 \times 10^{-6}$	$1.06 \times 10^{-7} \pm 5.63 \times 10^{-8}$	$3.40 \times 10^{-5} \pm 2.01 \times 10^{-5}$
Ellipsoid	$1.31 \times 10^{-3} \pm 8.09 \times 10^{-4}$	$4.03 \times 10^{-4} \pm 3.21 \times 10^{-4}$	$9.84 \times 10^{-8} \pm 5.85 \times 10^{-8}$	$6.53 \times 10^{-6} \pm 3.63 \times 10^{-6}$
HyperEllipsoid	$3.11 \times 10^{-4} \pm 2.52 \times 10^{-4}$	$3.84 \times 10^{-5} \pm 3.21 \times 10^{-5}$	$8.36 \times 10^{-8} \pm 3.56 \times 10^{-8}$	$5.95 \times 10^{-6} \pm 2.44 \times 10^{-6}$
RotatedEllipsoid	$1.60 \times 10^{-3} \pm 1.32 \times 10^{-3}$	$5.40 \times 10^{-4} \pm 3.82 \times 10^{-4}$	$9.21 \times 10^{-8} \pm 4.74 \times 10^{-8}$	$6.63 \times 10^{-6} \pm 4.00 \times 10^{-6}$
RotatedHyperEllipsoid	$3.27 \times 10^{-4} \pm 2.59 \times 10^{-4}$	$5.63 \times 10^{-5} \pm 3.85 \times 10^{-5}$	$8.75 \times 10^{-8} \pm 3.71 \times 10^{-8}$	$5.42 \times 10^{-6} \pm 2.14 \times 10^{-6}$
Rastingrin	$1.53 \times 10^{2} \pm 4.65 \times 10^{1}$	$1.59 \times 10^{2} \pm 5.59 \times 10^{1}$	1.24 ± 0.85	72.10 ± 38.80
Trid	$1.46 \times 10^{-11} \pm 1.49 \times 10^{-11}$	$1.46 \times 10^{-11} \pm 1.77 \times 10^{-11}$	$1.46 \times 10^{-7} \pm 8.11 \times 10^{-8}$	$2.74 \times 10^{-5} \pm 1.18 \times 10^{-5}$
Bohachevsky	$2.29 \times 10^{-2} \pm 2.44 \times 10^{-2}$	$3.89 \times 10^{-3} \pm 4.05 \times 10^{-3}$	$2.30 \times 10^{-7} \pm 3.71 \times 10^{-7}$	$6.15 \times 10^{-6} \pm 3.02 \times 10^{-6}$
CosineMixture	$9.14 \times 10^{-7} \pm 1.27 \times 10^{-6}$	$1.56 \times 10^{-7} \pm 1.49 \times 10^{-7}$	$9.25 \times 10^{-8} \pm 4.15 \times 10^{-8}$	$1.98 \times 10^{-5} \pm 5.82 \times 10^{-5}$
Schwefel02	$3.96 \times 10^{-3} \pm 2.95 \times 10^{-3}$	$1.68 \times 10^{-3} \pm 1.41 \times 10^{-3}$	$7.60 \times 10^{-8} \pm 2.70 \times 10^{-8}$	$4.00 \times 10^{-3} \pm 1.79 \times 10^{-2}$

