論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Learning Rate Adaptation for Evolution Strategies
著者(和文)	野村将寛
Author(English)	Masahiro Nomura
出典(和文)	学位:博士(工学), 学位授与機関:東京科学大学, 報告番号:甲第366号, 授与年月日:2025年3月26日, 学位の種別:課程博士, 審査員:小野 功,三宅 美博,山村 雅幸,瀧ノ上 正浩,小野 峻佑
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Institute of Science Tokyo, Report number:甲第366号, Conferred date:2025/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Doctoral Dissertation

Learning Rate Adaptation for Evolution Strategies

Institute of Science Tokyo

Masahiro Nomura

March 2025

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Isao Ono, for his inspiring guidance and unwavering support. From the very beginning, he has shown remarkable patience and provided thoughtful advice to help me grow as a scientist. The encouraging environment he cultivated inspired me to pursue a research path, and his invaluable insights have profoundly shaped my development both as a researcher and as an individual.

I would also like to extend my heartfelt thanks to Associate Professor Youhei Akimoto for his invaluable contributions to this project. His expert advice provided me with significant insights into conducting high-quality research. I feel truly fortunate to have had the opportunity to collaborate with him.

I would like to thank Professor Yoshihiro Miyake, Professor Masahiro Takinoue, Professor Masayuki Yamamura, and Associate Professor Shunsuke Ono for their help as members of the review committee for my doctoral dissertation.

I am grateful to Ryoki Hamano, Shota Saito, Professor Shinichi Shirakawa, and Assistant Professor Kento Uchida for their thought-provoking discussions on evolution strategies. I would like to extend special thanks to Yuta Saito for his inspiration on writing high-quality papers and to Masashi Shibata for his valuable advice on improving software quality. I thank the members of the Ono Laboratory for their comments and suggestions, especially Koki Ikeda and Kosuke Ujihara for their support during my time in the lab.

Finally, I want to thank my family for their steadfast support.

Abstract

Black-box continuous optimization is widely used for real-world problems because it does not require explicit information about the objective function (e.g., gradients or differentiability). This flexibility makes it suitable for complex problems where solution evaluation involves expensive numerical simulations. As a result, techniques to accelerate the optimization process, such as parallel evaluations, are important in practice. Among the factors that make black-box continuous optimization challenging, multimodality and noise are particularly significant.

Evolution strategies (ES) is a promising framework for black-box continuous optimization problems. ES employs a multivariate Gaussian distribution, parameterized by a mean vector and covariance matrix, to guide optimization. In each iteration, ES generates multiple candidate solutions, with the population size indicating the number of solutions generated. These candidate solutions are evaluated on the objective function and ranked. ES then updates the distribution parameters based on this ranking and a learning rate. This process repeats until a predefined stopping criterion is met. The objective function value and distribution parameters are used to define the stopping criterion. For example, the algorithm may stop when the best evaluation value fails to improve over a give period, or when the covariance matrix sufficiently shrinks (indicating convergence) or expands (indicating divergence). Prominent methods within ES include exponential natural evolution strategies (xNES) and covariance matrix adaptation evolution strategy (CMA-ES), both of which are partially explained by the information geometric optimization (IGO) framework.

A key practical issue for ES is determining the appropriate population size. While a smaller population size generally performs well for many unimodal problems, increasing the population size can be beneficial for more difficult tasks, such as those involving multimodal landscapes and additive noise. However, in a black-box scenario, understanding the problem structure of the objective func-

tion is challenging, which makes selecting the appropriate population size in advance is also challenging. To address this, online adaptation of the population size has been proposed to address the issue.

It has been observed that, in CMA-ES, increasing the population size has a similar effect to decreasing the learning rate for the mean vector. Inspired by this observation, this study focuses on learning rate adaptation rather than population size adaptation. We argue that the learning rate adaptation is more advantageous than the population size adaptation from a practical perspective, as the former is better suited for parallel implementations. For example, practitioners often want to set the population size to match a specific number of workers to avoid wasting computational resources. However, the population size adaptation may not always utilize resources efficiently, as the population size can fluctuate throughout the optimization process. In contrast, learning rate adaptation enables full exploitation of resources by fixing the population size at the maximum number of workers. Furthermore, with learning rate adaptation, parameter updates occur regularly, whereas ES with population size adaptation does not progress until all evaluations are complete, complicating the determination of an appropriate termination point.

This study consists of two works. In the first work, we propose a learning rate adaptation method aimed at accelerating optimization, using xNES as the optimization method. With a population size sufficient to solve the problem, the proposed method measures tendencies of updates in the distribution parameters and accelerates the optimization by increasing the learning rate appropriately when a sufficient tendency is detected. Our method enables a larger learning rate for relatively easy problems, resulting in faster search. Conversely, for more difficult problems (e.g., multimodal problems), it allows for a conservative learning rate, leading to a robust and stable search. Experimental evaluations on both unimodal and multimodal problems demonstrate that the proposed method works properly depending on a search situation and is effective over existing methods, such as those using a fixed learning rate.

In the second work, we propose a learning rate adaptation method for solving difficult tasks, such as multimodal or noisy problems, using CMA-ES as the optimization method. Unlike the first work, the second work accepts any population size without assuming it is sufficiently large. This study comprehensively explores the learning rate impact on IGO to demonstrate the necessity of a small learning rate by considering ordinary differential equations. Thereafter, it discusses the setting of an ideal learning rate. Based on these discussions, we develop a novel learning rate adaptation mechanism for CMA-ES that maintains a

constant signal-to-noise ratio. Additionally, we investigate the behavior of CMA-ES with the proposed learning rate adaptation mechanism through numerical experiments and compare the results with those obtained for CMA-ES with a fixed learning rate and with population size adaptation. The results show that CMA-ES with the proposed learning rate adaptation works well for multimodal and/or noisy problems without extremely expensive learning rate tuning.

While the proposed method in the second work was designed to solve difficult problems safely, we believe it can also be extended to accelerate optimization, aligning with the goals of the first work. This study marks an important step toward developing fully hyperparameter-free ES algorithms for general-purpose optimization.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Black-box continuous optimization is widely used for real-world problems because it does not require explicit information about the objective function $f(x)$ (e.g., gradients or differentiability). This flexibility makes it suitable for complex problems where solution evaluation involves expensive numerical simulations. As a result, techniques to accelerate the optimization process, such as parallel evaluations, are important in practice. Among the factors that make black-box continuous optimization challenging, multimodality and noise are particularly significant. Multimodality refers to the presence of multiple local optima, making it important to identify the global optimum. In this work, we focus on well-structured multimodal problems, which resemble unimodal functions with added (deterministic) noise, making them relatively easy to solve. In noisy problems, the observed function value is stochastically perturbed: $y = f(x) + \epsilon$, where y is the observed value and ϵ represents noise.

Evolution strategies (ES) [76, 83, 19, 34] forms a promising framework for black-box continuous optimization problems. ES employs a multivariate Gaussian distribution with parameters consisting of a mean vector m and a covariance matrix Σ to guide the optimization process. In each iteration, ES generates λ candidate solutions–where λ, known as the population size, defines the number of samples–from the distribution. These candidate solutions are evaluated on the objective function, and their rankings are determined accordingly. Notably, this evaluation step can be performed in parallel, highlighting the high parallelizability of ES. This capability is particularly advantageous for black-box continuous

optimization, where evaluating candidate solutions is often computationally expensive and time-consuming. Based on this ranking, ES updates the distribution parameters using the learning rate η: $m = m + \eta_m \Delta_m$ and $\Sigma = \Sigma + \eta_\Sigma \Delta_\Sigma$, where η_m and η_Σ are the learning rates for m and Σ, respectively, and Δ_m and Δ_Σ are their respective updates. This procedure is repeated until a predefined stopping criterion is met. The objective function value and distribution parameters are used to define the stopping criterion. For example, the algorithm may stop when the best evaluation value fails to improve over a give period, or when the covariance matrix sufficiently shrinks (indicating convergence) or expands (indicating divergence). Prominent methods within ES include exponential natural evolution strategies (xNES) and covariance matrix adaptation evolution strategy (CMA-ES), both of which are partially explained by the information geometric optimization (IGO) framework.

A key practical issue for ES is determining the appropriate population size λ. While a smaller λ generally performs well for many unimodal problems, increasing λ can be beneficial for more difficult tasks, such as those involving multimodal landscapes and additive noise. However, in a black-box scenario, understanding the problem structure of f is challenging, which makes selecting the appropriate λ value in advance is also challenging. To address this, online adaptation of λ has been proposed to address the issue [62, 64, 41, 61]. One prominent adaptation mechanism is population size adaptation (PSA)-CMA-ES [64], which has demonstrated exhibited promising performance on difficult tasks, including those with multimodal and additive noise characteristics.

It has been observed that, in CMA-ES, increasing λ has a similar effect to decreasing the learning rate η_m for m [59]. Inspired by this observation, this study focuses on learning rate η adaptation rather than population size λ adaptation. We argue that η adaptation is more advantageous than λ adaptation from a practical perspective, as the former is better suited for parallel implementations. For example, practitioners often want to set λ to match a specific number of workers to avoid wasting computational resources. However, λ adaptation may not always utilize resources efficiently, as λ values can fluctuate throughout the optimization process. In contrast, η adaptation enables full exploitation of resources by fixing λ at the maximum number of workers. Furthermore, with η adaptation, parameter updates occur regularly, whereas CMA-ES with λ adaptation does not progress until all λ evaluations are complete, complicating the determination of an appropriate termination point.

Table 1.1: Goals and requirements for λ in the first work (Chapter 3) and the second work (Chapter 4).

	Goal	Requirement for λ
Chapter 3	Acceleration	Large λ
Chapter 4	Multimodal & Noise	Any λ

1.2 Contributions

This study consists of two works. In the first work, we propose a learning rate adaptation method aimed at accelerating optimization, utilizing xNES as the optimization method. Given a population size λ sufficient to solve the problem, the proposed method assesses whether the learning rate should be increased by measuring tendencies in the updates of distribution parameter updates. When a sufficient tendency is identified, the method accelerates optimization by appropriately increasing the learning rate η_Σ. The tendency measurement is inspired by the approach used in the population size adaptation of CMA-ES [64]. In this work, the mean vector learning rate η_m is fixed at the default value of $\eta_m = 1$ in xNES, as it is already large and widely used in the ES literature. Our method enables a larger η_Σ for relatively easy problems, resulting in faster search. Conversely, for more difficult problems (e.g., multimodal problems), it allows for a conservative η_Σ, leading to a robust and stable search. Experimental evaluations on both unimodal and multimodal problems demonstrate that the proposed method works properly depending on a search situation and is effective over existing methods, such as those using a fixed learning rate.

In the second work, we propose a learning rate adaptation method for solving *difficult* tasks, such as multimodal or noisy problems, using CMA-ES as the optimization method. While the first work emphasized strategies for increasing the learning rate, the second focuses on methods for decreasing it. To demonstrate the necessity of a small learning rate, this study comprehensively explores the η impact on IGO by considering ordinary differential equations. However, simply setting a small learning rate can compromise the efficiency of optimization. To address this, the work explores the ideal choice of η, offering insights into the rational design of learning rates to enhance efficiency. Based on these discussions, we develop a novel η adaptation mechanism for CMA-ES that maintains a constant signal-to-noise ratio. Additionally, we investigate the behavior of CMA-ES with the proposed η adaptation mechanism through numerical experiments and

compare the results with those obtained for CMA-ES with a fixed η and with population size adaptation. The results show that CMA-ES with the proposed η adaptation works well for multimodal and/or noisy problems *without* extremely expensive η tuning. Importantly, while the first work requires determining a sufficiently large population size λ to solve the problem—particularly for multimodal tasks—the second study eliminates the need to predefine this value. This advancement significantly reduces the time and effort involved in hyperparameter tuning, making the approach more practical. Table 1.1 shows the summary of the goals and requirements for λ in the first and second works.

While the proposed method in the second work was originally designed to safely address difficult tasks, such as multimodal and noisy optimization problems, a simple modification enables it to be extended for acceleration purposes, aligning closely with the goals of the first work. Notably, the method's inherent flexibility makes it highly extensible, allowing for applications to a wide range of optimization algorithms. We strongly believe that this study represents a significant milestone toward the development of fully hyperparameter-free ES algorithms suitable for general-purpose optimization tasks.

1.3 Organization of This Paper

Chapter 2 presents the background knowledge necessary for this study. We begin by describing black-box continuous optimization, the problem setting of this study. Next, we focus on multimodality and additive noise—specific characteristics that make black-box continuous optimization particularly challenging. Finally, we introduce two prominent black-box continuous optimization methods, xNES and CMA-ES, which are employed in this study.

Chapter 3 proposes the learning rate adaptation method for accelerating optimization of xNES. In order to achieve efficient optimization, the proposed method increases the learning rate when a sufficient tendency of updates in distribution parameters is measured. The tendency measurement builds on the evolution path in the distribution parameter space, which accumulates parameter movements. Experimental evaluations on both unimodal and multimodal problems demonstrate that the proposed method works properly depending on a search situation and is effective over existing methods, such as those using a fixed learning rate.

Chapter 4 proposes the learning rate adaptation method for solving multimodal or noisy problems. To achieve this, we first explore and highlight the importance of a small learning rate. Building on these discussions, we develop a

novel learning rate adaptation mechanism for CMA-ES that maintains a constant signal-to-noise ratio. We investigate the behavior of CMA-ES with the proposed learning rate adaptation mechanism through numerical experiments and compare the results with those obtained for CMA-ES with a fixed learning rate and with population size adaptation. The results show that CMA-ES with the proposed learning rate adaptation performs effectively for multimodal and/or noisy problems, eliminating the need for extremely expensive learning rate tuning.

Finally, Chapter 5 concludes this study and presents the future works.

Chapter 2

Preliminaries

In Section 2.1, we first introduce black-box continuous optimization and then focus on multimodality and additive noise, which constitute the problem settings addressed in this work. After that, we present xNES [26] (Section 2.2) and CMA-ES [39] (Section 2.3), which are employed for our learning rate adaptation mechanisms. Finally, we provide IGO [73] (Section 2.4), which is a generalized framework that includes xNES and CMA-ES as its instances. In this study, we consider minimizing the objective function $f : \mathbb{R}^d \to \mathbb{R}$.

2.1 Black-Box Continuous Optimization

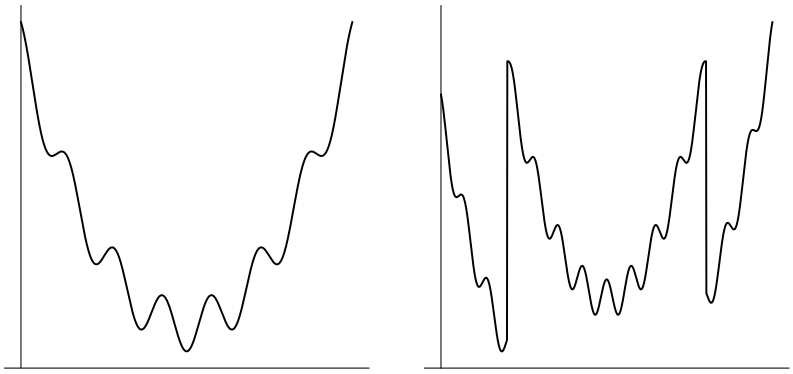
Black-box continuous optimization refers to a class of optimization problems where the objective function f is treated as a "black box". This means we can query the function to obtain an output $f(x)$ for a given input x, but we have no direct access to its internal structure, such as gradients or explicit algebraic representations. These problems often arise in real-world scenarios where the objective function is costly to evaluate, non-differentiable, and reliant on numerical simulations, making them challenging to solve effectively.

In the following, we describe two key aspects that make black-box continuous optimization more challenging: multimodality and additive noise.

2.1.1 Multimodality

Multimodal problems are characterized by the presence of multiple local optima, posing significant challenges for optimization algorithms in identifying

the global optimum. Without careful design, optimization algorithms tend to get trapped in local optima, often resulting in suboptimal solutions.



(a) Well-structured (b) Weakly structured

Figure 2.1: Illustrative examples of well-structured and weakly structured multimodal problems.

In this study, we classify multimodal problems into two categories: well-structured and weakly structured. Well-structured multimodal problems resemble unimodal functions with added (deterministic) noise, while weakly structured multimodal problems lack such clear structure. Figure 2.1 provides illustrative examples of well-structured and weakly structured multimodal problems. In this work, we focus on well-structured multimodal problems. Addressing weakly structured multimodal problems requires additional operations, such as restart strategies, and is therefore considered an interesting direction for future research.

2.1.2 Additive Noise

In Chapter 4, we also consider scenarios where the observed output of the objective function is given by:

$$y = f(x) + \epsilon, \tag{2.1}$$

where $f(x)$ represents the true objective function value and ϵ denotes additive (unbiased) noise. In this context, additive noise implies that the observed value y is the true value $f(x)$ perturbed by the noise term ϵ. This type of noise is prevalent in real-world problems, particularly those involving simulations and physical experiments.

2.2 xNES

Natural Evolution Strategies (NES) [95, 26, 88, 99] is a promising framework for black-box continuous optimization problems. Instead of directly seeking the optimal solution x^*, NES optimizes the parameter θ of a probability distribution $p(x|\theta)$. The expectation of the objective function $f(x)$ over the solution space $J(\theta) = \int f(x)p(x|\theta)dx$ is minimized by repeatedly updating the parameter of the probability distribution based on the estimated natural gradient [9]. This process of taking the expectation is sometimes referred to as stochastic relaxation. NES has been applied across various domains [87, 22, 42, 15, 40, 47, 96, 16, 20, 86, 21, 53, 98, 51].

xNES [26] is simple and promising variant of NES. It employes a multivariate normal distribution, $\mathcal{N}(m, \sigma^2 BB^\top)$, as the probability distribution. Here, $m \in \mathbb{R}^d$ is the mean vector, $\sigma \in \mathbb{R}_{>0}$ is the step-size, and $B \in \mathbb{R}^{d\times d}$ is the normalization transformation matrix where $\det(B) = 1$. The update of xNES is performed by using an estimated natural gradient in the parameter space of the multivariate normal distribution.

xNES first initializes the parameters $m^{(0)}$, $\sigma^{(0)}$, and $B^{(0)}$. Then, the following steps are repeated until a stopping criterion is met.

Step 1. Sampling and Evaluation

At iteration $t+1$ (where t begins at 0), λ candidate solutions x_i $(i = 1, 2, \cdots, \lambda)$ are sampled independently from $\mathcal{N}(m^{(t)}, (\sigma^{(t)})^2 B^{(t)}(B^{(t)})^\top)$, as follows. Generate d-dimensional standard normal vectors $z_i \sim \mathcal{N}(0, I)$ and compute $x_i = m^{(t)} + \sigma^{(t)} B^{(t)} z_i$. The solutions are evaluated on f and sorted in ascending order. Let $x_{i:\lambda}$ be the i-th best candidate solution, that is, $f(x_{1:\lambda}) \leqslant f(x_{2:\lambda}) \leqslant \cdots \leqslant f(x_{\lambda:\lambda})$ for minimization. In addition, we let $z_{i:\lambda}$ be the intermediate vectors corresponding to $x_{i:\lambda}$.

Step 2. Estimate Natural Gradient

Estimate the natural gradient based on the evaluated solutions as follows:

$$G_\delta = \sum_{i=1}^{\lambda} w_i z_{i:\lambda}, G_M = \sum_{i=1}^{\lambda} w_i (z_{i:\lambda} z_{i:\lambda}^\top - I),$$
$$G_\sigma = \mathrm{Tr}(G_M)/d, G_B = G_M - G_\sigma \cdot I,$$

where w_i is the weight function

$$w_i = \frac{\max\left(0, \ln\left(\frac{\lambda}{2}+1\right) - \ln(i)\right)}{\sum_{j=1}^{\lambda} \max\left(0, \ln\left(\frac{\lambda}{2}+1\right) - \ln(i)\right)} - \frac{1}{\lambda}.$$

The weight function holds $\sum_{i=1}^{\lambda} w_i = 0$. Note that xNES uses the weight function instead of using raw evaluation values. This technique is called *fitness shaping*, and it improves the robustness of the algorithm due to the invariance for the monotone transformation of the objective function.

Step 3. Update Distribution Parameters

Based on the estimated natural gradient, update the parameters of the multivariate normal distributions as follows:

$$\begin{aligned} m^{(t)} &= m^{(t)} + \eta_m \sigma B^{(t)} G_\delta, \\ \sigma^{(t)} &= \sigma^{(t)} \cdot \exp(\eta_\sigma / 2 \cdot G_\sigma), \\ B^{(t)} &= B^{(t)} \cdot \exp(\eta_B / 2 \cdot G_B), \end{aligned}$$

where η_m, η_σ, and η_B are the learning rates for updating m, σ, and B, respectively. These learning rates above have default values [26]; $\eta_m = 1$ and $\eta_\sigma = \eta_B = \frac{3}{5} \cdot \frac{(3+\log(d))}{d\sqrt{d}}$.

2.3 CMA-ES

The covariance matrix adaptation evolution strategy (CMA-ES) [39, 33] is among the most successful methods available for solving continuous black-box optimization problems; its effectiveness has been confirmed through various real-world applications [72, 50, 49, 58, 23, 28, 94, 89, 43, 74, 90, 75, 56, 17, 44, 45]. CMA-ES performs optimization by updating the multivariate Gaussian distribution; that is, it first samples candidate solutions from the distribution and then updates the distribution parameters (i.e., the mean vector m and covariance matrix $\Sigma = \sigma^2 C$) based on the objective function f. This update is partly based on the natural gradient descent [6, 73] of the expected f, and m and C in CMA-ES are updated to reduce the expected evaluation value.

CMA-ES first initializes the $m^{(0)}, \sigma^{(0)}$, and $C^{(0)}$ parameters. Thereafter, the following steps are repeated until a predefined stopping criterion is met.

Step 1. Sampling and Evaluation
At iteration $t+1$ (where t begins at 0), λ candidate solutions x_i $(i = 1, 2, \cdots, \lambda)$ are sampled independently from $\mathcal{N}(m^{(t)}, (\sigma^{(t)})^2 C^{(t)})$, as follows:

$$y_i = \sqrt{C^{(t)}} z_i, \tag{2.2}$$

$$x_i = m^{(t)} + \sigma^{(t)} y_i, \tag{2.3}$$

where $z_i \sim \mathcal{N}(0, I)$ and I is the identity matrix. The solutions are evaluated on f and sorted in ascending order. Let $x_{i:\lambda}$ be the i-th best candidate solution, that is, $f(x_{1:\lambda}) \leqslant f(x_{2:\lambda}) \leqslant \cdots \leqslant f(x_{\lambda:\lambda})$ for minimization. In addition, we let $y_{i:\lambda}$ and $z_{i:\lambda}$ be the intermediate vectors in Equations (2.2) and (2.3) corresponding to $x_{i:\lambda}$.

Step 2. Compute Evolution Paths
The weighted averages $dy = \sum_{i=1}^{\mu} w_i y_{i:\lambda}$ and $dz = \sum_{i=1}^{\mu} w_i z_{i:\lambda}$ of the intermediate vectors are calculated using the parent number $\mu \leqslant \lambda$ and weight function w_i, where $\sum_{i=1}^{\mu} w_i = 1$. The evolution paths are updated as follows:

$$p_\sigma^{(t+1)} = (1 - c_\sigma) p_\sigma^{(t)} + \sqrt{c_\sigma (2 - c_\sigma) \mu_w}\, dz, \tag{2.4}$$

$$p_c^{(t+1)} = (1 - c_c) p_c^{(t)} + h_\sigma^{(t+1)} \sqrt{c_c (2 - c_c) \mu_w}\, dy, \tag{2.5}$$

where $\mu_w = 1/\sum_{i=1}^{\mu} w_i^2$, c_σ, and c_c are the cumulation factors, and $h_\sigma^{(t+1)}$ is the Heaviside function, which is defined as follows [35]:

$$h_\sigma^{(t+1)} = \begin{cases} 1 & \text{if } \frac{\|p_\sigma^{(t+1)}\|^2}{1-(1-c_\sigma)^{2(t+1)}} < \left(2 + \frac{4}{d+1}\right) d, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

Step 3. Update Distribution Parameters
The distribution parameters are updated as follows [35]:

$$m^{(t+1)} = m^{(t)} + c_m \sigma^{(t)} dy, \tag{2.7}$$

$$\sigma^{(t+1)} = \sigma^{(t)} \exp\left(\min\left(1, \frac{c_\sigma}{d_\sigma}\left(\frac{\|p_\sigma^{(t+1)}\|}{\mathbb{E}[\|\mathcal{N}(0, I)\|]} - 1\right)\right)\right), \tag{2.8}$$

$$C^{(t+1)} = \left(1 + (1 - h_\sigma^{(t+1)}) c_1 c_c (2 - c_c)\right) C^{(t)} \tag{2.9}$$

$$+ \underbrace{c_1 \left[p_c^{(t+1)} \left(p_c^{(t+1)}\right)^\top - C^{(t)}\right]}_{\text{rank-one update}} + \underbrace{c_\mu \sum_{i=1}^{\mu} w_i \left[y_{i:\lambda} y_{i:\lambda}^\top - C^{(t)}\right]}_{\text{rank-}\mu\text{ update}},$$

where $\mathbb{E}[\|\mathcal{N}(0, I)\|] \approx \sqrt{d}\left(1 - \frac{1}{4d} + \frac{1}{21d^2}\right)$ denotes the expected Euclidean norm of the sample of a standard normal distribution and c_m is the learning rate for m, which is typically set to 1. c_1 and c_μ are the learning rates for the rank-one and -μ updates of C, respectively, and d_σ is the damping factor for the σ adaptation.

2.4 IGO

The IGO [73] framework is a unified framework for stochastic search methods, which has been actively studied in recent years [8, 2, 18, 93, 1, 31, 84].

Given a family of probability distributions parameterized by $\theta \in \Theta$, the original objective function f is transformed into a new objective function J_θ that is defined in the distribution-parameter space Θ.

For the family of Gaussian distributions, the IGO algorithms recover the pure rank-μ-update CMA-ES, eliminating the σ adaptation and rank-one update from the procedures in Section 2.3. To investigate the effects of learning rates on CMA-ES, we focus on their properties within the context of the IGO framework with a family of Gaussian distributions in Section 4.2. This section presents the background of the IGO framework.

Instead of minimizing the original objective f over the input domain $\mathbb{R}^d$, IGO maximizes a new objective J_θ over the distribution-parameter domain Θ. Let $u : [0, 1] \to \mathbb{R}$ be a bounded, non-increasing function, and P_θ be the Lebesgue measure on $\mathbb{R}^d$ corresponding to the probability density $p(x; \theta)$. We define the utility function W_θ^f as

$$W_\theta^f(x) = u(q_\theta(x)), \tag{2.10}$$

where $q_\theta(x)$ is the quantile function that is defined as $q_\theta(x) := P_\theta[y : f(y) \leqslant f(x)]$ for minimization. The objective updating θ, given the *current* distribution parameters $\theta^{(t)}$, is defined as the expectation of the weighted quantile function $W_{\theta^{(t)}}^f(x)$ over $p(x; \theta)$:

$$J_{\theta^{(t)}}(\theta) = \mathbb{E}_{x \sim p(x;\theta)}[W_{\theta^{(t)}}^f(x)]. \tag{2.11}$$

The objective $J_{\theta^{(t)}}(\theta)$ is maximized based on the *natural* gradient [9, 10]. By using the "log-likelihood trick" under some mild conditions, the *vanilla* gradient can be calculated as

$$\nabla_\theta J_{\theta^{(t)}}(\theta) = \mathbb{E}_{x \sim p(x;\theta)}[W_{\theta^{(t)}}^f(x) \nabla_\theta \ln p(x; \theta)]. \tag{2.12}$$

The *natural* gradient is obtained through the product of the inverse of the Fisher information matrix F at $\theta^{(t)}$ and the vanilla gradient as follows:

$$\tilde{\nabla}_\theta J_{\theta^{(t)}}(\theta) = \mathbb{E}_{x\sim p(x;\theta)}[W^f_{\theta^{(t)}}(x)\tilde{\nabla}_\theta \ln p(x;\theta)], \tag{2.13}$$

where $\tilde{\nabla}_\theta \ln p(x;\theta) = F^{-1}\nabla_\theta \ln p(x;\theta)$.

In practice, the integral cannot be calculated in a closed form and is therefore estimated using the Monte Carlo method as follows:

$$\tilde{\nabla}_\theta J_{\theta^{(t)}}(\theta) \approx \frac{1}{\lambda}\sum_{i=1}^{\lambda} W^f_{\theta^{(t)}}(x_i)\tilde{\nabla}_\theta \ln p(x_i;\theta), \tag{2.14}$$

where $\{x_i\}_{i=1}^{\lambda}$ are λ i.i.d. samples obtained from probability distribution $p(x_i;\theta)$. The IGO algorithms implement the IGO framework using the estimated natural gradient, whose updated equation is as follows:

$$\theta^{(t+1)} = \theta^{(t)} + \eta\sum_{i=1}^{\lambda}\frac{W^f_{\theta^{(t)}}(x_i)}{\lambda}\tilde{\nabla}_\theta \ln p(x_i;\theta^{(t)}), \tag{2.15}$$

where η denotes the learning rate. In practice, $W^f_{\theta^{(t)}}(x_i)$ is also estimated based on the *ranking* of $\{x_i\}_{i=1}^{\lambda}$.

As elucidated herein, the IGO framework, with a family of Gaussian distributions, recovers the rank-μ-update CMA-ES [6, 7, 73]. If the distribution parameter $\theta = (m^\top, \mathrm{vec}(C)^\top)^\top$, then [6]:

$$\tilde{\nabla}_\theta \ln p(x;\theta) = \begin{pmatrix} x-m \\ \mathrm{vec}((x-m)(x-m)^\top - C)\end{pmatrix}. \tag{2.16}$$

Thus, Eq. (2.15) can be rewritten as

$$m^{(t+1)} = m^{(t)} + \eta\sum_{i=1}^{\lambda}\frac{W^f_{\theta^{(t)}}(x_i)}{\lambda}(x_i - m^{(t)}), \tag{2.17}$$

$$C^{(t+1)} = C^{(t)} + \eta\sum_{i=1}^{\lambda}\frac{W^f_{\theta^{(t)}}(x_i)}{\lambda}\left((x_i - m^{(t)})(x_i - m^{(t)})^\top - C^{(t)}\right). \tag{2.18}$$

Consequently, by ignoring the σ adaptation and rank-one update in CMA-ES, assuming $c_m = c_\mu(:=\eta)$, and considering that w_i in CMA-ES is an approximation

of $W^f_{\theta^{(t)}}(x_i)/\lambda$ in the IGO update, the m and C updates through the IGO algorithm (Eqs. (2.17) and (2.18), respectively) align with those of CMA-ES (Eqs. (2.7) and (2.9), respectively). It should be noted that xNES can also be recovered by employing the different parametrization [73].

Chapter 3

Learning Rate Adaptation for Acceleration

3.1 Introduction

As with other evolution strategies, one of the critical parameters of xNES is a learning rate for the parameter of the probability distribution. If the learning rate is too large, the parameter update will be unstable and the performance will deteriorate. On the other hand, if the learning rate is too small, the speed of approaching the optimal solution will be slow, resulting in poor performance. Therefore, setting an appropriate learning rate is essential for maximizing the performance of xNES.

There are a few studies on learning rate adaptation of evolution strategies. The most closely related to ours is Self-CMA-ES proposed by Loshchilov et al. [57]. Self-CMA-ES is a method that adapts the learning rate by backtracking into the past and applying a maximum likelihood estimation approach to address the question: "What value of the learning rate would have produced better solutions?" Their experiments on unimodal benchmark problems have demonstrated that Self-CMA-ES enables more efficient search compared to CMA-ES when a sufficiently large population size is used. While promising, Self-CMA-ES requires an additional instance of CMA-ES, referred to as the *auxiliary* CMA-ES, which complicates the method and makes understanding the algorithm's behavior more challenging. Although a similar learning rate adaptation method based on meta-learning exists [77, 79], it adapts only the learning rate of the step-size, not the entire covariance matrix. Another related approach is DX-NES proposed

by Fukushima [24]. DX-NES accumulates the movement of the mean vector as an evolution path and classifies the current search phase into one of three categories based on the norm of the path. Each phase has a predefined learning rate, and DX-NES dynamically switches between them during the search to achieve efficient optimization. This learning rate switching has demonstrated strong empirical performance and has been adopted in subsequent studies on DX-NES [70, 67, 68]. However, selecting a learning rate from only three candidates is an ad hoc operation, and the discrete switching hinders a clear understanding of the algorithm's behavior. Intuitively, it seems more reasonable for the learning rate to transition continuously. In summary, we aim to achieve a simpler and more refined approach to learning rate adaptation for xNES.

In this chapter, we develop the learning rate adaptation method for accelerating optimization of xNES. To judge whether the learning rate should be increased or not, the proposed method measures tendencies in distribution parameter updates. Then, the proposed method accelerates the optimization by increasing the learning rate appropriately when a sufficient tendency is detected. The tendency measurement builds on the method utilized in the population size adaptation of CMA-ES [64]. Our method enables a larger learning rate for relatively easy problems, resulting in faster search. Conversely, for more difficult problems (e.g., multimodal problems), it allows for a small learning rate, leading to a robust and stable search. Experimental evaluations on both unimodal and multimodal problems demonstrate that the proposed method works properly depending on a search situation and is effective over existing methods, such as those using a fixed learning rate.

It is important to acknowledge that this study deals with well-structured multimodal problems rather than weakly structured ones: To address weakly structured multimodal problems, additional operations such as restart strategies [12, 32, 55, 97] are required. We consider this an important direction for future work.

The remainder of this chapter is organized as follows: Section 3.2 illustrates the optimization landscape on multimodal problems in view of stochastic relaxation. Section 3.3 presents the proposed learning rate adaptation method for acceleration. Section 3.4 evaluates the performance of the proposed method on unimodal and multimodal problems. Finally, Section 3.5 concludes this chapter by discussing the limitations of this study.

This chapter is based on the author's previous study [69].

3.2 Landscape on Multimodal Problems

In this section, we explore the optimization landscape of multimodal problems through the lens of xNES on the Rastrigin function. Section 3.2.1 examines the impact of stochastic relaxation, shedding light on the distinctive characteristics of optimization in the Gaussian distribution's parameter space. Next, Section 3.2.2 investigates the influence of population size by presenting a stochastically estimated version of the exact landscape. This analysis highlights the need to automatically calibrate the learning rate based on the population size. Finally, Section 3.2.3 emphasizes the importance of dynamically adapting the learning rate. This is illustrated through the landscapes of the Rastrigin function across different distribution parameters, motivating the need for dynamic learning rate adaptation during the optimization process.

3.2.1 Optimization in x-space vs. θ-space

In Section 3.1, we stated that the learning rate should be kept small for *difficult* problems, such as multimodal ones. This might be confusing to readers unfamiliar with evolution strategies, as in other fields, the small learning rate is often associated with the optimization getting trapped in local optima. For example, in stochastic gradient descent (SGD) for deep neural networks (i.e., non-convex optimization), it is widely known that a large learning rate can help escape local optima [48], which may seem to contradict our discussion.[1] To clarify why the learning rate of evolution strategies should be small for multimodal problems, in this section, we illustrate the optimization landscape. This illustration offers an intuition of the ideal search scenario for xNES to effectively solve such multimodal problems.

The important difference between SGD and xNES lies in the search space: SGD directly seeks the optimal solution x^* in the original search space, referred to as the x-space in this work. In contrast, xNES optimizes within the parameter space of a multivariate Gaussian distribution, referred to as the θ-space. It should be noted that the objective of xNES is the expectation of the objective function $J(\theta) := \mathbb{E}_{x\sim p(x;\theta)}[f(x)]$ rather than the objective function $f(x)$ itself. (Note that the xNES objective corresponds the IGO objective using a Gaussian distribution

[1] In the context of SGD for deep neural networks, a large learning rate is also known to be advantageous for improving generalization, as it encourages convergence to flat minima. However, this aspect is beyond the scope of our work, as our focus is on black-box continuous optimization.

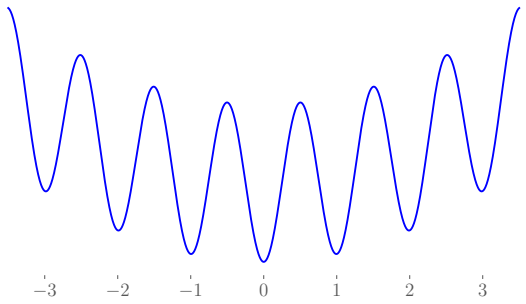


Figure 3.1: Rastrigin function.

and assuming $W_\theta^f = -f$.) Therefore, understanding the effect of this stochastic relaxation leads to a deeper understanding of the ideal behavior for xNES.

As a related note, the effect of the stochastic relaxation with Gaussian distribution is analyzed theoretically in [60]. However, their analysis treats the (co)variance as a fixed parameter, which differs from ours. We believe that adapting the (co)variance is critical for understanding the effect of stochastic relaxation in xNES, as demonstrated below.

To illustrate the optimization landscape from a stochastic relaxation perspective, we employ the 1-dimensional Rastrigin function $f(x) = 10 + x^2 - 10\cos(2\pi x)$, which is one of the representative well-structured multimodal benchmark problems (Fig. 3.1). For calculating the stochastic relaxation, we parameterize the Gaussian distribution by $\theta = (m, v)$, where m is the mean the v is the variance. Assuming that f is known, the expectation of the objective is calculated as follows:

$$J(\theta) = \mathbb{E}_{x\sim p(x;\theta=(m,v))}[10 + x^2 - 10\cos(2\pi x)] \tag{3.1}$$

$$= 10 + m^2 + v - 10\exp(-2\pi^2 v)\cos(2\pi m). \tag{3.2}$$

Note that we used

$$\cos(2\pi x) = \frac{\exp(i2\pi x) + \exp(-i2\pi x)}{2}, \tag{3.3}$$

where i is the complex number, and the moment-generating function of the exponential term is:

$$\mathbb{E}[\exp(ikx)] = \exp\left(ikm - \frac{1}{2}k^2 v\right). \tag{3.4}$$

For plotting the figure, we set the parameters' range to $m \in [-3.0, 3.0]$ and $v \in [0.0, 3.0]$. A 100×100 grid was created for each pair of m and v, and the evaluation values were calculated for each.

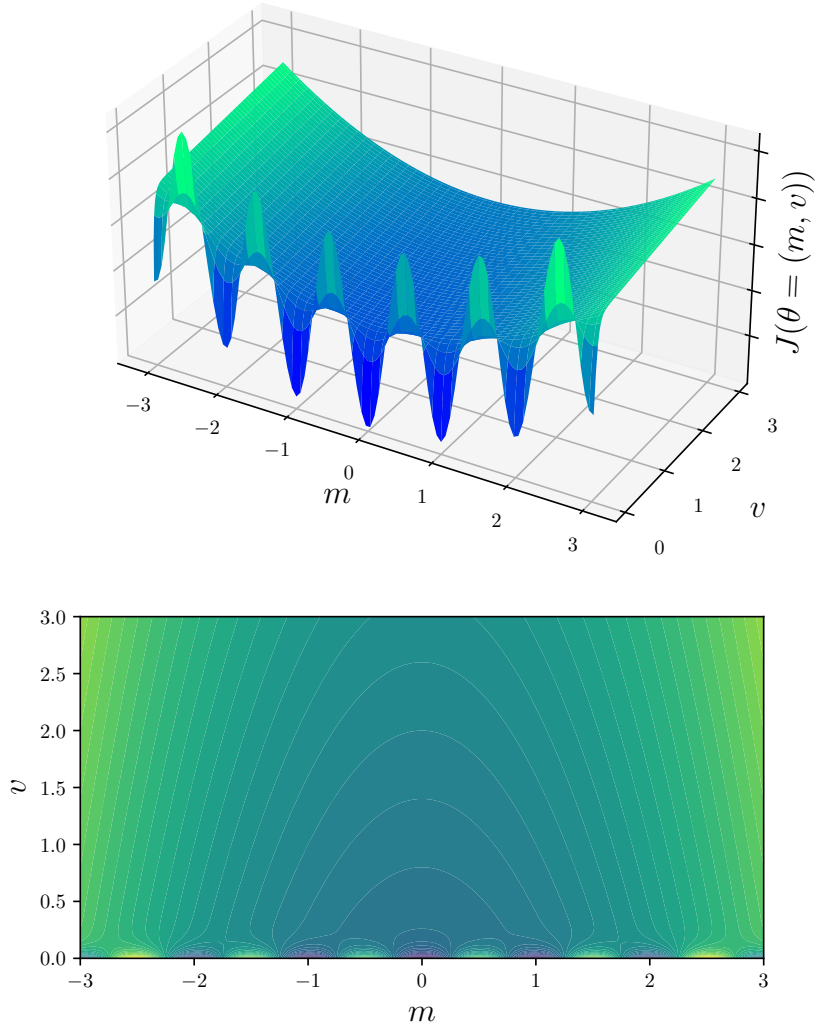


Figure 3.2: Landscape of the Rastrigin function with stochastic relaxation (θ-space). The upper figure shows the 3D plot, while the lower figure presents the 2D plot with contour lines.

Figure 3.2 illustrates the landscape of the Rastrigin function with stochastic relaxation. (For reference, we present the case of the Sphere function in Appendix A.) It should be noted that the dimension of the original landscape is one, the landscape with the stochastic relaxation is two (the mean vector m and the variance v). By comparing the original landscape (Fig. 3.1) and the landscape incorporating stochastic relaxation (Fig. 3.2), several key differences emerge. In the original landscape, multimodality is present across the entire region. To overcome local optima, the optimization method typically requires a large learning rate, as highlighted in the literature of the deep neural network optimization [48]. In contrast, in the landscape with stochastic relaxation, multimodality is confined to specific regions, namely regions with small v values. Assuming the optimization begins with a large v (i.e., an initial distribution with high entropy), the multimodality does not pose significant challenges during the early stages of optimization. As the optimization progresses, multimodality may arise in regions with small v values, at which point it becomes crucial to avoid local optima. (Note that $\lim_{\sigma \to 0} \mathbb{E}_{x \sim \mathcal{N}(m,\sigma^2)}[f(x)] = f(m)$, provided that $f(x)$ is continuous in the vicinity of m.) In such situations, a small learning rate appears to help the optimization process avoid local optima and converge toward the global optimum. A more detailed discussion of the behavior as the learning rate approaches zero is provided in Chapter 4.

Another interesting question might be: *If the landscape in the x-space is unimodal, is the landscape in the θ-space always unimodal?* It is unclear whether this holds for general problems, but it can be proven in a restricted scenario. Consider f as a quadratic convex function, $f(x) = x^\top A x$, where A is a positive definite matrix. In the θ-space with the multivariate Gaussian distribution $\theta = (m, \Sigma)$, we have $J(\theta) = E[f(x)] = m^\top A m + \mathrm{Tr}(A\Sigma)$. First, J is clearly convex with respect to m since A is positive definite. Next, since the trace is a linear operator, J is also convex with respect to Σ (as $\mathrm{Tr}(A\Sigma)$ is linear in Σ). Determining whether this result can be extended to a broader class of functions remains an interesting question.

In summary, the landscape with stochastic relaxation differs from the original landscape, leading to distinct optimization scenarios. Furthermore, based on the preceding discussion, keeping a small learning learning rate appears crucial for successful optimization within the stochastic relaxation regime. This contrasts with the non-convex optimization for deep neural networks, where a larger learning rate is typically necessary.

3.2.2 Effect of Population Size

In the actual optimization of xNES, the exact expectation of the objective function value with respect to the distribution parameters cannot be computed, as the objective function is black-box and thus must be estimated from samples. In order to gain an intuitive understanding of the effect of the population size in the context of stochastic relaxation, we illustrate the estimation version of Figure 3.2, using estimated evaluation values instead of the exact values. The evaluation value is computed using the Monte Carlo method:

$$J(\theta) = \mathbb{E}[f(x)] \approx \frac{1}{\lambda} \sum_{i=1}^{\lambda} f(x_i), \tag{3.5}$$

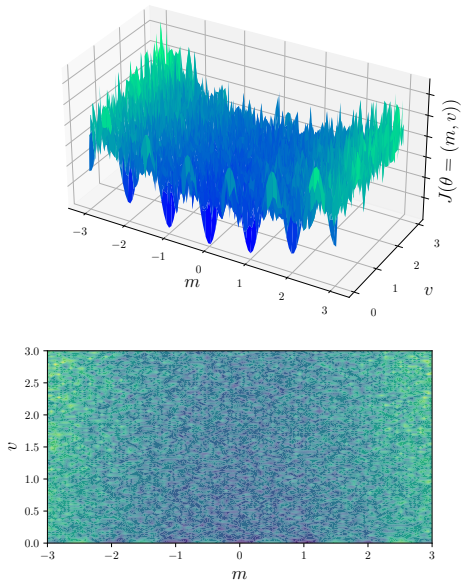
where x_i are samples drawn independently and identically distributed (i.i.d.) from the Gaussian distribution $\mathcal{N}(m, v)$. We vary the population size λ across $\{10, 100, 1000, 10000\}$. It is important to note that, in actual optimization, xNES utilizes ranking information rather than raw estimated evaluation values. Thus, the actual landscape from the perspective of xNES differs to some extent. Nevertheless, we believe the discussion in this section remains valuable, as it provides an intuitive understanding of the effect of population size.

Figure 3.3 illustrates the landscape with $\lambda \in \{10, 100, 1000, 10000\}$ for the Rastrigin function with stochastic relaxation. As expected, we observe that the landscape becomes severely rugged, especially for small values of λ. This suggests that maintaining a small learning rate is necessary for tackling such difficult problems. However, determining appropriate values of λ in advance is impractical, as it is time-consuming. Moreover, practitioners often prefer to set λ equal to the number of available workers for parallelization. This discussion emphasizes the importance of learning rate adaptation in practical situations.

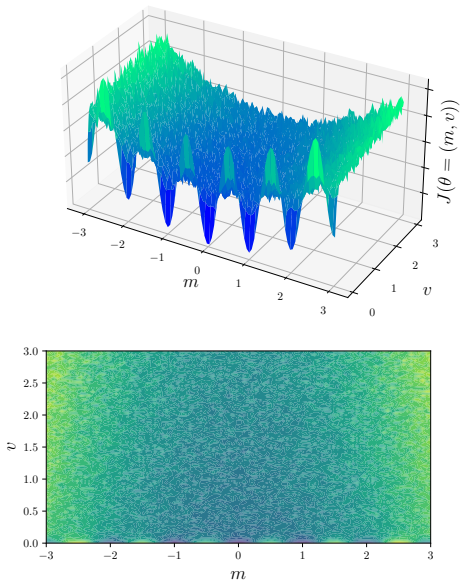
3.2.3 Importance of Adaptive Learning Rate

An interesting aspect of optimization on the Rastrigin function is that the difficulty changes dynamically depending on the location of the distribution parameters [82]. Figure 3.4 illustrates how the landscape varies with different parameter settings. We assume that the initial distribution has sufficient entropy (i.e., a large variance).