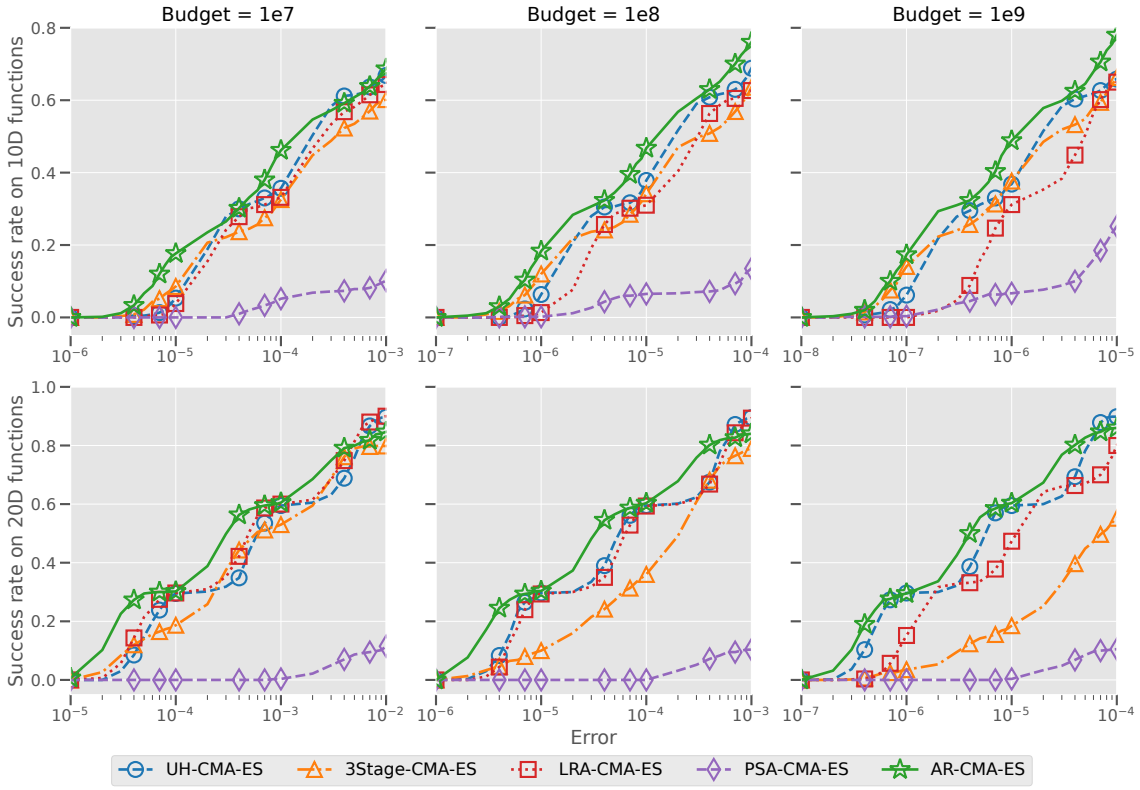


Figure 2: Empirical cumulative distribution functions (ECDFs) of the error ($\mathcal{L}(\vec{m}) - \mathcal{L}^*$) obtained from all 20 independent runs, three noise levels, and on all test functions. Top row: $d = 10$; bottom: $d = 20$. Three columns from left to right correspond to an evaluation budget of 10^7, 10^8, and 10^9, respectively.

Second, we record the trajectory of the center of mass $\vec{m}$ and compute the corresponding noiseless function values $\mathcal{L}(\vec{m})$. Then, we compute the empirical cumulative distribution function (ECDF) of the optimization error $\mathcal{L}(\vec{m}) - \mathcal{L}^*$ upon the termination of each method ($\mathcal{L}^*$ denotes the global optimal) for each combination of $d \in \{10, 20\}$ and evaluation budget in $\{10^7, 10^8, 10^9\}$. Formally, ECDF of an algorithm is defined as $\text{ECDF}(x) = \sum_{i=1}^{N} \mathbb{1}_{[e_i,\infty)}(x)/N$, where e_i is the optimization error observed in the i-th run. We show the main ECDF curves in Fig. 2, which aggregates over all functions and noise levels. Also, we included, in the appendix, the ECDFs on each function and noise level (Fig. 5 and 6).

Table 2: Wall-clock time comparison for a budget of 10^7 re-evaluations on $d = 20$, averaged over 20 runs (unit: secs) using an Intel Core i7-9750H processor (6 CPU cores) and 32GB RAM.