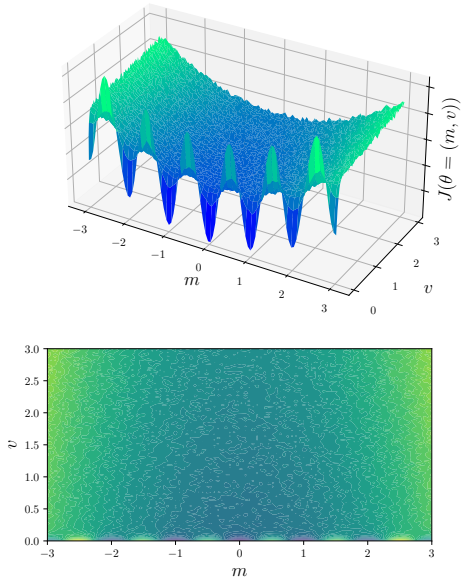
In the early stage (Fig.3.4 (a)), the landscape is observed to be unimodal. As a result, the optimization is expected to be relatively easy. As the optimization



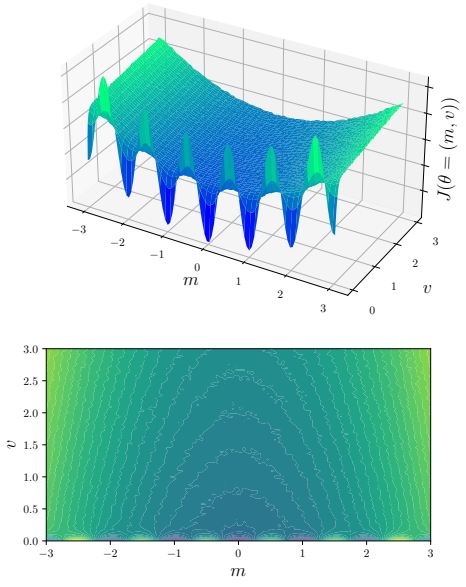
(a) $\lambda = 10$



(b) $\lambda = 100$



(c) $\lambda = 1000$

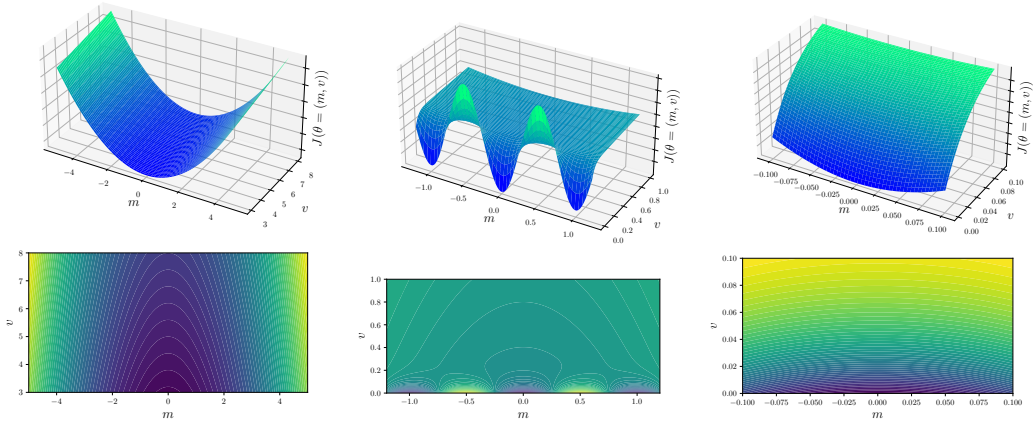


(d) $\lambda = 10000$

Figure 3.3: Landscape with $\lambda \in \{10, 100, 1000, 10000\}$ for the Rastrigin function with stochastic relaxation.

progresses, the search phase transitions to a new stage (Fig.3.4 (b)). In this stage, the landscape becomes multimodal, making the optimization significantly more challenging compared to the early stage. Finally, after converging to a single valley, the search phase transitions to the final stage (Fig. 3.4 (c)). In this stage, the landscape reverts to being unimodal, and the optimization becomes easier again.

This discussion highlights that optimization difficulty can shift dynamically depending on the problem structure. It also implies that a fixed learning rate, even if well-tuned, is insufficient to achieve optimal performance. Adaptivity of the learning rate is therefore essential in this context, motivating the development of an efficient learning rate adaptation method rather than relying solely on hyperparameter tuning.



(a) Early Stage (b) Middle Stage (c) Final Stage

Figure 3.4: Lanscape of the Rastrigin function with different regions. (a) Early Stage corresponds to the region with $m \in [-5, 5]$ and $v \in [3, 8]$. (b) Middle Stage corresponds to the region with $m \in [-1, 1]$ and $v \in [0, 1]$. (c) Final Stage corresponds to the region with $m \in [-0.1, 0.1]$ and $v \in [0, 0.1]$.

3.3 Learning Rate Adaptation

While default values of the learning rates are presented in xNES, Fukushima et al. have pointed out that the default values are too conservative in a certain situation

and there is much room for improvement [24]. However, simply increasing the learning rate causes performance degradation in problems where it is difficult to estimate the natural gradient. It is thus important to adapt the learning rate according to the search situation in order to maximize the performance of xNES.

In this work, we try to adapt the learning rates η_σ and η_B. That is, we focus on only the learning rates related to the covariance matrix. We fix $\eta_m = 1$, which is the default value presented in [26] and widely used in the literature of the CMA-ES [38, 33] as well.

To this end, we introduce a learning rate adaptation mechanism that dynamically adapts the learning rates based on the tendencies of updates in the distribution parameters. To quantify the tendencies of the updates, we introduce an evolution path in the *distribution parameter space* [64], which accumulates successive parameter movements. The length of the evolution path in the parameter space, described in detail in Section 3.3.1, is used to measure the tendencies of the updates. We believe that, if the length of the evolution path is larger than its expectation under a random function, the tendency is strong. On the contrary, we believe that, if the length of the evolution path is close to its expectation under a random function, the estimation is dominated by noise, and the tendency is weak.

In this study, we consider an evolution path in the parameter space of only the covariance matrix, not the mean vector, because the learning rate for the mean vector is fixed. This is different from existing studies that use the evolution path in the parameter space [62, 64].

3.3.1 Evolution Path for Covariance Matrix

In this work, we introduce an evolution path in the parameter space of the covariance matrix to quantify the tendencies of updates in the distribution parameters. We use a modification of the evolution path proposed in [64], which considers both the mean vector and the covariance matrix. Let $\theta = \mathrm{vech}(\Sigma)$ represent the parameter vector of the covariance matrix, where $\Sigma := \sigma^2 BB^\top$ and $\mathrm{vech}(A)$ denotes the vector consisting of the upper triangular elements of the symmetric matrix A. We also denote the parameter movement vector $\delta\theta^{(t+1)} = \mathrm{vech}(\delta\Sigma^{(t+1)})$, where

$$\delta\Sigma^{(t+1)} = (\sigma^{(t+1)})^2 B^{(t+1)} B^{(t+1)\top} - (\sigma^{(t)})^2 B^{(t)} B^{(t)\top}. \tag{3.6}$$

We then define the evolution path in the parameter space of the covariance

matrix.

$$p_{\theta_\Sigma}^{(t+1)} = (1-\beta)p_{\theta_\Sigma}^{(t)} + \sqrt{\beta(2-\beta)}\frac{F_{\Sigma^{(t)}}^{\frac{1}{2}}\delta\theta^{(t+1)}}{\mathbb{E}[\|F_{\Sigma^{(t)}}^{\frac{1}{2}}\delta\theta^{(t+1)}\|^2]^{\frac{1}{2}}}, \tag{3.7}$$

where β is a cumulation factor of the evolution path and $F_{\Sigma^{(t)}}$ is the Fisher information matrix of the covariance matrix of the multivariate normal distribution. The expectation $\mathbb{E}[\cdot]$ is taken under a random function $f(x) = \epsilon$, where ϵ is independently drawn from the identical distribution for each evaluation. We use the approximation of $\mathbb{E}[\|F_{\Sigma^{(t)}}^{\frac{1}{2}}\delta\Sigma^{(t+1)}\|^2]^{\frac{1}{2}}$, which will be derived in Section 3.3.3.

In the implementation, we adopt the matrix representation for clarity and convenience, as shown below:

$$p_{\Sigma}^{(t+1)} = (1-\beta)p_{\Sigma}^{(t)} + \sqrt{\beta(2-\beta)}\frac{(\Sigma^{(t)})^{-\frac{1}{2}}\delta\Sigma^{(t+1)}(\Sigma^{(t)})^{-\frac{1}{2}}}{\mathbb{E}[\|F_{\Sigma^{(t)}}^{\frac{1}{2}}\delta\theta^{(t+1)}\|^2]^{\frac{1}{2}}}, \tag{3.8}$$

Using the result from Eq. (21) and Appendix B in [62], the squared norm of the evolution path is expressed as:

$$\|p_{\theta_\Sigma}^{(t+1)}\|^2 = \frac{\mathrm{Tr}\left(\left(p_{\Sigma}^{(t+1)}\right)^2\right)}{2}. \tag{3.9}$$

It is important to note that this squared norm is associated with the Kullback-Leibler (KL) divergence of the parameter movements [62].

Due to the independence $p_{\theta_\Sigma}^{(t)} \perp \delta\theta^{(t+1)}$, the expectation of the squared norm under the random function is given by:

$$\mathbb{E}[\|p_{\theta_\Sigma}^{(t+1)}\|^2] = (1-\beta)^2\mathbb{E}[\|p_{\theta_\Sigma}^{(t)}\|^2] + \beta(2-\beta). \tag{3.10}$$

Defining $\gamma_{\theta_\Sigma}^{(t)} := \mathbb{E}[\|p_{\theta_\Sigma}^{(t)}\|^2]$, the value of $\gamma_{\theta_\Sigma}^{(t+1)}$ can be computed sequentially:

$$\gamma_{\theta_\Sigma}^{(t+1)} = (1-\beta)^2\gamma_{\theta_\Sigma}^{(t)} + \beta(2-\beta). \tag{3.11}$$

We analyze the tendencies of updates in the distribution parameters by comparing $\|p_{\theta_\Sigma}^{(t+1)}\|^2$ (i.e., $\mathrm{Tr}((p_{\Sigma}^{(t+1)})^2)/2$) with $\gamma_{\theta_\Sigma}^{(t+1)}$. Notably, when initialized with $p_{\theta_\Sigma}^{(0)} = 0$ (i.e., $p_{\Sigma}^{(0)} = O$; as in our study), it follows that $\gamma_{\theta_\Sigma}^{(0)} = 0$.

3.3.2 Updating Learning Rate

In this section, we give a procedure for the learning rate adaptation. When the tendency of the updates in the distribution parameters is strong, the learning rate should be increased, and when the tendency is weak, the learning rate should be decreased.

The learning rate adaptation is performed as follows:

$$\eta_\sigma^{(t+1)} = \eta_\sigma^{(t)} \exp\left(\beta_\sigma \left(\frac{\|p_{\theta_\Sigma}^{(t+1)}\|^2}{\alpha_\sigma} - \gamma_{\theta_\Sigma}^{(t+1)}\right)\right), \tag{3.12}$$

$$\eta_B^{(t+1)} = \eta_B^{(t)} \exp\left(\beta_B \left(\frac{\|p_{\theta_\Sigma}^{(t+1)}\|^2}{\alpha_B} - \gamma_{\theta_\Sigma}^{(t+1)}\right)\right), \tag{3.13}$$

where $\alpha_\sigma, \alpha_B, \beta_\sigma$, and β_B are pre-defined hyperparameters. It is possible to set different hyperparameters for η_σ and η_B, respectively, if needed. In this study, we employ the same value for easier interpretation, i.e., $\alpha := \alpha_\sigma = \alpha_B$ and $\beta := \beta_\sigma = \beta_B$.

We clip the learning rates to prevent them from being updated to unexpected ranges by the following equations:

$$\eta_\sigma^{(t+1)} \leftarrow \mathrm{clip}(\eta_\sigma^{(t+1)}, \eta_\sigma^{\min}, \eta_\sigma^{\max}), \tag{3.14}$$

$$\eta_B^{(t+1)} \leftarrow \mathrm{clip}(\eta_B^{(t+1)}, \eta_B^{\min}, \eta_B^{\max}), \tag{3.15}$$

where $\eta_\sigma^{\max}$ and $\eta_\sigma^{\min}$ are the maximum and the minimum values of the learning rate for step-size σ, respectively. Similarly, $\eta_B^{\max}$ and $\eta_B^{\min}$ are the maximum and the minimum values of the learning rate for the normalized transformation matrix B, respectively. The clip function is defined as $\mathrm{clip}(u, a, b) := \min(\max(u, a), b)$.

To prevent extrapolation in the update of the parameter, we set the maximum value of the learning rates to 1, i.e., $\eta_\sigma^{\max} = \eta_B^{\max} = 1$. This maximum value setting is based on the intuition that η_B should not exceed $\eta_m (= 1)$, given that the degrees of freedom in the covariance matrix are $O(d^2)$.[2] Also, the minimum value of the learning rates is set to the default value of xNES, as it is pointed out that the setting of the learning rates in xNES is often too conservative [24]. Therefore, we use the values recommended in [26] for $\eta_\sigma^{\min}$ and $\eta_B^{\min}$, i.e., $\eta_\sigma^{\min} = \eta_B^{\min} = \frac{3}{5} \cdot \frac{(3+\log(d))}{d\sqrt{d}}$

[2] Although the ideal settings of the learning rates in xNES are discussed in [80], the analysis assumes the covariance matrix to be an identity matrix, making it inapplicable to our discussion.

3.3.3 Derivation of Approximation Value

In this section, we derive an approximation of $\mathbb{E}[\|F_{\Sigma^{(t)}}^{\frac{1}{2}}\delta\theta^{(t+1)}\|^2]^{\frac{1}{2}}$, which represents a change of the KL divergence in terms of the covariance matrix.

Let $C^{(t)} = B^{(t)}B^{(t)\top}$, $\delta\Sigma = {\sigma^{(t+1)}}^2 C^{(t+1)} - {\sigma^{(t)}}^2 C^{(t)}$, $\Sigma^{-1} = {\sigma^{(t)}}^{-2} {C^{(t)}}^{-1}$, $\delta\sigma = \sigma^{(t+1)}/\sigma^{(t)}$, and $\delta C = C^{(t+1)} - C^{(t)}$. To derive the approximation, we use the Slepian-Bangs formula [85, 14] and obtain

$$\|F_{\Sigma^{(t)}}^{\frac{1}{2}}\delta\theta\|^2 = \frac{1}{2}\mathrm{Tr}\left(\delta\Sigma\Sigma^{-1}\delta\Sigma\Sigma^{-1}\right).$$

We will derive the expectation of this equation. From the result provided by Nishida and Akimoto [64], we can obtain[3]

$$\begin{aligned}\mathbb{E}[\mathrm{Tr}\left(\delta\Sigma\Sigma^{-1}\delta\Sigma\Sigma^{-1}\right)] = \mathbb{E}[\delta\sigma^4]\mathrm{Tr}\left(\mathbb{E}\left[\left({C^{(t)}}^{-1/2}\cdot\delta C\cdot {C^{(t)}}^{-1/2}\right)^2\right]\right)\\ + d(\mathbb{E}[\delta\sigma^4] - 2\mathbb{E}[\delta\sigma^2]) + d.\end{aligned}$$

We then need to derive the approximation of $\mathbb{E}[\delta\sigma^4], \mathbb{E}[\delta\sigma^2]$, and

$$\mathrm{Tr}\left(\mathbb{E}\left[\left({C^{(t)}}^{-1/2}\cdot\delta C\cdot {C^{(t)}}^{-1/2}\right)^2\right]\right).$$

Derivation of $\mathbb{E}[\delta\sigma^a]$ $(a = 2, 4)$:

The update equation of step-size in xNES can be rewritten as

$$\sigma^{(t+1)} = \sigma^{(t)}\cdot\exp\left(\frac{\eta_\sigma}{2d}\left(\sum_{j=1}^{d}\sum_{i=1}^{\lambda} w_i\left([z_i]_j^2 - 1\right)\right)\right).$$

Then, by the second order Taylor expansion, for any $a \in \mathbb{R}$,

$$\begin{aligned}\mathbb{E}[\delta\sigma^a] &= \mathbb{E}\left[\exp\left(a\frac{\eta_\sigma}{2d}\left(\sum_{j=1}^{d}\sum_{i=1}^{\lambda} w_i\left([z_i]_j^2 - 1\right)\right)\right)\right]\\ &\approx 1 + a\cdot\frac{\eta_\sigma\lambda}{2}\mathbb{E}[w_i([z_i]_j^2 - 1)] + \frac{a^2}{2}\cdot\left(\frac{\eta_\sigma}{2d}\right)^2\mathbb{E}\left[\left(\sum_{j=1}^{d}\sum_{i=1}^{\lambda} w_i\left([z_i]_j^2 - 1\right)\right)^2\right].\end{aligned}$$

[3]Here, we have corrected the minor typos that appeared in [69].

We will thus calculate the expectations in the above equation. First, from $\mathbb{E}[[z_i]^2] = \mathbb{V}[[z_i]] + \mathbb{E}[[z_i]] = 1$, $\mathbb{E}[w_i([z_i]_j^2 - 1)] = 0$. Next, noting that $\mathbb{V}\left[z^2\right] = 2$ and the independence,

$$\mathbb{E}\left[\left(\sum_{j=1}^{d}\sum_{i=1}^{\lambda} w_i\left([z_i]_j^2 - 1\right)\right)^2\right] = \underbrace{\left(\mathbb{E}\left[\sum_{j=1}^{d}\sum_{i=1}^{\lambda} w_i\left([z_i]_j^2 - 1\right)\right]\right)^2}_{=0} + \mathbb{V}\left[\sum_{j=1}^{d}\sum_{i=1}^{\lambda} w_i\left([z_i]_j^2 - 1\right)\right]$$
$$= \sum_{j=1}^{d}\sum_{i=1}^{\lambda} w_i^2 \mathbb{V}[[z_i]_j^2] = 2d/\mu_w,$$

where $\mu_w = \sum_{i=1}^{\lambda} 1/w_i^2$. By combining these results, $\mathbb{E}[\delta\sigma^a] \approx 1 + \frac{a^2\eta_\sigma^2}{4d\mu_w}$. We thus obtain

$$\mathbb{E}[\delta\sigma^4] \approx 1 + \frac{4\eta_\sigma^2}{d\mu_w}, \mathbb{E}[\delta\sigma^2] \approx 1 + \frac{\eta_\sigma^2}{d\mu_w},$$
$$\mathbb{E}[\delta\sigma^4] - 2\mathbb{E}[\delta\sigma^2] \approx \frac{2\eta_\sigma^2}{d\mu_w} - 1.$$

Derivation of $\mathrm{Tr}\left(\mathbb{E}\left[\left(C^{(t)^{-1/2}} \cdot \delta C \cdot C^{(t)^{-1/2}}\right)^2\right]\right)$**:**

Let $\Delta = \mathrm{Tr}\left(\sum_{i=1}^{\lambda} w_i(z_i z_i^\top - I)\right)$. The first order Taylor expansion of C in xNES can be obtained as

$$C^{(t+1)} \approx C^{(t)} + \eta_B C^{(t)^{1/2}}\left(\Delta - \frac{\mathrm{Tr}(\Delta)}{d} I\right) C^{(t)^{1/2}}.$$

From $\delta C \approx \eta_B C^{(t)^{1/2}}\left(\Delta - \frac{\mathrm{Tr}(\Delta)}{d} I\right) C^{(t)^{1/2}}$, $C^{(t)^{-1/2}} \delta C C^{(t)^{-1/2}} \approx \eta_B\left(\Delta - \frac{\mathrm{Tr}(\Delta)}{d} I\right)$,

$$\mathrm{Tr}\left(\mathbb{E}\left[\left(C^{(t)^{-1/2}} \cdot \delta C \cdot C^{(t)^{-1/2}}\right)^2\right]\right)$$
$$\approx \mathrm{Tr}\left(\mathbb{E}\left[\left(\eta_B\left(\Delta - \frac{\mathrm{Tr}(\Delta)}{d} I\right)\right)^2\right]\right)$$
$$= \mathbb{E}\left[\mathrm{Tr}\left(\eta_B\left(\Delta - \frac{\mathrm{Tr}(\Delta)}{d} I\right)\right)^2\right] = \eta_B^2\left\{\mathbb{E}\left[\mathrm{Tr}(\Delta^2)\right] - \frac{1}{d}\mathbb{E}[\mathrm{Tr}(\Delta)^2]\right\}.$$

Derivation of $\mathbb{E}\left[\mathrm{Tr}(\Delta^2)\right]$ **and** $\mathbb{E}[\mathrm{Tr}(\Delta)^2]$**:**

$$\begin{aligned}
\mathbb{E}\left[\mathrm{Tr}(\Delta^2)\right] &= \mathbb{E}\left[\mathrm{Tr}\left(\sum_{i=1}^{\lambda}\sum_{j=1}^{\lambda} w_i w_j (z_i z_i^\top - I)(z_j z_j^\top - I)\right)\right] \\
&= \sum_{i=1}^{\lambda}\sum_{j=1}^{\lambda} w_i w_j \mathbb{E}\left[\mathrm{Tr}\left((z_i z_i^\top - I)(z_j z_j^\top - I)\right)\right] \\
&= \sum_{i=1}^{\lambda}\sum_{j=1}^{\lambda} w_i w_j \mathbb{E}\left[\mathrm{Tr}\left(z_i z_i^\top z_j z_j^\top\right) - \mathrm{Tr}(z_i z_i^\top) - \mathrm{Tr}(z_j z_j^\top) + \mathrm{Tr}(I)\right] \\
&= \sum_{i=1}^{\lambda}\sum_{j=1}^{\lambda} w_i w_j (\mathbb{E}[(z_i^\top z_j)^2] - \mathbb{E}[z_i^\top z_i] - \mathbb{E}[z_j^\top z_j] + d).
\end{aligned}$$

Note that $\mathbb{E}[z_i^\top z_i] = d$. Then, $\forall i, j \geqslant 1 (i \neq j)$, $\mathbb{E}[(z_i^\top z_j)^2] = d, \mathbb{E}[(z_i^\top z_i)^2] = d^2 + 2d$. Therefore,

$$\begin{aligned}
\mathbb{E}\left[\mathrm{Tr}(\Delta^2)\right] &= \sum_{i=1}^{\lambda} w_i^2 (d^2 + 2d - d - d + d) + \sum_{i,j:i\neq j} w_i w_j \underbrace{(d - d - d + d)}_{=0} \\
&= \sum_{i=1}^{\lambda} w_i^2 (d^2 + d) = (d^2 + d)/\mu_w.
\end{aligned}$$

We then derive $\mathbb{E}\left[\mathrm{Tr}(\Delta)^2\right]$. First, $\Delta = \mathrm{Tr}\left(\sum_{i=1}^{\lambda} w_i (z_i z_i^\top - I)\right)$ is written as

$$\begin{aligned}
\mathrm{Tr}\left(\sum_{i=1}^{\lambda} w_i (z_i z_i^\top - I)\right) &= \sum_{i=1}^{\lambda} w_i (\mathrm{Tr}(z_i z_i^\top) - d) \\
&= \sum_{i=1}^{\lambda} w_i \mathrm{Tr}(z_i z_i^\top) \\
&= \sum_{i=1}^{\lambda} w_i \|z_i\|^2.
\end{aligned}$$

In the second line, we used $\sum_{i=1}^{\lambda} w_i = 0$. Then, $\mathbb{E}[\mathrm{Tr}(\Delta)^2] = \mathbb{E}[\sum_{i=1}^{\lambda} w_i^2 \|z_i\|^4 + \sum_{i,j:i\neq j} w_i w_j \mathbb{E}[\|z_i\|^2]\mathbb{E}[\|z_j\|^2] = (d^2+2d)/\mu_w - d^2/\mu_w = 2d/\mu_w$. We used $\sum_{i,j:i\neq j} w_i w_j = \sum_{i,j} w_i w_j - \sum_{i=1}^{\lambda} w_i^2 = -1/\mu_w$ due to $\sum_{i=1}^{\lambda} w_i = 0$.