Problem	AR-CMA-ES	CMA-ES	3-Stage-CMA-ES	UH-CMA-ES	LRA-CMA-ES	PSA-CMA-ES
Sphere	16.2 ± 0.44	22.7 ± 0.80	17.8 ± 0.83	19.6 ± 0.81	24.6 ± 2.78	25.4 ± 3.65
Ellipsoid	16.1 ± 0.31	29.4 ± 3.76	18.5 ± 1.85	21.1 ± 1.69	41.7 ± 2.08	36.9 ± 4.66
HyperEllipsoid	16.0 ± 0.00	28.4 ± 4.52	17.9 ± 0.45	20.2 ± 0.95	34.9 ± 2.10	29.8 ± 5.60
RotatedEllipsoid	16.0 ± 0.00	37.0 ± 4.37	18.2 ± 0.55	21.1 ± 1.07	46.1 ± 4.38	38.2 ± 4.30
RotatedHyperEllipsoid	16.0 ± 0.00	33.0 ± 1.57	17.4 ± 0.49	20.2 ± 0.89	34.5 ± 3.30	29.1 ± 2.55
Rastingin	16.6 ± 0.69	32.9 ± 3.30	18.1 ± 1.43	20.8 ± 1.67	34.9 ± 3.48	32.7 ± 3.25
Trid	18.0 ± 0.32	23.5 ± 0.51	18.0 ± 0.56	23.2 ± 1.07	34.6 ± 1.05	27.6 ± 0.82
CosineMixture	15.8 ± 0.37	30.4 ± 3.60	18.1 ± 0.72	17.1 ± 0.51	37.0 ± 4.28	32.3 ± 9.40
Bohachevsky	17.4 ± 0.51	38.4 ± 1.14	19.1 ± 0.45	22.7 ± 2.34	52.8 ± 3.35	45.2 ± 7.58
Schwefel	16.5 ± 0.51	26.9 ± 3.25	17.6 ± 0.60	20.3 ± 2.18	33.4 ± 1.35	26.9 ± 3.98

As we increase the budget and function dimension, and hence the hardness of the optimization task, AR-CMA-ES shows a substantial performance improvement compared to all other methods. Particularly for relatively higher dimensions ($d = 20$), we pointed out that the major benefit of our method lies in increasing the probability of hitting difficult error values quite a bit. As an example, with $d = 20$ and a budget of 10^7 function evaluations, AR-CMA-ES can reach an optimization error $\leq 4 \times 10^{-5}$ with approximately 27% probability. In contrast, for all other methods, the probability drops drastically, UH-CMA-ES: 9%, Three-Stage-CMA-ES: 12%, LRA-CMA-ES: 14%, and PSA-CMA-ES: 0%. With a higher budget of 10^9 function evaluations and $d = 20$, we observe a similar result; as such, our method found around 19% of solutions with an optimization error $\leq 4 \times 10^{-7}$, while UH-CMA-ES achieved only 10% and the other methods failed to achieve such threshold. However, we can observe two convergence points for all three budgets where several methods achieve a similar probability of success. With a budget 10^9 and $d = 20$, we observe that AR-CMA-ES and UH-CMA-ES achieve so probability of success at a precision of 10^{-6} (around 30%) and at a precision of 10^{-5} (around 60%). However, our method still shows a significantly higher cumulative probability at almost all error values. To see the effect of the noise level on the performance, we show in Fig. 4 (in Appendix D) the ECDF curves for each combination of dimensions, budgets, and noise levels. As the noise level increases, performance slightly decreases. This behavior is due to overestimation of M^*, as the number of function re-evaluations is linearly dependent on the noise.

For closer analysis, we showcase the ECDF and empirical convergence curve on the Trid function, which is a non-separable function across dimensions, making it a challenging problem for algorithms.

Fig. 3 (top) shows the ECDF on a 20-dimensional Trid function with a budget of 10^9 function evaluations and different noise levels ($\tau^2 \in \{1, 10, 100\}$). As discussed, the Trid function is non-separable across dimensions (the minimum cannot be found by searching along each dimension separately). We observe that AR-CMA-ES achieves substantial improvement compared to all other methods, while Three-Stage and PSA-CMA-ES failed to hit any small error value, indicated by their flat ECDF curve. In Fig. 3 (bottom), we draw the convergence curves - $\log_{10}(\mathcal{L}(\vec{m}) - \mathcal{L}^*)$ as a function of function evaluations. We see that AR-CMA-ES delivers a significantly steeper convergence than UH- and LRA-CMA-ES.

Performance on non-Lipschitz continuous functions. We also perform experiments with AR-CMA-ES on four ten-dimensional benchmark functions that are not Lipschitz continuous (see Table 4) to assess how its performance is affected when its core theoretical assumption does not hold. First, we compare the performance of AR-CMA-ES to CMA-ES in a noiseless setting, as shown in Table 5. The results indicate that AR-CMA-ES performs comparably to CMA-ES, with no statistically significant difference on most problems–except for the Sum Absolute function, where its performance is off by an order of magnitude. Next, we evaluate all relevant algorithms under a moderate noise level of $\tau^2 = 1$. As summarized in Table 4, AR-CMA-ES consistently outperforms CMA-ES on all but one benchmark–the non-smooth version of the Griewank function–where its performance drops by two orders of magnitude. Upon further analysis, we found this discrepancy to stem from inaccurate estimation of the Lipschitz constant, which is, in fact, ill-defined for this function. Compared to other algorithms, AR-CMA-ES ranks second on the Sum Absolute function (slightly behind UH-CMA-ES) and shows statistically indifferent results on the Nesterov F1 and F2 benchmarks. These results suggest that while AR-CMA-ES is designed with Lipschitz continuity in mind, it remains competitive even for non-smooth benchmark functions. Please refer to Appendix E for more details.