By combining these results,

$$\mathrm{Tr}\left(\mathbb{E}\left[\left(C^{(t)^{-1/2}} \cdot \delta C \cdot C^{(t)^{-1/2}}\right)^2\right]\right) = \frac{\eta_B^2}{\mu_w}(d^2 + d - 2).$$

Approximation Result:

From the results above,

$$\begin{aligned}
&\mathbb{E}[\mathrm{Tr}\left(\delta\Sigma\Sigma^{-1}\delta\Sigma\Sigma^{-1}\right)] \\
&= \underbrace{\mathbb{E}[\delta\sigma^4]}_{\approx 1+\frac{4\eta_\sigma^2}{d\mu_w}} \underbrace{\mathrm{Tr}\left(\mathbb{E}\left[\left(C^{(t)^{-1/2}} \cdot \delta C \cdot C^{(t)^{-1/2}}\right)^2\right]\right)}_{\approx \eta_B^2(d^2+d-2)/\mu_w} + d(\underbrace{\mathbb{E}[\delta\sigma^4] - 2\mathbb{E}[\delta\sigma^2]}_{\approx \frac{2\eta_\sigma^2}{d\mu_w}-1}) + d \\
&\approx \frac{1}{\mu_w}\left\{\eta_B^2\left(1+\frac{4\eta_\sigma^2}{d\mu_w}\right)(d^2+d-2) + 2\eta_\sigma^2\right\}.
\end{aligned}$$

Therefore,

$$\mathbb{E}[\|F_{\Sigma^{(t)}}^{\frac{1}{2}}\delta\theta^{(t+1)}\|^2] \approx \frac{1}{\mu_w}\left\{\frac{\eta_B^2}{2}\left(1+\frac{4\eta_\sigma^2}{d\mu_w}\right)(d^2+d-2) + \eta_\sigma^2\right\}. \tag{3.16}$$

We recalculate this approximation every iteration because it depends on the dynamically changing learning rates, η_σ and η_B.

3.3.4 Hyperparameter Effects

In this section, we discuss the effect of the hyperparameters α and β. We present an empirical analysis of the sensitivity of these hyperparameters in Appendix A.

For α, an intuitive choice is a value in $1.2 < \alpha < 2$. The reasoning is as follows: When $\alpha = 1$, taking the expectation over a random function, the following unbiased condition holds:

$$\mathbb{E}[\ln \eta_\Sigma^{(t+1)} \mid \eta_\Sigma^{(t)}] = \ln \eta_\Sigma^{(t)}, \tag{3.17}$$

where $\ln \eta_\Sigma$ represents either $\ln \eta_\sigma$ or $\ln \eta_B$.[4] This unbiased condition implies that if there is even a slight tendency, the learning rate is increased This is too aggressive and thus undesirable, especially for challenging problems (e.g. multimodal ones), where maintaining a small learning rate is often crucial. Therefore, a value greater than 1 (e.g., 1.2 or larger), yet not too large (e.g., 2.0 or smaller), is considered desirable. A larger α is preferable when aiming for greater stability. Nevertheless, determining the optimal α remains formidable, as it heavily depends on the structure of the problem.

[4]Note that $\mathbb{E}[\eta_\Sigma^{(t+1)} \mid \eta_\Sigma^{(t)}] > \eta_\Sigma^{(t)}$ even over a random function.

Note that in our approach, the tendency of updates in the distribution parameters becomes more pronounced as the population size λ increases, as explained below. We assume the weights are optimal [3], i.e., $w_i = -\mathbb{E}[\mathcal{N}_{i:\lambda}] / \sum_{i=1}^{\lambda} |\mathbb{E}[\mathcal{N}_{i:\lambda}]|$, where $\mathcal{N}_{i:\lambda}$ is the i-th smallest random variable among λ independently and standard normally distributed random variables, i.e., $\mathcal{N}_{1:\lambda} \leqslant \cdots \leqslant \mathcal{N}_{\lambda:\lambda}$. Under the optimal weights, $\mu_w \approx (2/\pi)\lambda \in \mathcal{O}(\lambda)$. This means that $\mathbb{E}[\|F_{\Sigma^{(t)}}^{\frac{1}{2}} \delta\theta^{(t+1)}\|^2] \approx \mathcal{O}(1/\lambda)$. Therefore, the learning rate tends to increase more readily as λ increases.

For β, it emphasize the tendency of updates across iterations. We consider the case where the updates are drawn i.i.d. from the distribution $\mathcal{D}(v, \mathrm{Cov}[v])$, where $\mathcal{D}(A, B)$ denotes a distribution with expectation A and (co)variance B. In this scenario, the expectation of the squared norm of the evolution path is approximated as follows [62]:

$$\mathbb{E}[\|p_{\theta_\Sigma}^{(t)}\|^2] \approx \frac{2-\beta}{\beta}\|v\|^2 + \mathrm{Tr}(\mathrm{Cov}(v)). \tag{3.18}$$

(We ignored $(1-\beta)^t$ and $(1-\beta)^{2t}$ by considering a sufficiently large t.) Thus, if there is a tendency (i.e., $\|v\| > 0$), the squared norm tends to become larger by taking a small $\beta < 1$ value.

3.3.5 Overall Procedure

The overall procedure of xNES with the proposed learning rate adaptation mechanism is shown in Algorithm 1. The parameters $\eta_\sigma^{\mathrm{def}}$ and η_B^{def} are the recommended setting of the learning rates in [26], and O is the zero matrix. The procedures in line 3-14 are the same as xNES. In line 15, the covariance movement matrix is updated. In line 16, the expectation of the length of the evolution path under a random function is approximated by using Eq. (3.16). In line 17, the evolution path in the parameter space of the covariance matrix is updated. In line 18, the length of the evolution path is calculated. In line 19, the normalization factor for the evolution path is updated. In line 20-21, the learning rates for the step-size and the normalization transformation matrix are updated with clipping.

3.4 Experiments and Discussions

In this section, we investigate the following research questions (RQs).

RQ1. When the learning rate is fixed, how does the evolution path in Eq. (3.8) of xNES behave on unimodal and multimodal functions?

Algorithm 1 xNES with the learning rate adaptation.

Input: $m^{(0)} \in \mathbb{R}^d, \sigma^{(0)} \in \mathbb{R}_{>0}, B^{(0)} \in \mathbb{R}^{d\times d}, \lambda \in \mathbb{N}$

Input: $\alpha_\sigma, \alpha_B, \beta_\sigma, \beta_B, \eta_\sigma^{\min}, \eta_B^{\min}, \eta_\sigma^{\max}, \eta_B^{\max}$

$t = 0, p_\Sigma^{(0)} = O, \gamma_\theta^{(0)} = 0, \eta_\sigma^{(0)} = \eta_\sigma^{\text{def}}, \eta_B^{(0)} = \eta_B^{\text{def}}, \eta_m = 1$

while stopping criterion not met **do**

  for $i \in \{1, \cdots, \lambda\}$ **do**

    $z_i \sim \mathcal{N}(0, I)$

    $x_i = m^{(t)} + \sigma^{(t)} B^{(t)} z_i$

  end for

  Evaluate the solutions and sort $\{(z_i, x_i)\}$

  $G_\delta = \sum_{i=1}^{\lambda} w_i z_i$

  $G_M = \sum_{i=1}^{\lambda} w_i (z_i z_i^\top - I)$

  $G_\sigma = \mathrm{Tr}(G_M)/d$

  $G_B = G_M - G_\sigma \cdot I$

  $m^{(t+1)} = m^{(t)} + \eta_m \sigma^{(t)} B^{(t)} G_\delta$

  $\sigma^{(t+1)} = \sigma^{(t)} \cdot \exp(\eta_\sigma^{(t)}/2 \cdot G_\sigma)$

  $B^{(t+1)} = B^{(t)} \cdot \exp(\eta_B^{(t)}/2 \cdot G_B)$

  $\delta\Sigma^{(t+1)} = (\sigma^{(t+1)})^2 B^{(t+1)} B^{(t+1)\top} - (\sigma^{(t)})^2 B^{(t)} B^{(t)\top}$

  Approximate $\mathbb{E}[\|F_{\Sigma^{(t)}}^{\frac{1}{2}} \delta\theta^{(t+1)}\|^2]^{\frac{1}{2}}$ by Eq. (3.16)

  $p_\Sigma^{(t+1)} = (1-\beta) p_\Sigma^{(t)} + \sqrt{\beta(2-\beta)} \frac{(\Sigma^{(t)})^{-\frac{1}{2}} \delta\Sigma^{(t+1)} (\Sigma^{(t)})^{-\frac{1}{2}}}{\mathbb{E}[\|F_{\Sigma^{(t)}}^{\frac{1}{2}} \delta\theta^{(t+1)}\|^2]^{\frac{1}{2}}}$

  $l_\theta^{(t+1)} = \mathrm{Tr}(p_\Sigma^{(t+1)^2})/2$

  $\gamma_{\theta_\Sigma}^{(t+1)} = (1-\beta)^2 \gamma_{\theta_\Sigma}^{(t)} + \beta(2-\beta)$

  $\eta_\sigma^{(t+1)} = \mathrm{clip}\left(\eta_\sigma^{(t)} \exp\left(\beta_\sigma \left(\frac{l_\theta^{(t+1)}}{\alpha_\sigma} - \gamma_{\theta_\Sigma}^{(t+1)}\right)\right), \eta_\sigma^{\min}, \eta_\sigma^{\max}\right)$

  $\eta_B^{(t+1)} = \mathrm{clip}\left(\eta_B^{(t)} \exp\left(\beta_B \left(\frac{l_\theta^{(t+1)}}{\alpha_B} - \gamma_{\theta_\Sigma}^{(t+1)}\right)\right), \eta_B^{\min}, \eta_B^{\max}\right)$

  $t \leftarrow t + 1$

end while

RQ2. How is the learning rate adapted in xNES with the proposed learning rate adaptation mechanism?

RQ3. Does xNES with the proposed learning rate adaptation mechanism achieve better performance than xNES with fixed learning rates?

We first describe the experimental setups in Section 3.4.1. In Section 3.4.2, we investigate the behavior of the evolution path in xNES with a fixed learning rate (RQ1). We then investigate the behavior of the evolution path and the learning rate in xNES with the *adaptive* learning rate mechanism (RQ2) in Section 3.4.3. Finally, we compare the performance of xNES with the proposed adaptive learning rate mechanism and that with fixed learning rates (RQ3) in Section 3.4.4. The code for running the proposed method is available at GitHub[5].

3.4.1 Experimental Setups

Table 3.1 shows the definition of benchmark problems used in the experiment. We employ two unimodal functions (Sphere and Ellipsoid) and two multimodal functions (Rastrigin and Bohachevsky). While the Rastrigin function has strong multimodality, the Bohachevsky function has relatively weak multimodality. In this experiment, we set the dimension to $d = 10$. The initial parameters are set to $m^0 = [3, \cdots, 3], \sigma^{(0)} = 2.0, B^{(0)} = I$ in the Sphere, Ellipsoid, and Rastrigin functions, and $m^0 = [8, \cdots, 8], \sigma^{(0)} = 7.0, B^{(0)} = I$ in the Bohachevsky function.

The hyperparameters for the proposed learning rate adaptation mechanism are: $\eta_\sigma^{\max} = \eta_B^{\max} = 1$, $\eta_\sigma^{\min} = \eta_B^{\min} = \frac{3}{5} \cdot \frac{(3+\log(d))}{d\sqrt{d}}$, as described in Section 3.3.2. We set $\alpha = 1.3$ and $\beta = 0.2$ based on the preliminary experiments (See Appendix A for the sensitivity analysis of these hyperparameters.). η_σ^{def} and η_B^{def} are set to their default values, i.e., $\eta_\sigma^{\text{def}} = \eta_B^{\text{def}} = \frac{3}{5} \cdot \frac{(3+\log(d))}{d\sqrt{d}}$.

3.4.2 Evolution Path with Fixed Learning Rate

Figure 3.5 shows a typical behavior of the best evaluation value $f(x_{\text{best}})$ and the length of the evolution path l_θ of xNES with a fixed learning rate on the benchmark problems. We use the default learning rate and set the population size $\lambda = 400$ to obtain a reliable estimation of the evolution path.

[5] https://github.com/nomuramasahir0/xnes-adaptive-lr

Table 3.1: Definitions of benchmark problems used in the experiments described in Chapter 3.

Definition
$f_{\text{Sphere}}(x) = \sum_{i=1}^{d} x_i^2$
$f_{\text{Ellipsoid}}(x) = \sum_{i=1}^{d} (1000^{\frac{i-1}{d-1}} x_i)^2$
$f_{\text{Rastrigin}}(x) = 10d + \sum_{i=1}^{d} (x_i^2 - 10\cos(2\pi x_i))$
$f_{\text{Bohachevsky}}(x) = \sum_{i=1}^{d-1} (x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7)$

In the result of the Sphere and Ellipsoid functions where $f(x_{\text{best}})$ is improved quickly, the length of the evolution path $l(\theta)$ becomes long (> 1). We believe this is because detecting the tendency of updates is easy on such unimodal problems.

On the other hand, in the multimodal functions where $f(x_{\text{best}})$ may not be improved easily, different behavior from that of the unimodal functions appears. In the Bohachevsky function, which is a relatively weakly multimodal function, we can observe that the length of the evolution path slightly decreases once. In the Rastrigin function, which has strong multimodality, such a decreasing behavior is prominent in the beginning of the optimization. In fact, the length of the evolution path takes a value close to 1, which is the expected amount of change in KL divergence in a random function, in the period where the number of evaluations is between about 0.5×10^5 and about 1.2×10^5.

3.4.3 Behavior of Learning Rate Adaptation

A typical behavior of xNES with the proposed learning rate adaptation mechanism is depicted in Figure 3.6. In addition to the learning rates η_σ and η_B, the corresponding objective function value $f(x_{\text{best}})$ and the length of the evolution path l_θ are also shown. We employ $\lambda = 30$ for the Sphere and Ellipsoid functions, $\lambda = 300$ for the Rastrigin function, $\lambda = 50$ for the Bohachevsky function, respectively. It is observed in each function that the learning rates also increase when the length of the evolution path increases.

To investigate the effect of the setting of the population size, we conduct an experiment with $\lambda = 10, 20, 30, 40$, and 50 on the 10-dimensional Sphere function. Figure 3.7 shows the result of the experiment. In $\lambda = 10$, the length of the evolution path l_θ does not increase and the learning rates, η_σ and η_B, are then not changed at all. We believe this is due to the tendency of updates in the distribution parameters to be weak when using a small population size. We observe that,

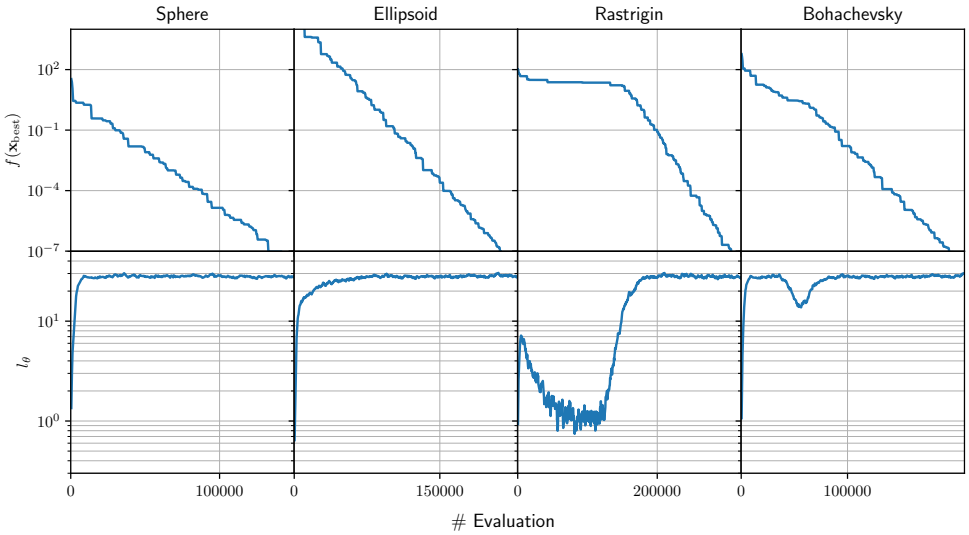


Figure 3.5: Typical behavior of xNES with a fixed learning rate on the 10-dimensional benchmark problems. The horizontal axis represents the number of evaluations. The vertical axes represent the best evaluation value $f(x_{\text{best}})$ and the length of the evolution path l_θ, respectively.

as λ is increased, the length of the evolution path increases, and, as a result, the learning rate also increases. The result suggests that the proposed mechanism can adapt the learning rate appropriately, measuring the tendency of updates in the distribution parameters. This dynamic learning rate adaptation depending on the population size is an advantage over DX-NES [24], which statically injects the population size into the setting of the learning rate.

3.4.4 Fixed Learning Rate vs. Adaptive Learning Rate

To check the effectiveness of the proposed mechanism, we compare the performance of xNES with the proposed learning rate adaptation mechanism and that of xNES with fixed learning rates (= the default value $\times 1, 2, 4, 6, 8$, and 10). The performance metrics is the SP1 value, which is the average number of evaluations until $f(x_{\text{best}})$ reaches a target function value over successful trials divided by the success rate [13, 12]. The SP1 value estimates the expected number of evaluations needed to meet the success criterion, assuming that the expected number of

evaluations for successful and unsuccessful runs are identical.[6] The target function value is set to 10^{-8}. A trial is successful if the target function value is found. We set the maximum number of evaluations to 5×10^5. For the Sphere function and the Ellipsoid function, we employ the population size $\lambda = 10, 20, 30, 40$, and 50. Note that the recommended value of the population size presented in [26] is included, i.e., $4 + \lfloor 3 \ln(10) \rfloor = 10$. For the Rastrigin function, we employ the population size $\lambda = 200, 250, 300, 350$, and 400. For the Bohachevsky function, we employ the population size $\lambda = 30, 40, 50, 60$, and 70. We perform 50 trials to calculate the performance metrics for the Sphere and Ellipsoid functions. We perform 200 trials to calculate it for the Rastrigin and Bohachevsky functions.

Figure 3.8 shows the result of the experiment. We first compare the proposed mechanism (red) and xNES with the default learning rate (blue). In the Sphere and Ellipsoid functions, when $\lambda = 10$, the performance is almost the same, which is consistent in Section 3.4.3. As λ increases, the proposed mechanism shows better performance than xNES with the default learning rate. This is because the tendency of updates in the distribution parameters becomes strong when λ is large, increasing the learning rate and accelerating the search. In the Rastrigin and Bohachevsky functions, the proposed mechanism outperforms xNES with the default learning rate due to the adaptive learning rate.

Next, we compare the proposed mechanism (red) and xNES with other fixed learning rates. In all the benchmark problems, when λ is large, the performance of the proposed mechanism is close to that of xNES with the fixed learning rate of the default value times 8 (pink). However, xNES with the fixed learning rate of the default value times 8 fails to find the optimum in the Sphere and Ellipsoid functions with small population sizes ($\lambda = 10, 20$, and 30) because the learning rate is too high. On the other hand, the proposed mechanism avoids significantly increasing the learning rate when the population size is small, enabling a stable search.

From the result in the multimodal functions, we can observe that the proposed mechanism is competitive with xNES with high learning rates when the population size is large. In particular, in the Rastrigin function, the proposed mechanism and xNES with the fixed learning rate of the default value times 8 and 10 achieve almost the same performance in terms of the average number

[6] When applying CMA-ES to well-structured multimodal problems in practical situations, the expected number of evaluations for unsuccessful runs may be smaller than for successful runs. This is because unsuccessful runs are often terminated early based on a predefined stopping criterion. In such cases, the SP1 value provides an upper bound on the expected number of evaluations needed to meet the success criterion.

of evaluations of successful trials divided by the success rate. This means that the number of evaluations required to find the optimum is about the same if an appropriate restart is performed when the optimum is failed to find. Figure 3.9 shows the success rate of the proposed mechanism (red), xNES with the fixed learning rate of the default value times 8 (pink), and xNES with the fixed learning rate of the default value times 10 (cyan). While these methods are competitive when the population size is large, xNES with the fixed learning rates are more likely to fail when the population size is small. This result suggests that the proposed mechanism is more robust than xNES with a fixed learning rate. A higher success rate in the proposed mechanism is also practically beneficial, as it is often difficult to implement an appropriate restart strategy.

3.5 Conclusion

In this chapter, we proposed a novel learning rate adaptation mechanism for xNES. The mechanism dynamically adapts the learning rate based on tendencies observed in the updates of the distribution parameters. The method for detecting these tendencies draws inspiration from the population size adaptation mechanism of CMA-ES [62, 64]. Specifically, we introduced an evolution path in the parameter space of the covariance matrix. By evaluating the length of this evolution path, we adapt the learning rates associated with the covariance matrix accordingly. Numerical experiments conducted on both unimodal and multimodal benchmark functions demonstrate that the proposed mechanism effectively adapts the learning rates based on the tendencies of the updates of the distribution parameters. Furthermore, xNES equipped with the proposed mechanism achieved performance comparable to that of xNES with a carefully tuned fixed learning rate, eliminating the need for extensive parameter tuning. The proposed mechanism can be extended to high-dimensional problems. We believe that integrating the proposed learning rate adaptation mechanism into separable NES [81, 78] offers a straightforward approach to addressing such problems, making it a promising direction for future work.

However, this study has several limitations that should be addressed in future work. While we focused on proposing a simple and extensible learning rate adaptation mechanism, exhaustive experiments were not conducted. Thus, an important direction for future research is to evaluate the performance of the proposed mechanism across a broader range of benchmark problems and experimental settings.

Additionally, the current approach requires an approximation of the expected value of the KL divergence between updates in distribution parameters in updating the learning rate. Obtaining such an approximation may be challenging for more complex optimization algorithms (e.g., DX-NES [24]), as these methods often rely on search histories during optimization, which can invalidate the assumptions made during derivation. Relaxing these assumptions presents an interesting direction for future work.

The proposed method partially alleviates the burden of tuning the population size by automatically adapting the learning rate according to the given population size. However, it does not fully address the dependency on population size. In our experiments, we predefined a sufficient population size for solving the benchmark problems (e.g., selecting the population size $\lambda \in \{200, 250, 300, 350, 400\}$ for the Rastrigin function). In a black-box scenarios, such prior information is not available in general, necessitating tedious trial and error. Since our focus of this study is the optimization efficiency, this limitation was not addressed. A more thorough treatment of this issue will be provided in Chapter 4.

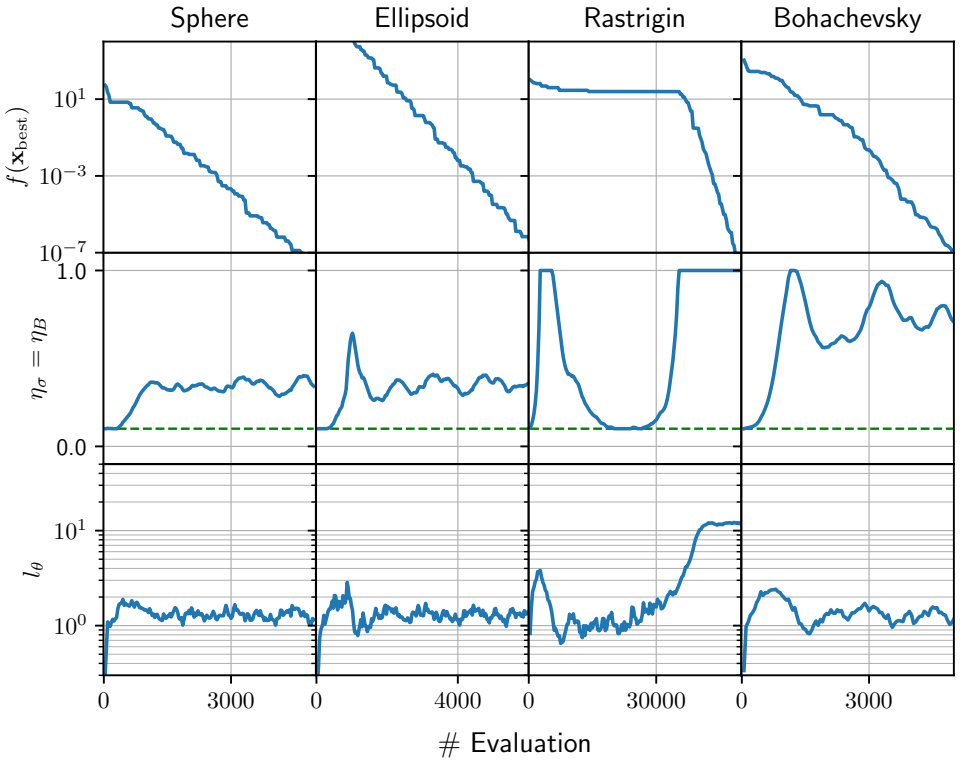


Figure 3.6: Typical behavior in xNES with the proposed learning rate adaptation mechanism on the 10-dimensional benchmark problems. The green dotted line in the learning rate graphs indicates the default value. The horizontal axis represents the number of evaluations. The vertical axes represent the best evaluation value $f(x_{\text{best}})$, the learning rates η_σ and η_B, and the length of the evolution path l_θ, respectively.

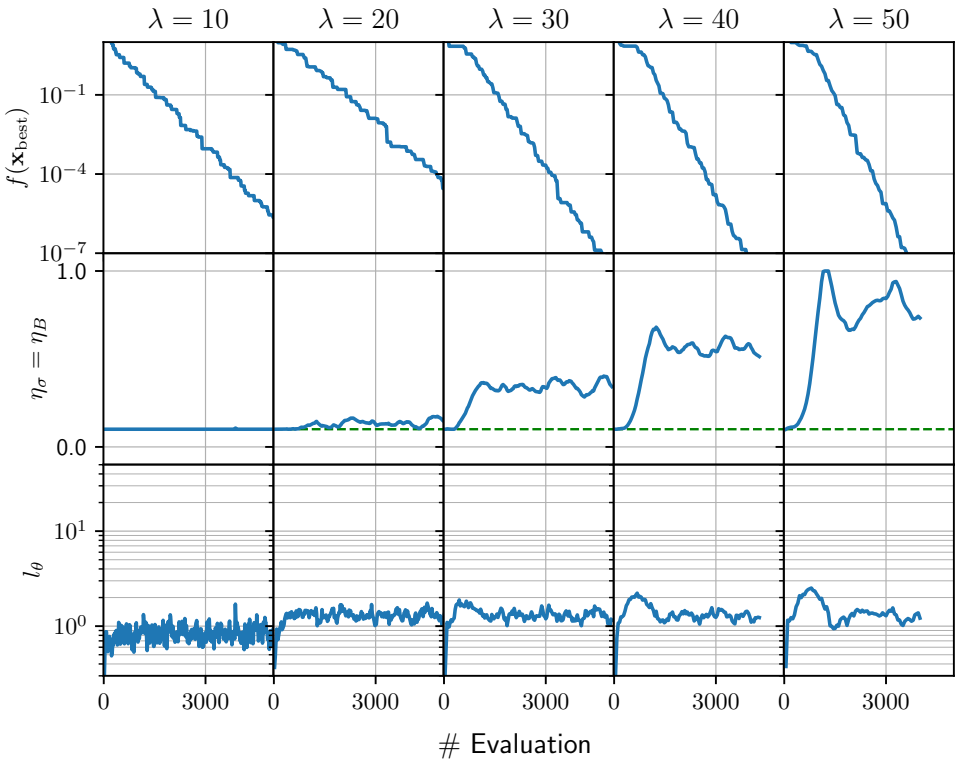


Figure 3.7: Typical behavior in xNES with the proposed learning rate adaptation mechanism on the 10-dimensional Sphere function. The experiment is performed with the population size $\lambda = 10, 20, 30, 40$, and 50. The green dotted line in the learning rate graphs indicates the default value.

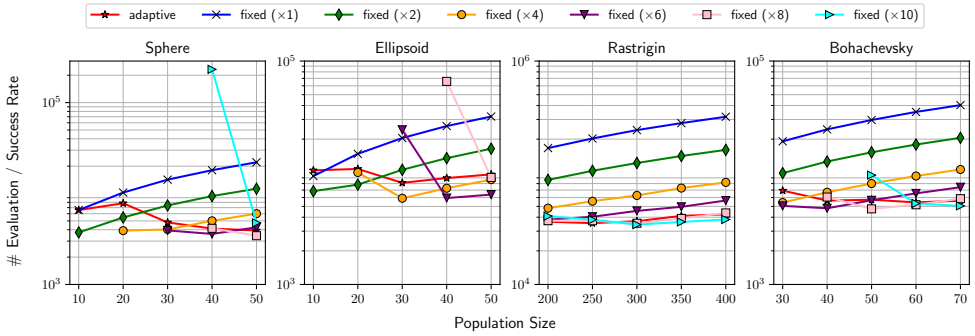


Figure 3.8: Performance comparison of xNES with the proposed learning rate adaptation mechanism (red) and xNES with the fixed learning rates (blue, green, yellow, purple, pink, and cyan) on 10-dimensional benchmark problems. The horizontal axis represents the population size. The vertical axis represents the average number of evaluations divided by the success rate, which is the smaller, the better it is. Note that, if no successful trials exist at a population size, nothing is plotted at the population size.

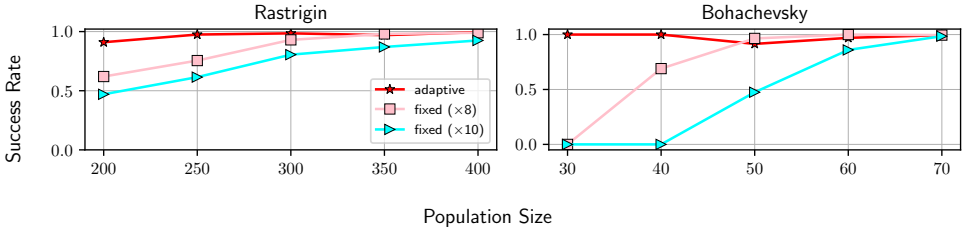


Figure 3.9: Success rate of xNES with the proposed learning rate adaptation method (red), xNES with the fixed learning rate of the default value times 8 (pink), and xNES with the fixed learning rate of the default value times 10 (cyan) in the multimodal functions.

Chapter 4

Learning Rate Adaptation for Multimodal and Noisy Problems

4.1 Introduction

CMA-ES is practically useful as it is a quasi-hyperparameter-free algorithm; practitioners can use it without hyperparameter tuning because default values are provided for all hyperparameters through theoretical analysis and extensive empirical evaluations. Specifically, the hyperparameter values are automatically computed using dimension d and population size λ, where $\lambda = 4 + \lfloor 3 \ln(d) \rfloor$ by default.

Although the default λ value works well for various unimodal problems, increasing it can help solve *difficult* tasks, such as solving multimodal and additive noise problems [37, 62, 64]. However, in a black-box scenario, determining the problem structure of f is challenging. Thus, determining the appropriate λ value in advance is also challenging, and online adaptation of λ has been proposed to address the issue [62, 64, 41, 61]. Population size adaptation (PSA)-CMA-ES [64] is a representative λ adaptation mechanism that has exhibited promising performance for difficult tasks, including multimodal and additive noise problems.

It has been observed that, in CMA-ES, increasing λ has an effect similar to decreasing the m learning rate, that is, η_m [59][1]. Indeed, the m and Σ learning rates, that is, η, is another hyperparameter that critically affects performance. An excessively large η value results in unstable parameter updates, whereas an

[1]Note that, in Ref. [59], the rank-one update was excluded from CMA-ES. In this study, however, we consider CMA-ES including the rank-one update.

excessively small value degrades search efficiency. Miyazawa and Akimoto [59] reported that CMA-ES with even a relatively small λ (e.g., $\lambda = \sqrt{d}$) solves multimodal problems through an appropriate setting of η. However, determining the appropriate η value is difficult in practice because prior knowledge is often limited and hyperparameter tuning entails expensive numerical investigations.

Therefore, online adaptation of η based on the problem difficulty constitutes an important advancement as it will allow practitioners to *safely* use CMA-ES without requiring prior knowledge or expensive trial-and-error calculations. In particular, we believe that η adaptation is more advantageous than λ adaptation from a practical perspective because the former is more suitable for parallel implementations. For example, practitioners often wish to specify a certain number of workers as the value of λ value to avoid wasting computational resources. However, λ adaptation may not always effectively utilize the available resources, as the values vary during the optimization process. By contrast, η adaptation allows complete exploitation of the available resources because the value of λ is fixed as the maximum number of workers. Moreover, in η adaptation, the parameters are regularly updated, whereas CMA-ES with λ adaptation does not progress until all λ solutions are evaluated, making it difficult to determine the search termination point.

Although online η adaptation itself is not new and several studies have attempted to adapt η values in CMA-ES variants, these adaptations targeted *speed-up* [69, 25, 57]. One notable exception is the η adaptation proposed by Krause [54] that aims to solve additive noise problems through new evolution strategies. However, it estimates the problem difficulty through resampling, that is, by repeatedly evaluating the *same* solution; thus, it is not suitable for solving (noiseless) multimodal problems. Furthermore, as it involves significant modifications of the internal parameters of the evolution strategies, applying it directly to CMA-ES is challenging.

This study aimed to develop CMA-ES to solve multimodal and additive noise problems without extremely expensive η tuning or adjusting any other CMA-ES parameters except η. To achieve this, we first examined the impact of learning rate. Our results suggest that (i) difficult problems can be relatively easily solved by decreasing the learning rate and aligning the parameter behavior with the trajectory of an ordinary differential equation (ODE), and (ii) the optimal learning rate is approximately proportional to the signal-to-noise ratio (SNR). Based on these observations, we propose an η adaptation mechanism for CMA-ES—called the learning rate adaptation (LRA)—that adapts η to maintain a constant SNR. The key feature of the proposed method is that it does not require specific

knowledge of the internal mechanism of the distribution-parameter update to estimate the SNR. Consequently, the proposed method is widely applicable to various CMA-ES variants, such as diagonal decoding (dd)-CMA [5], even though this study considers the most commonly used CMA-ES, which combines weighted recombination, step-size σ adaptation, rank-one update, and rank-μ update.

It should be noted that our work focuses on well-structured multimodal problems rather than weakly structured ones, as in the previous studies on λ adaptation in CMA-ES [64]. Using our method alone cannot solve the weakly structured multimodal problems and may even be detrimental to these problems. To address such problems, we believe the integration of restart strategies (e.g., BIPOP-CMA-ES [32]) is necessary, which is beyond this study; thus, we have left it for future work.

The remainder of this chapter is organized as follows: Section 4.2 closely examines and explains the impact of the learning rate and presents the discussion for determining the ideal learning rate. In particular, Section 4.2.4 explains why the learning rate adaptation described in Chapter 3 cannot be directly applied to this study. We believe this discussion highlights the need for a new learning rate adaptation mechanism to effectively address multimodal and noisy problems. Section 4.3 presents the proposed η adaptation mechanism based on SNR estimation. Section 4.4 evaluates the performance of the proposed η adaptation for noiseless and noisy problems. Finally, Section 4.5 concludes the chapter and suggests future research directions.

This chapter is based on the author's previous study [65, 66].

4.2 Learning Rate Impact

In this section, we discuss the impact of the learning rate on CMA-ES. First, Section 4.2.1 summarizes existing research on adjusting the population size, which is a common practice for difficult tasks, such as multimodal problems, and the relation between the population size and learning rate. In Section 4.2.2, we discuss the behavior from the perspective of ODEs for small learning rates. Consequently, we demonstrate that difficult problems can be solved by reducing the learning rate (i.e., closer to the solution of the ODE). However, it should be noted that an excessively small learning rate can reduce the search efficiency. Therefore, Section 4.2.3 discusses the determination of the optimal learning rate.

4.2.1 Relation Between Population Size and Learning Rate

Previous studies generally focused on increasing the population size λ to solve multimodal problems. Hansen and Kern [37] reported that CMA-ES with a sufficiently large population size can often solve multimodal problems with high probability. Based on this observation, Auger and Hansen [12] proposed IPOP-CMA-ES, which doubles the population size with each restart. Although these studies considered CMA-ES with default learning rates, Miyazawa and Akimoto [59] experimentally evaluated the performance of CMA-ES using small learning rates and showed that multimodal problems, such as the Rastrigin function, can be solved by setting sufficiently small learning rates *without* using a large population size. This empirical observation suggests that the effect of increasing the population size is similar to that of decreasing the learning rate.

Here, we organize the relation between the population size and learning rate more formally. First, we examine the relation based on the results of quality gain analysis. For the infinite-dimensional Sphere function, the optimal value of the normalized step-size $\bar{\sigma}^*$, whose normalized step-size is defined as $\bar{\sigma} := \sigma\eta_m d/\|m - x^*\| = O(\sigma\eta_m)$, is $\bar{\sigma}^* = -\mu_w \sum_{i=1}^{\lambda} w_i \mathbb{E}[\mathcal{N}_{i:\lambda}] \approx \sqrt{2/\pi}\lambda \in O(\lambda)$ [11, 3]. Hence, the optimal step-size is $\sigma^* \in O(\lambda/\eta_m)$, which clearly demonstrates that increasing λ corresponds to decreasing η_m. In other words, as the population size increases or learning rate decreases, the optimal step-size increases. Miyazawa and Akimoto [59] hypothesized that CMA-ES with small learning rates can solve multimodal problems owing to the effect of maintaining a large step-size.

Next, we offer another characterization of the relation between the population size and the learning rate, by viewing IGO algorithms as discretizations of stochastic differential equations (SDEs) [46]. For conciseness, we define the natural gradient $g(\theta) := \mathbb{E}_{x\sim p(x;\theta)}[W_\theta^f(x)\tilde{\nabla}_\theta \ln p(x;\theta)]$ and let its Monte Carlo estimation $\hat{g}^{(\lambda)}(\theta) := (1/\lambda)\sum_{i=1}^{\lambda} \hat{g}_i(\theta)$, where $\hat{g}_i(\theta) := W_\theta^f(x_i)\tilde{\nabla}_\theta \ln p(x_i;\theta)$, where $\hat{g}_i(\theta)$ is an unbiased estimator of $g(\theta)$. Note that, in practice, W_θ^f must also be estimated using the Monte Carlo method; thus, $\hat{g}_i(\theta)$ does not necessarily provide an unbiased estimation of $g(\theta)$. However, we assume the availability of W_θ^f for this discussion. Subsequently, we denote the covariance of $\hat{g}_i(\theta)$ as $S(\theta)$. By using this notation, the IGO update in Eq.(2.15) can be written as $\theta^{(t+1)} = \theta^{(t)} + \eta\hat{g}^{(\lambda)}(\theta^{(t)})$. Given a sufficiently large population size λ, the following is valid according to the central limit theorem:

$$\hat{g}^{(\lambda)}(\theta) \sim \mathcal{N}\left(g(\theta), \frac{1}{\lambda}S(\theta)\right). \tag{4.1}$$

Based on this result, we can rewrite Eq.(2.15) as follows:

$$\theta^{(t+1)} = \theta^{(t)} + \eta g(\theta^{(t)}) + \eta(\hat{g}^{(\lambda)}(\theta^{(t)}) - g(\theta^{(t)})), \tag{4.2}$$

where $\hat{g}^{(\lambda)}(\theta^{(t)}) - g(\theta^{(t)}) \sim \mathcal{N}(0, (1/\lambda)S(\theta))$. Hence, using the newly introduced random variable $\epsilon_\theta \sim \mathcal{N}(0, S(\theta))$, the IGO update can be rewritten as follows:

$$\theta^{(t+1)} = \theta^{(t)} + \eta g(\theta^{(t)}) + \frac{\eta}{\sqrt{\lambda}}\epsilon_{\theta^{(t)}}. \tag{4.3}$$

Consequently, we consider the following SDE:

$$d\theta = g(\theta)dt + \sqrt{\frac{\eta}{\lambda}}R(\theta)dW(t), \tag{4.4}$$

where $R(\theta)R(\theta)^\top = S(\theta)$ and $\{W(t)\}$ is the standard Wiener process. By discretizing the SDE using the Euler–Maruyama method [52], with the learning rate η, we obtain an equation identical to Eq. (4.3). Therefore, from the SDE perspective, the learning rate and the population size appear only in the form of the ratio η/λ, which implies that the effect of increasing λ is similar to that of decreasing η.

In summary, although previous studies primarily adjusted the population size for solving multimodal problems, we empirically and theoretically observed that increasing the population size and decreasing the learning rate have similar effects on the optimal step-size and noise.

4.2.2 Effect of Decreasing the Learning Rate from an ODE Perspective

When the learning rate approaches zero, the IGO algorithm is reduced to the following ODE [4]:

$$\frac{d\theta}{dt} = \mathbb{E}_{x\sim p(x;\theta)}[W_\theta^f(x)\tilde{\nabla}_\theta \ln p(x;\theta)]. \tag{4.5}$$

To illustrate the algorithm behavior from an ODE perspective, we consider minimizing the 1-dimensional Rastrigin function $f_{\text{Rastrigin}}(x) = 10 + x^2 - 10\cos(2\pi x)$, which is a well-structured multimodal problem (Fig. 3.1). Assuming that $W_\theta^f =$

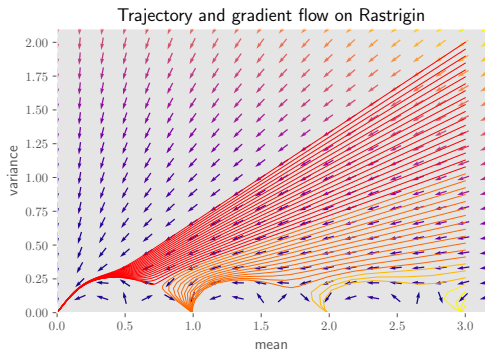


Figure 4.1: ODE trajectories and gradient flows of the Rastrigin function. The different colors (red, orange, yellow-orange, and yellow) of the ODE trajectories indicate different attractors.

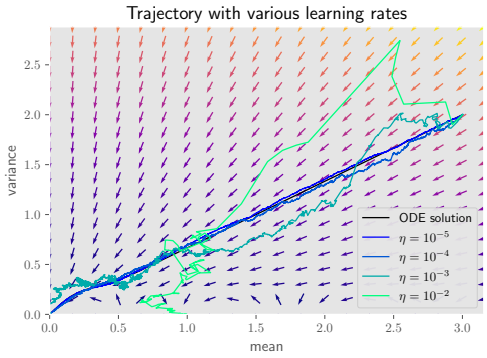


Figure 4.2: Typical parameter trajectories of the Rastrigin function under various learning rates ($\eta = 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}$). The ODE solution (black) is also illustrated for reference.

$-f$ and parameterizing our Gaussian distribution using $\theta = (m, v)$, where m is the mean and v is the variance, the ODEs are calculated as follows:

$$\frac{dm}{dt} = -2mv - 20\pi v \sin(2\pi m) \exp(-2\pi^2 v), \tag{4.6}$$

$$\frac{dv}{dt} = -2v^2 - 40\pi^2 v^2 \cos(2\pi m) \exp(-2\pi^2 v). \tag{4.7}$$

Figure 4.1 shows the ODE trajectories and gradient flows of the Rastrigin function. The experiments were conducted using initial distribution parameters $m = 3.0$ and $v \in [0.02, 2.0]$. It is evident that ODEs with large initial variances exhibit trajectories converging to the optimal solution $(m^*, v^*) = (0, 0)$. Given that the algorithm behavior tends to approach the trajectory of the corresponding ODE, as the learning rate decreases, we hypothesize that such multimodal problems can be solved by adequately decreasing the learning rate and employing a sufficiently large variance.

To verify this hypothesis, we evaluated the behavior of the distribution parameters for various learning rates. For this, we employed the following discretized versions of Eq. (4.6) and (4.7) using the Euler method:

$$m^{(t+1)} = m^{(t)} - \eta(2mv + 20\pi v \sin(2\pi m) \exp(-2\pi^2 v)), \tag{4.8}$$

$$v^{(t+1)} = v^{(t)} - \eta(2v^2 + 40\pi^2 v^2 \cos(2\pi m) \exp(-2\pi^2 v)), \tag{4.9}$$

where η denotes the learning rate; we used η values of $10^{-5}, 10^{-4}, 10^{-3}$, and 10^{-2}. The initial distribution parameters were set as $(m, v) = (3.0, 2.0)$. Figure 4.2 shows the typical behaviors of the parameter trajectories for various learning rates. It is evident that as the learning rate decreases, the corresponding trajectory approaches the ODE solution, which is also evident from the design of the Euler method. However, as the learning rate increases, the trajectory deviates from the ODE trajectory and tends to become trapped in the local optima, failing to find the optimal solution. These findings suggest the importance of setting a small learning rate for multimodal problems that can be solved by moving the distribution parameters along the ODE trajectory.

Although earlier discussions focused on multimodal problems, we believe that decreasing the learning rate is equally important for problems with unbiased additive noise, represented as $f(x) + \epsilon$, where ϵ is an unbiased random variable, that is, $\mathbb{E}[\epsilon] = 0$. This is because, in cases with unbiased noise, the corresponding ODE remains unchanged compared with noiseless ones. That is, by decreasing the learning rate and aligning the parameter updates with the corresponding ODE trajectory, the distribution-parameter value can be guided closer to the optimal solution.

4.2.3 Optimal Learning Rate

Although setting a small learning rate can be beneficial for solving multimodal and noisy problems, as discussed in Section 4.2.2, using an excessively small value can result in slow convergence. In this section, we explore the optimal value of the learning rate.

For simplicity, we consider the minimization of $\mathbb{E}[f(x)] = \int f(x)p(x;\theta)dx =: J(\theta)$ and assume that J is twice differentiable. Additionally, we let Δ be an unbiased estimator of $\tilde{\nabla}J(\theta)$. In this case, the one-step update is $\theta - \eta \cdot \Delta$. Using the Taylor approximation, we obtain the following:

$$J(\theta - \eta \cdot \Delta) = J(\theta) - \eta \nabla J(\theta)^\top \Delta + \frac{1}{2}\eta^2 \Delta^\top H \Delta + o(\eta^2 \|\Delta\|^2) \tag{4.10}$$

$$\approx J(\theta) - \eta \nabla J(\theta)^\top \Delta + \frac{1}{2}\eta^2 \Delta^\top H \Delta, \tag{4.11}$$

where $H := \nabla^2 J(\theta)$. Considering the expectations over Δ, we obtain the follow-

ing:

$$\mathbb{E}_\Delta[J(\theta - \eta \cdot \Delta)] \approx J(\theta) - \eta \nabla J(\theta)^\top \tilde{\nabla} J(\theta) + \frac{1}{2}\eta^2 \left(\tilde{\nabla} J(\theta)^\top H \tilde{\nabla} J(\theta) + \mathrm{Tr}(H \,\mathrm{Cov}[\Delta]) \right). \tag{4.12}$$

Thereafter, the expected improvement is approximated as follows:

$$J(\theta) - \mathbb{E}_\Delta[J(\theta - \eta \cdot \Delta)] \approx \eta \nabla J(\theta)^\top \tilde{\nabla} J(\theta) - \frac{1}{2}\eta^2 \left(\tilde{\nabla} J(\theta)^\top H \tilde{\nabla} J(\theta) + \mathrm{Tr}(H \,\mathrm{Cov}[\Delta]) \right). \tag{4.13}$$

By taking the derivative w.r.t. η and solving it for zero, we approximate the *optimal* learning rate as follows:

$$\nabla J(\theta)^\top \tilde{\nabla} J(\theta) - \eta \left(\tilde{\nabla} J(\theta)^\top H \tilde{\nabla} J(\theta) + \mathrm{Tr}(H \,\mathrm{Cov}[\Delta]) \right) = 0 \tag{4.14}$$

$$\therefore \eta^* \approx \frac{\nabla J(\theta)^\top \tilde{\nabla} J(\theta)}{\tilde{\nabla} J(\theta)^\top H \tilde{\nabla} J(\theta) + \mathrm{Tr}(H \,\mathrm{Cov}[\Delta])} \tag{4.15}$$

$$= \frac{\|\tilde{\nabla} J(\theta)\|_F^2}{\|\tilde{\nabla} J(\theta)\|_H^2 + \mathrm{Tr}(H \,\mathrm{Cov}[\Delta])}, \tag{4.16}$$

where F is the Fisher information matrix at θ, and $\|\tilde{\nabla} J(\theta)\|_M := (\tilde{\nabla} J(\theta)^\mathrm{T} M \tilde{\nabla} J(\theta))^{1/2}$ is the norm under M.