6 CONCLUSION

In this paper, we propose AR-CMA-ES, a novel noise-handling method for the famous CMA-ES algorithm under additive Gaussian white noise. We consider the expected improvement of the noiseless function value in one iteration of CMA-ES and derive a lower bound on it, provided the noise level and the Lipschitz constant of the function's gradient. Normalizing the lower bound by the re-evaluation number gives us an efficiency metric. Solving for the maximum efficiency, we obtain a simple expression of the optimal re-evaluation number.

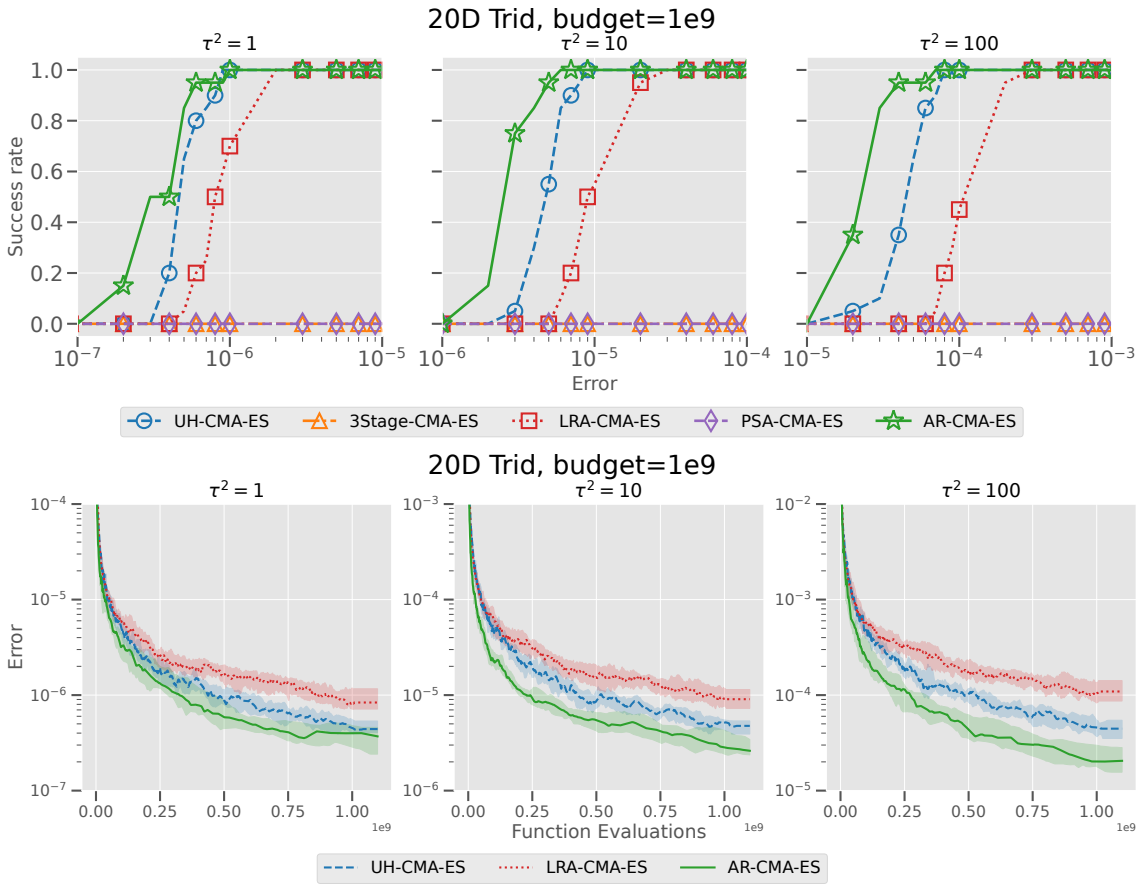


Figure 3: Top: On 20-dimensional Trid function (unimodal and non-separable), the empirical cumulative distribution function (ECDF) of the error ($\mathcal{L}(\vec{m}) - \mathcal{L}^*$) obtained with 10^9 function evaluation budget for three different noise levels ($\tau^2 \in \{1, 10, 100\}$) separately. Bottom: Mean convergence curve - $\log_{10}(\mathcal{L}(\vec{m}) - \mathcal{L}^*)$ as a function of function evaluations. We see AR-CMA-ES outperforms other methods, and it is more advantageous when the noise τ gets larger. We observe that both Three-Stage and PSA-CMA-ES completely failed on this function.

This adaptive strategy enhances CMA-ES's performance by efficiently allocating function (re)-evaluation without significant computational overheads. AR-CMA-ES substantially outperforms several state-of-the-art noise-handling methods for CMA-ES and demonstrates a consistent advantage across different test functions, search dimensions, and noise levels. While AR-CMA-ES demonstrates significant improvements in handling additive noise, it exhibits the following limitations:

- **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 multiplicative noise or other forms of complex noise patterns, the derived expression might not hold any longer. Further research is needed to extend the method to handle a broader range of noise types effectively.
- **Impact of noise level**: The number of function re-evaluations in AR-CMA-ES is linearly dependent on the noise level τ. As the noise level increases, this dependency can lead to a huge re-evaluation number, which might not be the best choice in high-noise environments.
- **Limited empirical validation**: While AR-CMA-ES demonstrates performance benefits on artificial test 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.

For future works, we will focus on addressing the above limitations and testing them on real-world optimization problems.