To obtain crucial insights into the determination of the optimal learning rate, we first assume $H \approx cF$ for a positive constant c. This assumption is partially relevant in scenarios where the covariance matrix of CMA-ES successfully learns the shape of a quadratic function. This concept can be illustrated as follows: For a function $f(x) = \frac{1}{2} x^\top A x$, the Hessian H is $\mathrm{diag}(A, 0)$. Consequently, given that $F = \mathrm{diag}(\Sigma^{-1}, \Sigma^{-1} \otimes \Sigma^{-1}/2)$, if $A \propto \Sigma^{-1}$, then, to a certain extent, the Hessian H in the m-part approximates cF for some c value; however, this does not apply to H in the Σ-part. Based on this assumption, the optimal learning rate can be written as

$$\eta^* \approx \frac{1}{c} \cdot \frac{1}{1 + \mathrm{SNR}^{-1}} \propto \frac{1}{1 + \mathrm{SNR}^{-1}}. \tag{4.17}$$

where $\mathrm{SNR} := \frac{\|\tilde{\nabla} J(\theta)\|_F^2}{\mathrm{Tr}(F \,\mathrm{Cov}[\Delta])}$. A high SNR increases the η^* value, which aligns with our intuitive expectations. In the next section, we propose an LRA mechanism based on these insights into the optimal learning rate.

4.2.4 Limitation of Learning Rate Adaptation Proposed in Chapter 3

One might wonder whether the learning rate adaptation method proposed in Chapter 3 can be applied to this study. However, to the best of our knowledge, the answer is not straightforward. In this section, we discuss the underlying reasons.

The goal of this chapter is to address difficult problems, such as multimodal and noisy optimization tasks. To tackle these problems effectively, enhancing the estimation accuracy of updates is critical, as elaborated in Section 4.2.2. We can easily see that the population size λ has a *direct* impact on the estimation accuracy of updates. For example, in noiseless scenarios, when sampling from an independent and identically distributed (i.i.d.) distribution, the standard deviation of the estimate decreases as $O(1/\sqrt{\lambda})$. In contrast, the learning rate does not directly influence the estimation accuracy of updates (at least on a single iteration). To establish such a connection, it is necessary to consider the concentration of updates over *multiple* iterations rather than focusing on a single iteration. This aspect will be discussed in detail in Section 4.3.1. This critical consideration was not addressed in the method described in Chapter 3. Consequently, we required a new learning rate adaptation approach that integrates this novel insight.

It should be noted that the learning rate adaptation proposed in Chapter 3 is not necessarily unusable in our case. For instance, when optimizing multimodal or noisy problems using the method from Chapter 3, the tendencies of updates in the distribution parameters weaken, leading to a decreased learning rate. Consequently, the method may exhibit behavior similar to that of the new approach introduced in this chapter. However, the method in Chapter 3 does not explicitly address estimation accuracy. In contrast, this chapter establishes a direct connection between the learning rate and estimation accuracy and addresses it explicitly in the proposed method. Thus, the method introduced in this chapter is more principled and sophisticated.

4.3 LRA Mechanism

We consider the updating of the distribution parameters $\theta_m = m$ and $\theta_\Sigma = \mathrm{vec}(\Sigma)$, where vec is the vectorization operator and $\Sigma = \sigma^2 C$ for the standard CMA-ES. Let $\Delta_m^{(t)} = m^{(t+1)} - m^{(t)}$ and $\Delta_\Sigma^{(t)} = \mathrm{vec}(\Sigma^{(t+1)} - \Sigma^{(t)})$ be the original updates

of m and Σ, respectively. Subsequently, we introduce the learning rate factors $\eta_m^{(t)}$ and $\eta_\Sigma^{(t)}$. The modified updates are performed as $\theta_m^{(t+1)} = \theta_m^{(t)} + \eta_m^{(t)}\Delta_m^{(t)}$ and $\theta_\Sigma^{(t+1)} = \theta_\Sigma^{(t)} + \eta_\Sigma^{(t)}\Delta_\Sigma^{(t)}$. Finally, $\eta_m^{(t)}$ and $\eta_\Sigma^{(t)}$ are adapted individually.

It is important to clarify that the learning rate η_Σ used in this chapter differs from its definition in Chapter 3. In this chapter, the learning rate is applied as a multiplier to the original update, which already incorporates the (original) learning rate implicitly. As a result, setting $\eta_\Sigma = 1$ in this chapter recovers the original CMA-ES. In contrast, in Chapter 3, the learning rate η_Σ directly corresponds to the learning rate included in the original update.

4.3.1 Main Concept

We adapt the learning rate factor η for the component θ (either $\theta_m = m$ or $\theta_\Sigma = \mathrm{vec}(\Sigma)$) of the distribution parameters based on the SNR of the update as follows:

$$\mathrm{SNR} := \frac{\|\mathbb{E}[\Delta]\|_F^2}{\mathrm{Tr}(F\,\mathrm{Cov}[\Delta])} = \frac{\|\mathbb{E}[\Delta]\|_F^2}{\mathbb{E}[\|\Delta\|_F^2] - \|\mathbb{E}[\Delta]\|_F^2}. \tag{4.18}$$

The Fisher metric is selected as it offers invariance against probability distribution parameterization. We attempt to adapt η such that $\mathrm{SNR} = \alpha\eta$, where $\alpha > 0$ is a hyperparameter that determines the target SNR.

The following rationale is employed for selecting this concept: We assume that η is sufficiently small such that the distribution parameters do not change significantly over n iterations. Thus, we assume $\theta^{(t+k)} \approx \theta^{(t)}$ for $k = 1, \ldots, n$. Subsequently, $\{\Delta^{(t+k)}\}_{k=0}^{n-1}$ are roughly considered as i.i.d. Hence, n steps of the update are as follows:

$$\theta^{(t+n)} = \theta^{(t)} + \eta \sum_{k=0}^{n-1} \Delta^{(t+k)} \tag{4.19a}$$

$$\approx \theta^{(t)} + \mathcal{D}\left(n\eta\mathbb{E}[\Delta], n\eta^2\,\mathrm{Cov}[\Delta]\right), \tag{4.19b}$$

where $\mathcal{D}(A, B)$ is a distribution with expectation A and (co)variance B. Thus, by setting a small η value and considering the results of $n = 1/\eta$ updates, we obtain an update that is more concentrated around the expected behavior than that expected for an update using $\eta = 1$. The expected change in θ over $n = 1/\eta$ iterations, measured using the squared Fisher norm, which approximates the Kullback–Leibler (KL) divergence between $\theta^{(t)}$ and $\theta^{(t+n)}$, is $\|\mathbb{E}[\Delta]\|_F^2 + \eta\,\mathrm{Tr}(F\,\mathrm{Cov}[\Delta])$, where the former and latter terms come from the signal and noise, respectively.

The SNR over n iterations is $\frac{\|\mathbb{E}[\Delta]\|_F^2}{\eta \operatorname{Tr}(F \operatorname{Cov}[\Delta])} = \frac{1}{\eta}\mathrm{SNR}$. Therefore, maintaining SNR = $\alpha\eta$ implies maintaining the SNR at α over $n = 1/\eta$ iterations, independent of η.

The rationale for using SNR can also be elucidated from the perspective of the optimal learning rate η^* derived in Section 4.2.3. The results showed that $\eta^* \propto 1/(1+\mathrm{SNR}^{-1})$ approximately holds under some assumptions. Additionally, we assume a relatively small SNR, for example, SNR $\lessapprox 1$ (this assumption is validated in Appendix B.3). In this case, the approximation $1/(1+\mathrm{SNR}^{-1}) \approx$ SNR is roughly valid. Thus, $\eta^* \propto$ SNR can be considered to be valid. As stated previously, we controlled η such that SNR $= \alpha\eta$. Consequently, this leads to $\eta \propto$ SNR, which is considered to be nearly optimal.

4.3.2 SNR Estimation

We estimate $\|\mathbb{E}[\Delta]\|^2$ and $\mathbb{E}[\|\Delta\|^2]$ for each component (m and Σ) using moving averages. We let $\mathcal{E}^{(0)} = \mathbf{0}$ and $\mathcal{V}^{(0)} = 0$, and update them as follows:

$$\mathcal{E}^{(t+1)} = (1-\beta)\mathcal{E}^{(t)} + \beta\tilde{\Delta}^{(t)}, \tag{4.20a}$$

$$\mathcal{V}^{(t+1)} = (1-\beta)\mathcal{V}^{(t)} + \beta\|\tilde{\Delta}^{(t)}\|_2^2, \tag{4.20b}$$

where β is a hyperparameter; $\tilde{\Delta}^{(t)}$ is the update at iteration t in the local coordinate at which the F at $\theta^{(t)}$ becomes the identity; $\|\cdot\|_2$ is the ℓ_2-norm. Thereafter, $\frac{2-\beta}{2-2\beta}\|\mathcal{E}\|_2^2 - \frac{\beta}{2-2\beta}\mathcal{V}$ and $\mathcal{V}$ are considered estimates of $\|\mathbb{E}[\Delta]\|_2^2$ and $\mathbb{E}[\|\Delta\|_2^2]$, respectively (the derivation is included in Appendix B.1).

The rationale for our estimators is as follows. Suppose that η_m and η_Σ are sufficiently small for us to assume that the parameters m and Σ do not change significantly over n iterations. Subsequently, the $\tilde{\Delta}^{(t+i)}$ $(i = 0, .., n-1)$ are considered to be located on the same local coordinates and distributed independently and identically. Then, ignoring the $(1-\beta)^n$ terms, we obtain

$$\mathcal{E}^{(t+n)} \sim \mathcal{D}\left(\mathbb{E}[\tilde{\Delta}], \frac{\beta}{2-\beta}\operatorname{Cov}[\tilde{\Delta}]\right). \tag{4.21}$$

(Again, the derivation is presented in Appendix B.1.) Thus, we have $\mathbb{E}[\|\mathcal{E}\|_2^2] \approx \|\mathbb{E}[\tilde{\Delta}]\|_2^2 + \frac{\beta}{2-\beta}\operatorname{Tr}(\operatorname{Cov}[\tilde{\Delta}])$. Similarly, it is apparent that $\mathbb{E}[\mathcal{V}] \approx \mathbb{E}[\|\tilde{\Delta}\|_2^2] = \|\mathbb{E}[\tilde{\Delta}]\|_2^2 + \operatorname{Tr}(\operatorname{Cov}[\tilde{\Delta}])$.

The SNR is then estimated as

$$\mathrm{SNR} := \frac{\|\mathbb{E}[\tilde{\Delta}]\|^2}{\mathrm{Tr}(\mathrm{Cov}[\tilde{\Delta}])} = \frac{\|\mathbb{E}[\tilde{\Delta}]\|^2}{\mathbb{E}[\|\tilde{\Delta}\|^2] - \|\mathbb{E}[\tilde{\Delta}]\|^2} \tag{4.22a}$$

$$\approx \frac{\|\mathcal{E}\|_2^2 - \frac{\beta}{2-\beta}\mathcal{V}}{\mathcal{V} - \|\mathcal{E}\|_2^2} =: \widehat{\mathrm{SNR}}. \tag{4.22b}$$

4.3.3 Learning Rate Factor Adaptation

We attempt to adapt η such that $\widehat{\mathrm{SNR}} = \alpha\eta$, where $\alpha > 0$ is the hyperparameter. This adaptation is expressed as follows:

$$\eta \leftarrow \eta \exp\left(\min(\gamma\eta, \beta)\Pi_{[-1,1]}\left(\frac{\widehat{\mathrm{SNR}}}{\alpha\eta} - 1\right)\right), \tag{4.23}$$

where $\Pi_{[-1,1]}$ is the projection onto $[-1, 1]$ and γ is a hyperparameter. If $\widehat{\mathrm{SNR}} > \alpha\eta$, η increases, and vice versa. Owing to these feedback mechanisms, $\widehat{\mathrm{SNR}}/(\alpha\eta)$ is expected to remain near 1. In the above expression, the projection $\Pi_{[-1,1]}$ is introduced to prevent a significant change in η during an iteration, and the damping factor $\min(\gamma\eta, \beta)$ is introduced because of the following reasons. First, the factor β is introduced to allow for the effect of the change in the previous η value to appear in $\widehat{\mathrm{SNR}}$. Second, the factor $\gamma\eta$ is introduced to prevent the η value from changing more than $\exp(\gamma)$ or $\exp(-\gamma)$ over $1/\eta$ iterations. Based on the η update through Eq. (4.23), the upper bound is set to 1 using $\eta \leftarrow \min(\eta, 1)$, to prevent unstable behavior. Although allowing η values >1 would accelerate the optimization, we do not consider this because we aim to safely solve difficult problems. Extending this method for acceleration represents an intriguing avenue for future work, as elaborated in Section 5.2.

4.3.4 Local Coordinate-System Definition

Although we estimate the SNR based on the updates $\Delta^{(\cdot)}$, naïvely accumulating these updates $\Delta^{(\cdot)}$ may result in unintentional behavior, as illustrated in the following example. Consider a scenario wherein $p(x; \theta^{(t)}) = \mathcal{N}(0, 100I)$, $p(x; \theta^{(t+1)}) = \mathcal{N}(0, 50I)$, and $p(x; \theta^{(t+2)}) = \mathcal{N}(0, 25I)$. In this case, the covariance matrix of the distribution decreases at a constant rate. Consequently, each KL divergence is $D_{\mathrm{KL}}(p(x; \theta^{(t)})||p(x; \theta^{(t+1)})) = D_{\mathrm{KL}}(p(x; \theta^{(t+1)})||p(x; \theta^{(t+2)}))$. This

implies that the distribution is moving at a uniform pace in terms of the KL divergence. However, the updates are $\mathrm{vec}^{-1}(\Delta_\Sigma^{(t)}) = -50I$ and $\mathrm{vec}^{-1}(\Delta_\Sigma^{(t+1)}) = -25I$, whose scales are different.[2] Thus, accumulating these effects will result in unintentional behavior.

To address these issues, we ensure parameterization invariance by defining the local coordinate system [62, 64] such that the Fisher information matrices, F_m and F_Σ, corresponding to each component of the distribution parameters, m and Σ, respectively, are the identity matrices. It is well-known that $F_m = \Sigma^{-1}$ and $F_\Sigma = 2^{-1}\Sigma^{-1} \otimes \Sigma^{-1}$, and their square roots are $\sqrt{F_m} = \sqrt{\Sigma}^{-1}$ and $\sqrt{F_\Sigma} = 2^{-\frac{1}{2}}\sqrt{\Sigma}^{-1} \otimes \sqrt{\Sigma}^{-1}$. Therefore, we define

$$\tilde{\Delta}_m = \sqrt{\Sigma}^{-1}\Delta_m, \tag{4.24a}$$

$$\tilde{\Delta}_\Sigma = 2^{-\frac{1}{2}}\,\mathrm{vec}(\sqrt{\Sigma}^{-1}\,\mathrm{vec}^{-1}(\Delta_\Sigma)\sqrt{\Sigma}^{-1}). \tag{4.24b}$$

Actually, in the previous example, the local coordinate system allows us to easily verify that $\mathrm{vec}^{-1}(\tilde{\Delta}_\Sigma^{(t)}) = \mathrm{vec}^{-1}(\tilde{\Delta}_\Sigma^{(t+1)})$. This observation aligns with intuitive expectations in view of the KL divergence and suggests the validity of accumulating the updates $\tilde{\Delta}$ instead of the original Δ.

4.3.5 Covariance Matrix Decomposition

After updating the covariance matrix $\Sigma^{(t+1)} = \Sigma^{(t)} + \eta_\Sigma^{(t)}\,\mathrm{vec}^{-1}(\Delta_\Sigma^{(t)})$, it must be split into σ and C. For this, we adopt the following strategy:

$$\sigma^{(t+1)} = \det(\Sigma^{(t+1)})^{\frac{1}{2d}}, \tag{4.25a}$$

$$C^{(t+1)} = (\sigma^{(t+1)})^{-2}\Sigma^{(t+1)}. \tag{4.25b}$$

4.3.6 Step-size Correction

Updating the learning rate for the m, i.e., η_m, changes the appropriate σ. Through a quality gain analysis that analyzed the expected f value improvement in a single step, a previous study [3] demonstrated that the optimal σ value is proportional to $1/\eta_m$ for infinite-dimensional convex quadratic functions. Therefore, to maintain the optimal σ value under η_m variations, we correct σ after each η_m

[2]Here, we have corrected the minor typos that appeared in [66].

update as follows:

$$\sigma^{(t+1)} \leftarrow \frac{\eta_m^{(t)}}{\eta_m^{(t+1)}} \sigma^{(t+1)}. \tag{4.26}$$

4.3.7 Overall Procedure

Algorithm 2 presents the overall LRA-CMA-ES procedure. At Line 2, the old parameters $m^{(t)}$, $\sigma^{(t)}$, and $C^{(t)}$ are input into CMA($\cdot$), which outputs new parameters $m^{(t+1)}$, $\sigma^{(t+1)}$, and $C^{(t+1)}$ by executing Steps 1–3 described in Section 2.3.

Note that the internal parameters such as the evolution paths p_σ and p_c, are updated and stored in CMA($\cdot$). However, these values were omitted for simplicity. The subscript $\cdot_{\{m,\Sigma\}}$ (e.g., as in $\eta_{\{m,\Sigma\}}$) indicates that there are parameters for m and Σ, respectively. For example, $\mathcal{E}_{\{m,\Sigma\}}^{(t+1)} \leftarrow (1-\beta_{\{m,\Sigma\}})\mathcal{E}_{\{m,\Sigma\}}^{(t)} + \beta_{\{m,\Sigma\}}\tilde{\Delta}_{\{m,\Sigma\}}^{(t)}$ is an abbreviation for the following two update equations: $\mathcal{E}_m^{(t+1)} \leftarrow (1-\beta_m)\mathcal{E}_m^{(t)} + \beta_m\tilde{\Delta}_m^{(t)}$ and $\mathcal{E}_\Sigma^{(t+1)} \leftarrow (1-\beta_\Sigma)\mathcal{E}_\Sigma^{(t)} + \beta_\Sigma\tilde{\Delta}_\Sigma^{(t)}$.

4.4 Experiments and Discussions

This study included various experiments to investigate the following research questions (RQs):

RQ1. Does the η adaptation in LRA-CMA-ES behave appropriately in accordance with the problem structure?

RQ2. Can LRA-CMA-ES solve multimodal and noisy problems even though a default λ value is used? How does its efficiency compare to that of CMA-ES with a fixed η value?

RQ3. How does the performance change with changes in LRA-CMA-ES hyperparameters?

RQ4. How does the performance depend on the population size λ?

RQ5. What are the differences in the performances of LRA-CMA-ES, which adapts the learning rate, and PSA-CMA-ES [64], which adapts the population size?

The remainder of this section is organized as follows. The experimental setups are described in Section 4.4.1. Section 4.4.2 demonstrates η adaptation in

Algorithm 2 LRA-CMA-ES

Input: $m^{(0)} \in \mathbb{R}^d, \sigma^{(0)} \in \mathbb{R}_{>0}, \lambda \in \mathbb{N}, \alpha, \beta_{\{m,\Sigma\}}, \gamma \in \mathbb{R}$

Set: $t = 0, C^{(0)} = I, \eta^{(0)}_{\{m,\Sigma\}} = 1, \mathcal{E}^{(0)} = \mathbf{0}, \mathcal{V}^{(0)} = 0$

while stopping criterion not met **do**

$m^{(t+1)}, \sigma^{(t+1)}, C^{(t+1)} \leftarrow \mathrm{CMA}(m^{(t)}, \sigma^{(t)}, C^{(t)})$

// calculate parameter one-step differences

$\Delta_m^{(t)} \leftarrow m^{(t+1)} - m^{(t)}$

$\Sigma^{(t+1)} \leftarrow \left(\sigma^{(t+1)}\right)^2 C^{(t+1)}$

$\Delta_\Sigma^{(t)} \leftarrow \mathrm{vec}\left(\Sigma^{(t+1)} - \Sigma^{(t)}\right)$

// local coordinate

$\tilde{\Delta}_m^{(t)} \leftarrow \sqrt{\Sigma^{(t)}}^{-1} \Delta_m^{(t)}$

$\tilde{\Delta}_\Sigma^{(t)} \leftarrow 2^{-1/2} \mathrm{vec}\left(\sqrt{\Sigma^{(t)}}^{-1} \mathrm{vec}^{-1}\left(\Delta_\Sigma^{(t)}\right) \sqrt{\Sigma^{(t)}}^{-1}\right)$

// update evolution paths and estimate SNR

$\mathcal{E}^{(t+1)}_{\{m,\Sigma\}} \leftarrow (1 - \beta_{\{m,\Sigma\}}) \mathcal{E}^{(t)}_{\{m,\Sigma\}} + \beta_{\{m,\Sigma\}} \tilde{\Delta}^{(t)}_{\{m,\Sigma\}}$

$\mathcal{V}^{(t+1)}_{\{m,\Sigma\}} \leftarrow (1 - \beta_{\{m,\Sigma\}}) \mathcal{V}^{(t)}_{\{m,\Sigma\}} + \beta_{\{m,\Sigma\}} \| \tilde{\Delta}^{(t)}_{\{m,\Sigma\}} \|_2^2$

$\widehat{\mathrm{SNR}}_{\{m,\Sigma\}} \leftarrow \frac{\| \mathcal{E}^{(t+1)}_{\{m,\Sigma\}} \|_2^2 - \frac{\beta_{\{m,\Sigma\}}}{2-\beta_{\{m,\Sigma\}}} \mathcal{V}^{(t+1)}_{\{m,\Sigma\}}}{\mathcal{V}^{(t+1)}_{\{m,\Sigma\}} - \| \mathcal{E}^{(t+1)}_{\{m,\Sigma\}} \|_2^2}$

// update learning rates

$\eta^{(t+1)}_{\{m,\Sigma\}} \leftarrow \eta^{(t)}_{\{m,\Sigma\}}$

$\cdot \exp\left(\min(\gamma \eta^{(t)}_{\{m,\Sigma\}}, \beta_{\{m,\Sigma\}}) \Pi_{[-1,1]}\left(\frac{\widehat{\mathrm{SNR}}_{\{m,\Sigma\}}}{\alpha \eta_{\{m,\Sigma\}}} - 1\right)\right)$

$\eta^{(t+1)}_{\{m,\Sigma\}} \leftarrow \min(\eta^{(t+1)}_{\{m,\Sigma\}}, 1)$

// update parameters with adaptive learning rates

$m^{(t+1)} \leftarrow m^{(t)} + \eta_m^{(t+1)} \Delta_m^{(t)}$

$\Sigma^{(t+1)} \leftarrow \Sigma^{(t)} + \eta_\Sigma^{(t+1)} \mathrm{vec}^{-1}(\Delta_\Sigma^{(t)})$

// decompose Σ to σ and C

$\sigma^{(t+1)} \leftarrow \det(\Sigma^{(t+1)})^{\frac{1}{2d}}, C^{(t+1)} \leftarrow (\sigma^{(t+1)})^{-2} \Sigma^{(t+1)}$

// σ correction

$\sigma^{(t+1)} \leftarrow \sigma^{(t+1)} (\eta_m^{(t)} / \eta_m^{(t+1)})$

$t \leftarrow t + 1$

end while

LRA-CMA-ES for noiseless and noisy problems (**RQ1**). Section 4.4.3 compares LRA-CMA-ES with CMA-ES with fixed η values (**RQ2**). Section 4.4.4 investigates the effects of LRA-CMA-ES hyperparameters (**RQ3**). Additional experimental results for the hyperparameters are presented in Appendix B.5. Section 4.4.5 evaluates the performance differences under various different population sizes (**RQ4**). Finally, Section 4.4.6 compares LRA-CMA-ES with PSA-CMA-ES (**RQ5**). Our code is available at GitHub[3][4].