REFERENCES

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

[2] Akiko N. Aizawa and Benjamin W. Wah. 1994. Scheduling of Genetic Algorithms in a Noisy Environment. *Evolutionary Computation* 2, 2 (1994), 97–122. https://doi.org/10.1162/evco.1994.2.2.97

[3] Dirk V Arnold. 2002. *Noisy optimization with evolution strategies*. Vol. 8. Springer Science & Business Media.

[4] Hans-Georg Beyer, Markus Olhofer, and Bernhard Sendhoff. 2004. On the Impact of Systematic Noise on the Evolutionary Optimization Performance—A Sphere Model Analysis. *Genetic Programming and Evolvable Machines* 5, 4 (2004), 327–360. https://doi.org/10.1023/B:GENP.0000036020.79188.a0

[5] Xavier Bonet-Monroig, Hao Wang, Diederick Vermetten, Bruno Senjean, Charles Moussa, Thomas Bäck, Vedran Dunjko, and Thomas E O'Brien. 2023. Performance comparison of optimization methods on variational quantum algorithms. *Physical Review A* 107, 3 (2023), 032407.

[6] Torsten F Bosse and H Martin Bücker. 2024. A piecewise smooth version of the Griewank function. *Optimization Methods and Software* (2024), 1–11.

[7] Chris Cade, Lana Mineh, Ashley Montanaro, and Stasja Stanisic. 2020. Strategies for solving the Fermi-Hubbard model on near-term quantum computers. *Phys. Rev. B* 102 (Dec 2020), 235122. Issue 23. https://doi.org/10.1103/PhysRevB.102.235122

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

[9] Jacob de Nobel, Diederick Vermetten, Hao Wang, Carola Doerr, and Thomas Bäck. 2021. Tuning as a means of assessing the benefits of new ideas in interplay with existing algorithmic modules. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 1375–1384.

[10] Michael Emmerich, Ofer M. Shir, and Hao Wang. 2018. Evolution Strategies. In *Handbook of Heuristics*, Rafael Martí, Panos M. Pardalos, and Mauricio G. C. Resende (Eds.). Springer, 89–119. https://doi.org/10.1007/978-3-319-07124-4_13

[11] Javier González, Zhenwen Dai, Philipp Hennig, and Neil D. Lawrence. 2016. Batch Bayesian Optimization via Local Penalization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016 (JMLR Workshop and Conference Proceedings, Vol. 51)*, Arthur Gretton and Christian C. Robert (Eds.). JMLR.org, 648–657. http://proceedings.mlr.press/v51/gonzalez16a.html

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

[13] Mert Gürbüzbalaban and Michael L Overton. 2012. On Nesterov's nonsmooth Chebyshev–Rosenbrock functions. *Nonlinear Analysis: Theory, Methods & Applications* 75, 3 (2012), 1282–1289.

[14] Nikolaus Hansen. 2016. The CMA Evolution Strategy: A Tutorial. *CoRR* abs/1604.00772 (2016). arXiv:1604.00772 http://arxiv.org/abs/1604.00772

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

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

[17] Julien Walden Huang, Stephen J. Roberts, and Jan-Peter Calliess. 2023. On the Sample Complexity of Lipschitz Constant Estimation. *Trans. Mach. Learn. Res.* 2023 (2023). https://openreview.net/forum?id=UIalYAHdBH

[18] Mykel J Kochenderfer and Tim A Wheeler. 2019. *Algorithms for optimization*. MIT Press.

[19] Zhenhua Li, Shuo Zhang, Xinye Cai, Qingfu Zhang, Xiaomin Zhu, Zhun Fan, and Xiuyi Jia. 2022. Noisy Optimization by Evolution Strategies With Online Population Size Learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52, 9 (2022), 5816–5828. https://doi.org/10.1109/TSMC.2021.3131482

[20] Ilya Loshchilov and Frank Hutter. 2016. CMA-ES for Hyperparameter Optimization of Deep Neural Networks. *CoRR* abs/1604.07269 (2016). arXiv:1604.07269 http://arxiv.org/abs/1604.07269

[21] Joaquim RRA Martins and Andrew Ning. 2021. *Engineering design optimization*. Cambridge University Press.

[22] Kouhei Nishida and Youhei Akimoto. 2016. Population size adaptation for the CMA-ES based on the estimation accuracy of the natural gradient. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. 237–244.

[23] Kouhei Nishida and Youhei Akimoto. 2018. PSA-CMA-ES: CMA-ES with population size adaptation. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018, Kyoto, Japan, July 15-19, 2018*, Hernán E. Aguirre and Keiki Takadama (Eds.). ACM, 865–872. https://doi.org/10.1145/3205455.3205467

[24] V. Nissen and J. Propach. 1998. On the robustness of population-based versus point-based optimization in the presence of noise. *IEEE Transactions on Evolutionary Computation* 2, 3 (1998), 107–119. https://doi.org/10.1109/4235.735433

[25] Masahiro Nomura, Youhei Akimoto, and Isao Ono. 2023. 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*. 839–847.

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

[27] Jonathan E Rowe et al. 2021. Evolutionary Algorithms for Solving Unconstrained, Constrained and Multi-objective Noisy Combinatorial Optimisation Problems. *arXiv preprint arXiv:2110.02288* (2021).

[28] Tim Salimans, Jonathan Ho, Xi Chen, and Ilya Sutskever. 2017. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. *CoRR* abs/1703.03864 (2017). arXiv:1703.03864 http://arxiv.org/abs/1703.03864

[29] Ofer M. Shir, Jonathan Roslund, Zaki Leghtas, and Herschel Rabitz. 2012. Quantum control experiments as a testbed for evolutionary multi-objective algorithms. *Genetic Programming and Evolvable Machines* 13 (2012), 445–491. Issue 4. https://doi.org/10.1007/s10710-012-9164-7

[30] Roman G. Strongin, Konstantin Barkalov, and Semen Bevzuk. 2019. Acceleration of Global Search by Implementing Dual Estimates for Lipschitz Constant. In *Numerical Computations: Theory and Algorithms - Third International Conference, NUMTA 2019, Crotone, Italy, June 15-21, 2019, Revised Selected Papers, Part II (Lecture Notes in Computer Science, Vol. 11974)*, Yaroslav D. Sergeyev and Dmitri E. Kvasov (Eds.). Springer, 478–486. https://doi.org/10.1007/978-3-030-40616-5_46

[31] Kento Uchida, Kenta Nishihara, and Shinichi Shirakawa. 2024. CMA-ES with Adaptive Reevaluation for Multiplicative Noise. *arXiv preprint arXiv:2405.11471* (2024).

[32] Konstantinos Varelas, Anne Auger, Dimo Brockhoff, Nikolaus Hansen, Ouassim Ait ElHara, Yann Semet, Rami Kassab, and Frédéric Barbaresco. 2018. 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*. Springer, 3–15.