4.4.1 Experimental Setups

The benchmark problem definitions and initial distributions are presented in Table 4.1 and Table 4.2. In each case (except for the Rosenbrock function), the global optimal solution is at $x = 0$. However, for the Rosenbrock function, it is at $x = 1$. As unimodal problems, we employ the Sphere, Ellipsoid, and Rosenbrock functions. The reason for using unimodal problems is to ensure that the performance of LRA-CMA-ES does not degrade significantly compared to CMA-ES with the default learning rate. The Ellipsoid function is an ill-conditioned problem, whereas the Rosenbrock function has dependencies between variables. Although the Rosenbrock function has local minima, in our study, it can be regarded as an almost unimodal problem. As well-structured multimodal problems, we employ the Ackley, Schaffer, Rastrigin, Bohachevsky, and Griewank functions. These problems are composed of a quadratic convex function and oscillatory nonconvex function, which resembles the noise. In the Ackley, Rastrigin, Bohachevsky, and Griewank functions, the oscillatory function is added to the convex function, whereas in the Schaffer function, the oscillatory function affects the convex function multiplicatively. This causes the Schaffer function to have fine oscillations around the optimal solution. Noteworthy, the optimization for the Griewank function gets easier as the dimension increases [37]. Similar to [37], we imposed additional bounds on the Ackley function. For noisy problems, we considered an additive Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ with σ_n^2 variance. It is worth noting that the proposed method maintains the affine invariance of CMA-ES because LRA-CMA-ES does not rely on a specific parameterization, as discussed in Section 4.3.4. Consequently, although many of the employed benchmark problems are separable, the experimental results obtained from a benchmark function can still be generalized to the experimental results of its rotated

[3] https://github.com/nomuramasahir0/cma-learning-rate-adaptation
[4] https://github.com/CyberAgentAILab/cmaes [71]

Table 4.1: Definitions of benchmark problems used in the experiments described in Chapter 4. For ease of reference, we have reproduced even the benchmark problems that overlap with Table 3.1.

Definitions
$f_{\text{Sphere}}(x) = \sum_{i=1}^{d} x_i^2$
$f_{\text{Ellipsoid}}(x) = \sum_{i=1}^{d} (1000^{\frac{i-1}{d-1}} x_i)^2$
$f_{\text{Rosenbrock}}(x) = \sum_{i=1}^{d-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$
$f_{\text{Ackley}}(x) = 20 - 20 \cdot \exp(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}) + e - \exp(\frac{1}{d}\sum_{i=1}^{d} \cos(2\pi x_i))$
$f_{\text{Schaffer}}(x) = \sum_{i=1}^{d-1} (x_i^2 + x_{i+1}^2)^{0.25} \cdot [\sin^2(50 \cdot (x_i^2 + x_{i+1}^2)^{0.1}) + 1]$
$f_{\text{Rastrigin}}(x) = 10d + \sum_{i=1}^{d} (x_i^2 - 10\cos(2\pi x_i))$
$f_{\text{Bohachevsky}}(x) = \sum_{i=1}^{d-1} (x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7)$
$f_{\text{Griewank}}(x) = \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos(x_i/\sqrt{i}) + 1$

version.

In all the experiments (except for those in Section 4.4.5), we set the default $\lambda = 4 + \lfloor 3 \ln d \rfloor$. The result when changing λ can be found in Ref. [37] and Section 4.4.5. For the dimension d, we employed $d \in \{10, 20, 30, 40\}$ for noiseless problems and $d = 10$ for noisy problems. Additionally, we set the LRA-CMA-ES hyperparameters as $\alpha = 1.4$, $\beta_m = 0.1$, $\beta_\Sigma = 0.03$, and $\gamma = 0.1$ based on preliminary experiments. As noted above, Section 4.4.4 presents an analysis of the hyperparameters sensitivity. [5] The values of other internal parameters of CMA-ES were set to those recommended in [35].

4.4.2 Learning Rate Behavior

Figure 4.3 shows the typical LRA-CMA-ES behaviors for noiseless problems, wherein η_Σ maintained relatively large values for the Sphere function. However, it exhibits significantly smaller values for the Ellipsoid and Rosenbrock functions. We believe that this behavior is undesirable, because the default η value already works well for these unimodal problems. Although η can be increased by changing the hyperparameters of the proposed η adaptation, this change may be detrimental for multimodal problems.

[5] While this setting was chosen to ensure stable performance of LRA-CMA-ES, a more effective configuration may be achievable if the problem structure is known; see Appendix B.4 for details.

Table 4.2: Initial distributions for each benchmark problem.

Functions	Initial Distributions
Sphere	$m^{(0)} = [3, \ldots, 3], \sigma^{(0)} = 2$
Ellipsoid	$m^{(0)} = [3, \ldots, 3], \sigma^{(0)} = 2$
Rosenbrock	$m^{(0)} = [0, \ldots, 0], \sigma^{(0)} = 0.1$
Ackley	$m^{(0)} = [15.5, \ldots, 15.5], \sigma^{(0)} = 14.5$
Schaffer	$m^{(0)} = [55, \ldots, 55], \sigma^{(0)} = 45$
Rastrigin	$m^{(0)} = [3, \ldots, 3], \sigma^{(0)} = 2$
Bohachevsky	$m^{(0)} = [8, \ldots, 8], \sigma^{(0)} = 7$
Griewank	$m^{(0)} = [305, \ldots, 305], \sigma^{(0)} = 295$

It is evident that η_m is slightly smaller for multimodal problems than for unimodal problems. Particularly, for the Rastrigin function, η_m and η_Σ clearly decrease at the beginning of the optimization, which reflects the difficulty of multimodal problem optimization. Subsequently, η increases as optimization becomes as easy as that for a unimodal problem. This behavior demonstrates that LRA-CMA-ES can adapt η according to the search difficulty.

Figure 4.4 shows the typical η adaptation behavior for noisy problems. The noise has a negligible effect in the early stages; thus, the η behavior for noisy problems is similar to that for noiseless problems. However, as the optimization proceeds and the function value approaches the same scale as that of the noise value, the noise starts having a critical effect. Consequently, the η value decreases. This adaptation ensures that the SNR remains constant. Notably, similar behavior can be observed for the noisy Rastrigin function, which features both noise and multimodality.

4.4.3 Effects of LRA

Figures 4.5 and 4.6 show the performances of LRA-CMA-ES and that of CMA-ES with a fixed learning rate ($\eta_m, \eta_\Sigma \in \{10^0, 10^{-1}, 10^{-2}\}$) for the noiseless problems. Note that CMA-ES with $\eta_m = 1.0$ and $\eta_\Sigma = 1.0$ is the original CMA-ES with the default η value. Each trial was considered successful if $f(m)$ reached the target value 10^{-8} before 10^7 function evaluations or before a numerical error occurred because of an excessively small σ. In addition to the success rate, we employed the SP1 value [13, 12], which is the average number of evaluations among successful trials until achieving the target value divided by the success

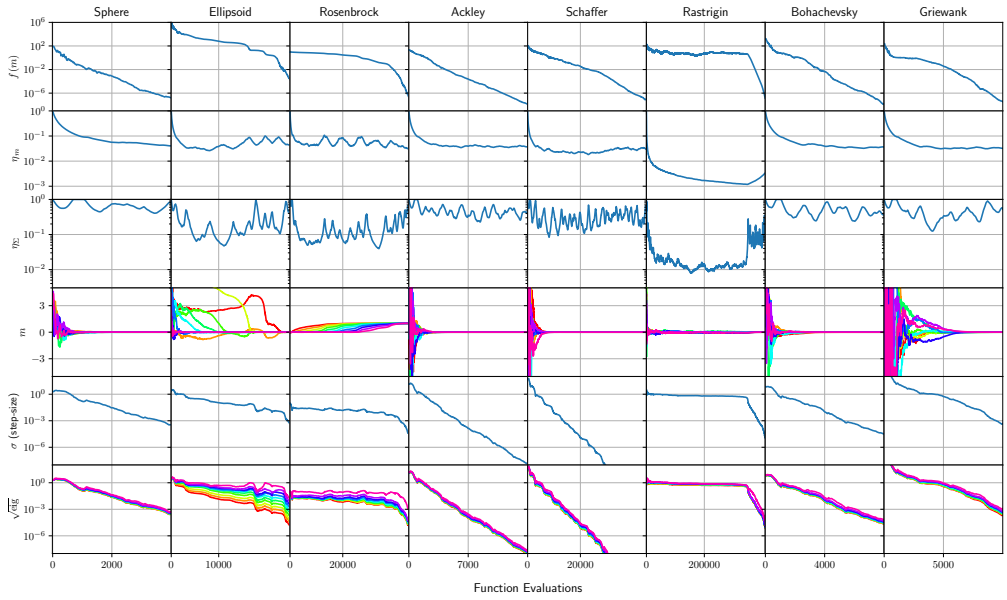


Figure 4.3: Typical LRA-CMA-ES behaviors for 10-dimensional (10-D) noiseless problems. The coordinates of m and the square roots of the eigenvalues of $\sigma^2 C$ (denoted by $\sqrt{\text{eig}}$) are indicated with different colors.

rate. 30 trials were conducted for each setting.

To compare the performances of these strategies for the noisy problems, we employed the empirical cumulative density function (ECDF) of COCO, a platform for comparing continuous optimizers in a black-box setting [36]. Using N_{target} target values, we recorded the number of evaluations until $f(m)$ (noiseless) reached each target value for the first time, and set the maximum function evaluation to 10^8. We collected data by running N_{trial} independent trials, and obtained a total of $N_{\text{target}} \cdot N_{\text{trial}}$ targets for each problem. Thereafter, we set the target values to $10^{6-9(i-1)/(N_{\text{target}}-1)}$ for $i = 1, \ldots, N_{\text{target}}$, with $N_{\text{target}} = 30$. By executing $N_{\text{trial}} = 20$ trials, 600 targets were obtained for each problem. Figure 4.7 shows the target value percentages obtained for each number of evaluations.

Noiseless Problems

We compared the success rates of LRA-CMA-ES and CMA-ES with fixed η values, as shown in Figure 4.5. For the multimodal problems, CMA-ES with a large η often failed to reach the optimum. However, CMA-ES with a small η exhibited a

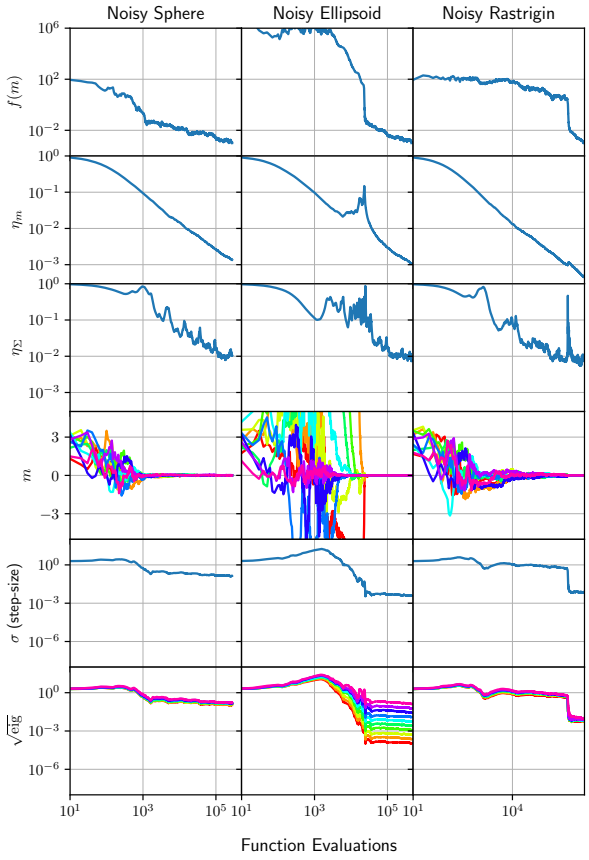


Figure 4.4: Typical LRA-CMA-ES behaviors for 10-D noisy problems. The noise variance σ_n^2 was set to 1.

high success rate, indicating a clear dependence on η. By contrast, LRA-CMA-ES exhibited a relatively good success rate, even though no η tuning was required. It is noteworthy that LRA-CMA-ES succeeded in all trials for the Rastrigin function even though the default population size (e.g., $\lambda = 15$ for $d = 40$) was used and η was not tuned in advance.

However, LRA-CMA-ES performance for the Schaffer function degraded at $d = 40$. From the results indicating that CMA-ES with an appropriately tuned, small η achieved a relatively high success rate, the LRA-CMA-ES result may have been obtained because η was not appropriately adapted in that case. This will be investigated in future work.

Figure 4.6 shows the SP1 results for LRA-CMA-ES and CMA-ES with fixed

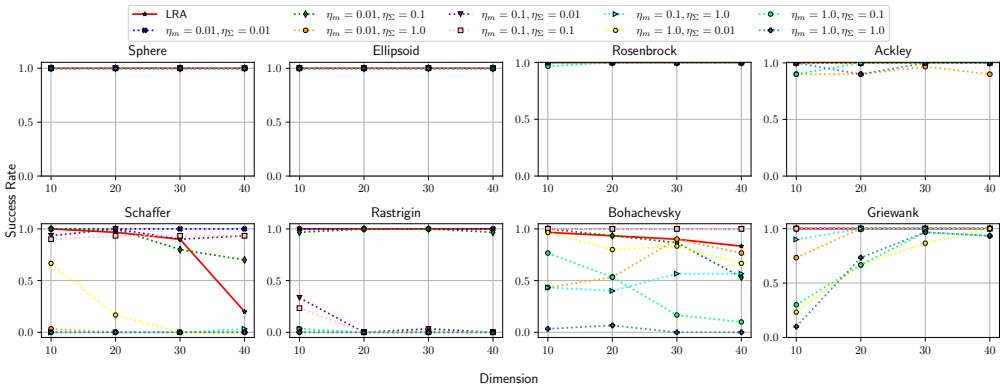


Figure 4.5: Success rates according to the number of dimensions (noiseless problems).

η values. CMA-ES with the default η values ($\eta_m = 1.0, \eta_\Sigma = 1.0$) outperformed the other methods for unimodal problems; however, the performance degraded significantly for multimodal problems owing to optimization failures. By contrast, the CMA-ES with a small η sometimes exhibited good performance for such multimodal problems; however, it was not efficient for unimodal and relatively easy multimodal problems. Therefore, for CMA-ES with a fixed η value, a clear trade-off in efficiency exists based on the η setting. By contrast, LRA-CMA-ES exhibited stable and relatively good performance for unimodal and multimodal problems. Again, η was not tuned, which is significantly expensive in practice. There is scope for improvement of the LRA-CMA-ES performance on unimodal problems; however, the current sub-par performance can be somewhat mitigated by changing the hyperparameters. The effects of the hyperparameters are discussed in Section 4.4.4.

Noisy Problems

Figure 4.7 shows the ECDF results for both LRA-CMA-ES and CMA-ES with fixed η values. We considered two noise strengths, weak and strong, that is, $\sigma_n^2 = 1$ and 10^6, respectively.

Under the weak-noise setting, CMA-ES with a small η value reached all the target values. By contrast, CMA-ES with a large η value failed to approach the global optimum and yielded a sub-optimal solution. LRA-CMA-ES achieved similar performance to CMA-ES with a small η value without tuning. However, under the strong-noise setting, even CMA-ES with a small η stopped improv-

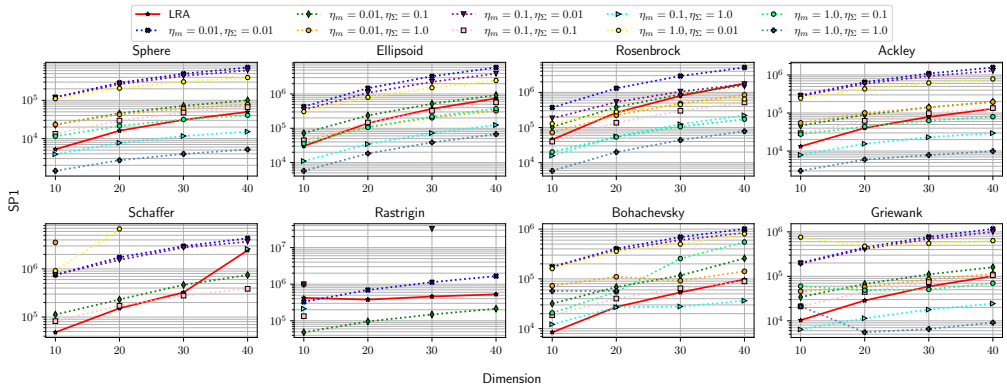


Figure 4.6: SP1 values according to the number of dimensions (noiseless problems). A missing point indicates the algorithm's failure in all trials.

ing the f value before reaching the global optimum. By contrast, LRA-CMA-ES continued improving the f value. Notably, the results for the noisy Rastrigin function suggest that LRA-CMA-ES can simultaneously handle both noise and multimodality.

4.4.4 Effects of Hyperparameters

Figure 4.8 shows the success rates and SP1 values with respect to α for the 30-dimensional (30-D) noiseless Sphere, Schaffer, and Rastrigin functions. For the Sphere function, a lower SP1 value could be achieved with a smaller α value. However, an excessively large α results in optimization failures for multimodal problems. Therefore, the current setting of $\alpha = 1.4$ seems reasonable; however, further investigations are required.

Figure 4.9 shows the success rates and SP1 values with respect to β_Σ. We observe that an excessively large β_Σ causes optimization failures in the Rastrigin function. Conversely, an excessively small β_Σ results in slow convergence. An additional result ($\beta_\Sigma \in \{0.01, 0.02, ..., 0.05\}$) is presented in Appendix B.5.

We also conducted similar experiments on the hyperparameters β_m and γ, to confirm their effects. These hyperparameters mildly impacted the overall performance compared to α and β_Σ (these results are also presented in Appendix B.5).

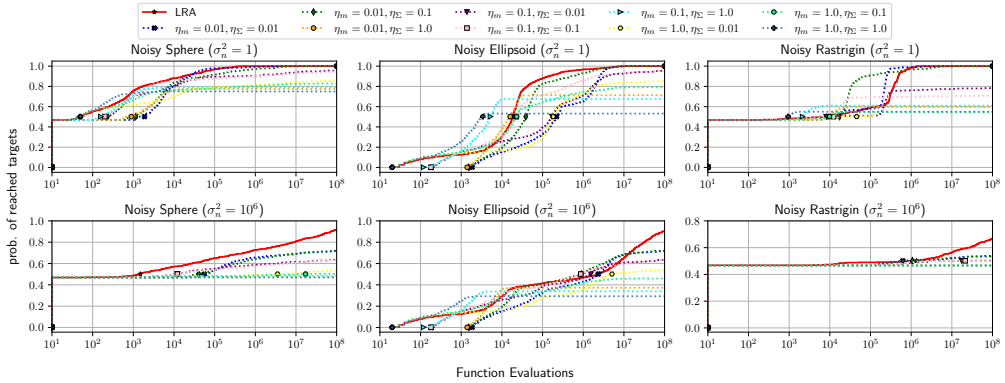


Figure 4.7: Empirical cumulative density function for 10-D noisy problems, with σ_n^2 set to 1 or 10^6.

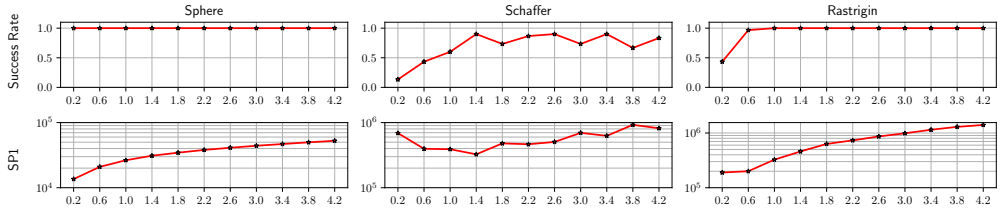


Figure 4.8: Success rates and SP1 values with hyperparameter α for 30-D noiseless problems (30 trials).

4.4.5 Effects of Population Size

Although we used the *default* population size, $\lambda = 4 + \lfloor 3 \log(d) \rfloor$, in all the experiments, practitioners may want to employ different population sizes to fully utilize their parallel environments. In this section, we describe the experiments conducted to investigate the effects of population size.

Figure 4.10 shows the success rates and SP1 values with respect to the populatioon size $\lambda \in \{14, 28, 42, 56, 70\}$ for the 30-D noiseless Sphere, Schaffer, and Rastrigin functions. Although the SP1 value worsens with a larger λ for the Rastrigin function, it appears to have a mild impact for the Sphere and Schaffer functions. Figure 4.11 shows typical behaviors of LRA-CMA-ES with $\lambda \in \{14, 42, 70\}$ on the 30-D Sphere function. As λ increases, it can be observed that the learning rates (especially η_m) also generally increase linearly. This is because, as λ increases, the SNR also increases, allowing for a larger learning rate to maintain the target SNR. This phenomenon can also be theoretically explained as follows:

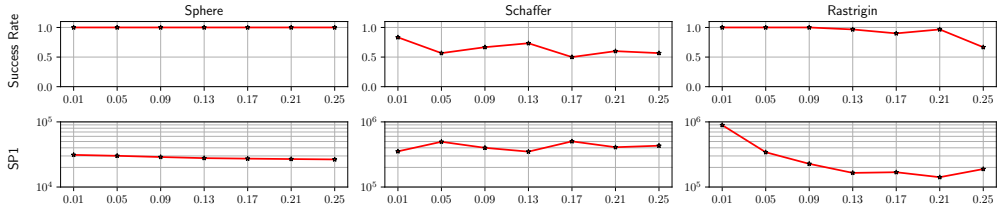


Figure 4.9: Success rates and SP1 values with hyperparameter β_Σ for 30-D noiseless problems (30 trials).

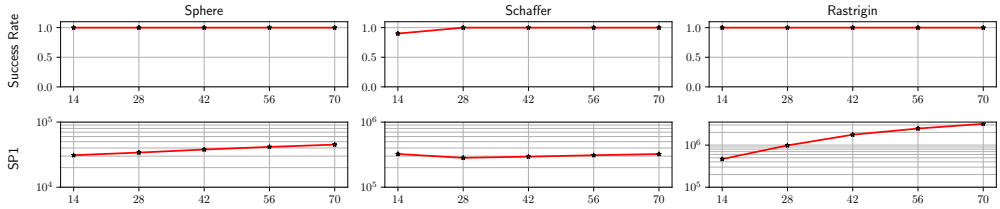


Figure 4.10: Success rates and SP1 values with $\lambda \in \{14, 28, 42, 56, 70\}$ for 30-D noiseless problems (30 trials).

The SNR analysis for the infinite-dimensional Sphere function in Appendix B.3 shows that under the assumption of the optimal step-size, SNR $\approx O(\lambda/d)$. In this case, increasing λ can linearly increase the SNR; therefore, it is expected that the learning rate can be kept linearly larger, which is consistent with our empirical findings. However, this analysis was conducted using the (infinite-dimensional) Sphere function; thus, this discussion cannot be directly applied to multimodal problems.

Additionally, we investigated the behavior for larger population sizes using various values of $\lambda \in \{500, 1000, 1500, 2000, 2500\}$, as shown in Figure 4.12. Compared to the results for $\lambda \in \{14, 28, 42, 56, 70\}$, the SP1 value remains almost constant for the Rastrigin function with respect to the λ value. However, in the Sphere and Schaffer functions, the SP1 value deteriorates slightly for larger λ values. This may be partially because the proposed method is designed to solve difficult problems (e.g., $\eta_m, \eta_\Sigma \leqslant 1$). Although more aggressive learning rate updates may improve the performance, such strategies were beyond the scope of this study.

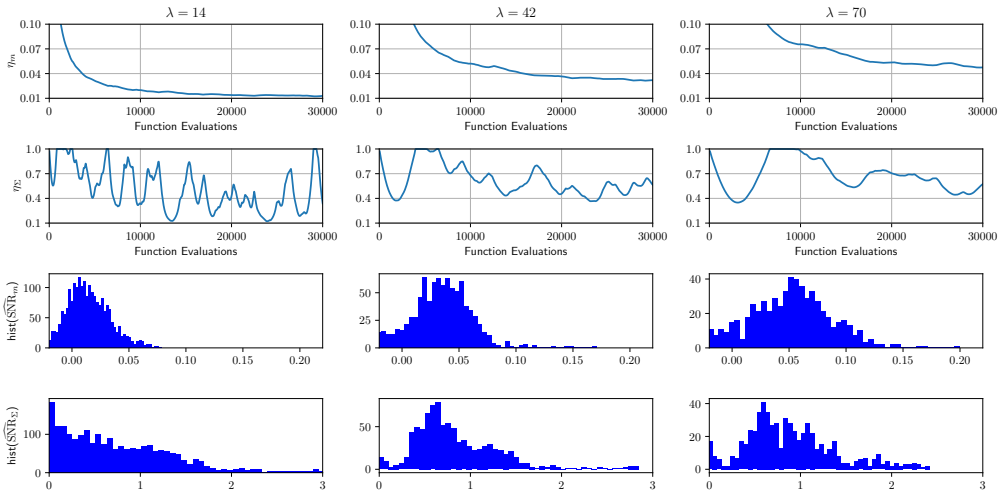


Figure 4.11: LRA-CMA-ES behaviors on the 30-D Sphere function with $\lambda \in \{14, 42, 70\}$. η_m, η_Σ, and the histograms of the estimated SNR w.r.t. m and Σ, in this order from the top. The y-axes in η_m and η_Σ are shown on the linear scale rather than the log scale.

4.4.6 LRA-CMA-ES vs. PSA-CMA-ES

We compared the performance of the proposed LRA-CMA-ES with that of PSA-CMA-ES [64], which is a state-of-the-art population size adaptation method. For a fair comparison, we employed almost the same procedure and hyperparameters for PSA-CMA-ES as those for the CMA-ES described in Section 2.3. The only difference was that PSA-CMA-ES required additional normalization factors (Eqs. (6) and (7) in [64]) to derive an approximate value for the parameter movement. For the step-size correction in PSA-CMA-ES, we used Blom's approximation to cal-

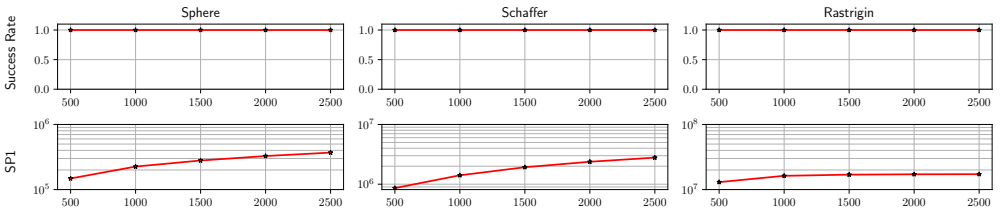


Figure 4.12: Success rates and SP1 values with $\lambda \in \{500, 1000, 1500, 2000, 2500\}$ for 30-D noiseless problems (30 trials).

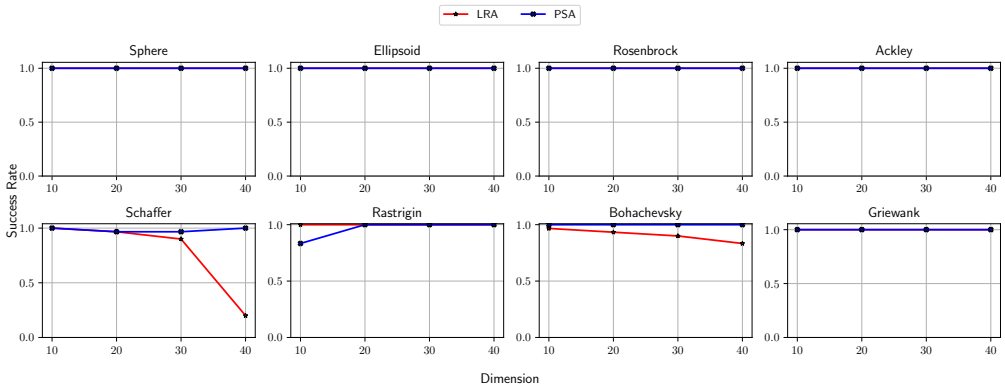


Figure 4.13: Performances of LRA-CMA-ES and PSA-CMA-ES: success rates according to the number of dimensions (noiseless problems).

culate the weighted average of the expected value of normal-order statistics [3]. Additionally, we used the recommended values for the PSA-CMA-ES hyperparameters [64]. The experimental settings were the same as those described in Section 4.4.3. All LRA-CMA-ES results were obtained from Section 4.4.3.

Figures 4.13 and 4.14 show the success rates and SP1 values, respectively, for the noiseless problems, wherein it is evident that PSA-CMA-ES exhibits better results than LRA-CMA-ES for most problems. Figure 4.15 illustrates the ECDF for noisy problems. The performance of LRA-CMA-ES is better than that of PSA-CMA-ES in most of the tested cases. For example, for the Rastrigin function with a strong-noise setting (bottom right of Figure 4.15), PSA-CMA-ES stopped improving the function value, whereas LRA-CMA-ES continued improving it. These results suggest that LRA-CMA-ES and PSA-CMA-ES are suitable for different problems. However, these performance differences can be mitigated to a certain degree by adjusting the hyperparameters of each method and do not necessarily suggest that there is a fundamental performance difference between learning rate and population size adaptations. Although we still argue that LRA is more practically useful than population size adaptation, as described in Section 4.1, a detailed comparison of these methods will be an interesting direction for future work.

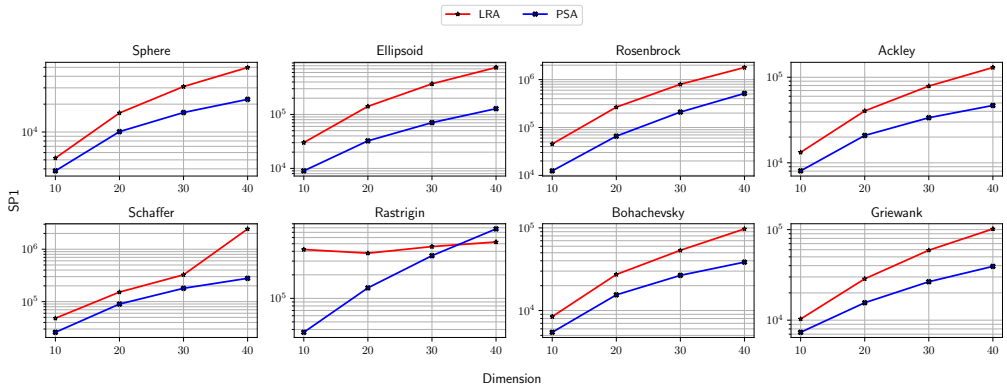


Figure 4.14: Performaces of LRA-CMA-ES and PSA-CMA-ES: SP1 values according to the number of dimensions (noiseless problems).

4.5 Conclusion

This study presented the design principles and practices of LRA for CMA-ES. We first demonstrated that difficult problems can be solved relatively easily by decreasing the learning rate and ensuring that the parameter behavior was closer to the ODE trajectory. However, decreasing it excessively worsened the search efficiency. Therefore, we attempted to determine the optimal learning rate for maximizing the expected improvement, which was nearly proportional to the SNR under some assumptions. Based on these observations, we developed a new LRA mechanism to solve multimodal and noisy problems using CMA-ES without extremely expensive learning-rate tuning. The basic concept of the proposed algorithm, LRA-CMA-ES, is to adapt the learning rate such that the SNR can be kept constant, which is nearly optimal based on the optimal learning rate discussion. Experiments involving noiseless multimodal problems revealed that the proposed LRA-CMA-ES can adapt the learning rate appropriately depending on the search situation, and it works well without tuning the learning rate. Additionally, LRA-CMA-ES provided better solutions for noisy problems, even under strong-noise settings, which yielded problems that could not be solved by CMA-ES with a fixed learning rate. In conclusion, LRA-CMA-ES effectively facilitates the solving of multimodal and noisy problems to a certain extent, eliminating the need for tuning the learning rate.

However, the proposed LRA-CMA-ES has some limitations, which will be addressed in future research. First, it experienced several failures for the 40-D Schaffer function, although CMA-ES with an appropriately small learning rate

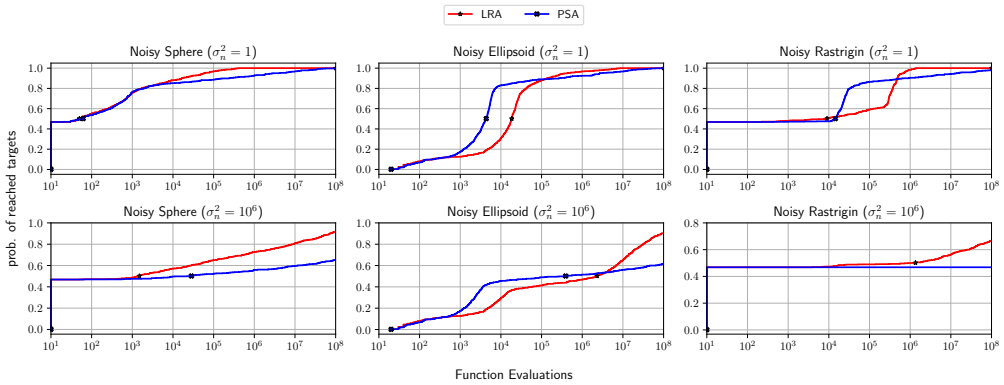


Figure 4.15: Performances of LRA-CMA-ES and PSA-CMA-ES: Empirical cumulative density function for 10-D noisy problems, with σ_n^2 set to 1 or 10^6.

succeeded with a high probability. We believe that a detailed analysis of the SNR adaptation behavior is crucial to determine the reasons for this failure. On a related note, our understanding of the appropriate hyperparameter settings in the proposed LRA mechanism remains limited. Our experiments revealed that the hyperparameter settings affect the trade-off between stability and convergence speed. Through experiment, we identified the hyperparameters that perform relatively well for noiseless and noisy problems; however, better configuration methods must be developed. For example, the constant value $O(1)$ is used for the cumulation factors β_m and β_Σ; however, it may be more reasonable to consider that these factors depend on the parameter degrees of freedom, that is, $\beta_m = O(1/d)$ and $\beta_\Sigma = O(1/d^2)$. In addition, the method for determining the appropriate value of the target SNR α can be refined further. The SNR analysis presented in Appendix B.3 implies that $\alpha = O(\lambda/d)$ is reasonable for an infinite-dimensional Sphere function. On the other hand, because this analysis cannot be directly applied to multimodal or noisy problems, there is still room to discuss the best method for determining α. A deeper understanding of the hyperparameter effects is crucial for improving the reliability of the proposed LRA-CMA-ES.

Additionally, LRA-CMA-ES, which mainly focuses on well-structured multimodal problems, alone cannot solve weakly structured ones, as discussed in Section 4.1. To address these situations, integrating restart strategies with LRA-CMA-ES is a promising direction.

Finally, developing a more rational LRA approach is an intriguing topic for future research. Although our discussion in Section 4.2.3 offers valuable insights into designing an ideal learning rate, it was based on several assumptions and

has the potential for improvement. A more detailed theoretical study could result in a more rational design for learning rates, which is crucial for advancing this line of research.

Chapter 5

Conclusion

5.1 Summary of Contributions

Throughout this doctoral dissertation, we pursued learning rate adaptation for evolution strategies (ES). Despite the practical importance of the learning rate, there are few studies to automatically adapt it and thus this had remained significant obstacle to be addessed. To deal with this issue, we conducted two learning rate adaptation works, employing exponential natural evolution strategies (xNES) and covariance matrix adaptation evolution strategy (CMA-ES) as promising ES variants.

The first major contribution of this dissertation is the introduction of the learning rate adaptation method *for acceleration*. In Chapter 3, we utilized xNES as the underlying optimization algorithm. Our investigation began with analyzing the impact of stochastic relaxation, which provided critical insights into why the learning rate adaptation is essential for enhancing xNES. Subsequently, we proposed a novel learning rate adaptation method tailored for xNES. The main concept of our approach is to dynamically adapt the learning rate by increasing it when sufficient tendencies are detected in the updates of the distribution parameters. To quantify these tendencies, we introduced an evolution path in the distribution parameter space, inspired by its use in population size adaptation within CMA-ES [62, 64]. The experimental results on unimodal and multimodal problems demonstrated that the proposed method works properly depending on a search situation and is effective over the existing method, i.e., using the fixed learning rate.

The second major contribution of this dissertation is the introduction of the

learning rate adaptation method *for solving difficult problems such as multimodal and noisy ones*. In Chapter 4, we utilized CMA-ES as the underlying optimization algorithm. This study first comprehensively explored the impact of learning rate on CMA-ES performance and demonstrated the necessity of a *small* learning rate by considering ordinary differential equations. Thereafter, it discussed the setting of an ideal learning rate. Based on these discussions, we developed a novel learning rate adaptation mechanism for CMA-ES that maintains a constant signal-to-noise ratio (SNR). Additionally, we investigated the behavior of CMA-ES with the proposed learning rate adaptation mechanism through numerical experiments and compare the results with those obtained for CMA-ES with a fixed learning rate and with population size adaptation. The results show that CMA-ES with the proposed learning rate adaptation works well for multimodal and/or noisy problems *without* extremely expensive learning rate tuning.

Although the learning rate plays a critical role in the performance of evolution strategies, research on learning rate adaptation methodologies has remained relatively unexplored. We hope that this study marks a significant step forward in advancing practical and impactful research in this promising area.

5.2 Relationship Between First and Second Works

One might wonder whether the methods proposed in the first and second works can be combined. Unfortunately, such integration appears challenging. For instance, the main concept of the method introduced in the second work (Chapter 4) is maintaining a constant SNR by dynamically adapting the learning rate. Incorporating a different learning rate adaptation mechanism, such as the one proposed in the first work, would introduce conflicting mechanisms, thereby undermining the core concept of the second method. However, extending the method from the second work to achieve acceleration—the focus of the first work—appears more feasible. Technically, this can be achieved by modifying the LRA-CMA-ES to remove the upper bound of one in Eq. (4.23). While this extension is conceptually straightforward, it remains untested, and further refinements, such as hyperparameter tuning, may be necessary to achieve satisfactory performance. Exploring this direction represents an important step toward developing a truly general-purpose learning rate adaptation method.

5.3 Future Work

This work established a solid foundation and introduced practical mechanisms for learning rate adaptation. However, several challenges and open questions remain for future exploration. In this section, we highlight key directions for further research.

5.3.1 Beyond Continuous Optimization

While this study focused on black-box optimization in continuous domains, the proposed methods can be extended to other domains. For instance, the learning rate adaptation approach introduced in Chapter 3 can be integrated into the general IGO framework, provided the expectation of the KL divergence between updates can be computed or reasonably approximated.

Moreover, the approach introduced in Chapter 4 is highly general and can be extended to a broad range of optimizers. Notably, while it was applied to the CMA-ES in Chapter 4, the method does not rely on any specific implementation details of CMA-ES for learning rate adaptation. This inherent flexibility makes it well-suited for integration into more complex and sophisticated optimization algorithms, presenting an exciting direction for future research.

Another avenue worth exploring is the integration of the learning rate adaptation approach into the CMA-ES with Margin [29, 30], which applies CMA-ES to mixed-integer black-box optimization problems. Notably, the CMA-ES with Margin has yet to be tested on multimodal or noisy problems, highlighting its extension to these challenging scenarios via the learning rate adaptation as a promising direction for future research. Exploring the application of our approach to CMA-ES-SoP [91], which extends beyond integers to handle generalized discrete spaces, represents another compelling direction for the same reason.

5.3.2 Beyond Well-Structured Multimodal Problems

This study focused on well-structured multimodal problems and did not address weakly structured ones. However, in black-box scenarios, such problem structures cannot be assumed in advance. Therefore, it is desirable to extend our method to handle general optimization problems, including weakly structured multimodal ones.

To address this issue, we plan to enhance our learning rate adaptation method by incorporating restart strategies (e.g., [32]) and learning rate scheduling. These

additions are expected to improve the method's reliability and robustness. Notably, the PSA-CMA-ES with a restart strategy [63] has already demonstrated successful performance, further motivating this approach.

5.3.3 Beyond Additive Noise

While this study focused on additive noise, exploring other types of noise offers an intriguing direction for future research. Recently, Uchida et al.[92] proposed a reevaluation-based approach for efficiently managing multiplicative noise. Their method integrates our learning rate adaptation approach[65], demonstrating its effectiveness in improving performance on multimodal problems. Building on this foundation, developing new methods that leverage our approach as a core component represents a promising avenue for further exploration.

5.3.4 Beyond Synthetic Benchmark Problems

In this study, we employed benchmark problems to gain a precise understanding of the problem structure, enabling a detailed and thorough discussion of our approach. Moving forward, it is crucial to validate the practical effectiveness of our method in real-world optimization problems.

Encouragingly, following the publication of our conference paper [65], several successful applications of our approach have already emerged. For instance, Zhang and Chen [100] applied our learning rate adaptation methodology to rigid 2D/3D registration for robotic navigation in spine surgery. Their results show that LRA-CMA-ES achieves more than twice the speed of standard CMA-ES. Similarly, Green and Lundgren [27] utilized our learning rate adaptation approach to enhance gravitational-wave search efforts. They reported that the combination of CMA-ES with increasing population restart and our learning rate adaptation approach delivered the best performance. We believe that applying the method to such real-world problems is essential for uncovering significant but unresolved practical challenges in the learning rate adaptation approach.

Bibliography

[1] Y. Akimoto. Analysis of Surrogate-Assisted Information-Geometric Optimization Algorithms. *Algorithmica*, 86(1):33–63, 2024.

[2] Y. Akimoto, A. Auger, and N. Hansen. Comparison-Based Natural Gradient Optimization in High Dimension. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 373–380. ACM, 2014.

[3] Y. Akimoto, A. Auger, and N. Hansen. Quality Gain Analysis of the Weighted Recombination Evolution Strategy on General Convex Quadratic Functions. *Theoretical Computer Science*, 832:42–67, 2020.

[4] Y. Akimoto, A. Auger, and N. Hansen. An ODE Method to Prove the Geometric Convergence of Adaptive Stochastic Algorithms. *Stochastic Processes and their Applications*, 145:269–307, 2022.

[5] Y. Akimoto and N. Hansen. Diagonal Acceleration for Covariance Matrix Adaptation Evolution Strategies. *Evolutionary Computation*, 28(3):405–435, 2020.

[6] Y. Akimoto, Y. Nagata, I. Ono, and S. Kobayashi. Bidirectional Relation between CMA Evolution Strategies and Natural Evolution Strategies. In *International Conference on Parallel Problem Solving from Nature*, pages 154–163, 2010.

[7] Y. Akimoto, Y. Nagata, I. Ono, and S. Kobayashi. Theoretical Foundation for CMA-ES from Information Geometry Perspective. *Algorithmica*, 64:698–716, 2012.

[8] Y. Akimoto and Y. Ollivier. Objective Improvement in Information-Geometric Optimization. In *Proceedings of the twelfth workshop on Foundations of genetic algorithms XII*, pages 1–10, 2013.

[9] S.-i. Amari and S. C. Douglas. Why Natural Gradient? In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 1213–1216, 1998.

[10] S.-i. Amari and H. Nagaoka. *Methods of Information Geometry*, volume 191. 2000.

[11] D. V. Arnold. Optimal Weighted Recombination. In *Foundations of Genetic Algorithms*, pages 215–237, 2005.

[12] A. Auger and N. Hansen. A Restart CMA Evolution Strategy with Increasing Population Size. In *IEEE Congress on Evolutionary Computation*, volume 2, pages 1769–1776. IEEE, 2005.

[13] A. Auger and N. Hansen. Performance Evaluation of an Advanced Local Search Evolutionary Algorithm. In *2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 1777–1784. IEEE, 2005.

[14] W. J. BANGS II. *Array Processing with Generalized Beam-Formers*. Yale University, 1971.

[15] S. A. Basilio and L. Goliatt. Gradient Boosting Hybridized with Exponential Natural Evolution Strategies for Estimating the Strength of Geopolymer Self-Compacting Concrete. *Knowledge-Based Engineering and Sciences*, 3(1):1–16, 2022.

[16] S. d. C. A. Basílio, C. M. Saporetti, and L. Goliatt. An Interdependent Evolutionary Machine Learning Model Applied to Global Horizontal Irradiance Modeling. *Neural Computing and Applications*, 35(16):12099–12120, 2023.

[17] E. A. Bedolla-Montiel, J. T. Lange, A. Pérez de Alba Ortíz, and M. Dijkstra. Inverse Design of Crystals and Quasicrystals in a Non-Additive Binary Mixture of Hard Disks. *The Journal of Chemical Physics*, 160(24), 2024.

[18] H.-G. Beyer. Convergence Analysis of Evolutionary Algorithms That Are Based on the Paradigm of Information Geometry. *Evolutionary Computation*, 22(4):679–709, 2014.

[19] H.-G. Beyer and H.-P. Schwefel. Evolution Strategies–a Comprehensive Introduction. *Natural computing*, 1:3–52, 2002.

[20] G. Cuccu, M. Luciw, J. Schmidhuber, and F. Gomez. Intrinsically Motivated Neuroevolution for Vision-based Reinforcement Learning. In *2011 IEEE International Conference on Development and Learning (ICDL)*, volume 2, pages 1–7. IEEE, 2011.

[21] G. Cuccu, J. Togelius, and P. Cudré-Mauroux. Playing Atari with Few Neurons: Improving the Efficacy of Reinforcement Learning by Decoupling Feature Extraction and Decision Making. *Autonomous Agents and Multi-Agent Systems*, 35(2):17, 2021.

[22] Z. Fan, Z. Zeng, C. Zhang, Y. Wang, K. Song, H. Dong, Y. Chen, and T. Ala-Nissila. Neuroevolution Machine Learning Potentials: Combining High Accuracy and Low Cost in Atomistic Simulations and Application to Heat Transport. *Physical Review B*, 104(10):104309, 2021.

[23] G. Fujii, Y. Akimoto, and M. Takahashi. Exploring optimal topology of thermal cloaks by CMA-ES. *Applied Physics Letters*, 112(6), 2018.

[24] N. Fukushima, Y. Nagata, S. Kobayashi, and I. Ono. Proposal of distance-weighted exponential natural evolution strategies. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 164–171. IEEE, 2011.

[25] A. Gissler, A. Auger, and N. Hansen. Learning Rate Adaptation by Line Search in Evolution Strategies with Recombination. In *Proceedings of the Genetic and Evolutionary Computation Conference*, page 630–638, 2022.

[26] T. Glasmachers, T. Schaul, S. Yi, D. Wierstra, and J. Schmidhuber. Exponential Natural Evolution Strategies. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 393–400, 2010.

[27] S. Green and A. Lundgren. GWtuna: Trawling through the data to find Gravitational Waves with Optuna and Jax. *arXiv preprint arXiv:2411.03207*, 2024.

[28] D. Ha and J. Schmidhuber. World Models. *arXiv preprint arXiv:1803.10122*, 2018.

[29] R. Hamano, S. Saito, M. Nomura, and S. Shirakawa. CMA-ES with Margin: Lower-Bounding Marginal Probability for Mixed-Integer Black-Box Optimization. In *Proceedings of the genetic and evolutionary computation conference*, pages 639–647, 2022.

[30] R. Hamano, S. Saito, M. Nomura, and S. Shirakawa. Marginal Probability-Based Integer Handling for CMA-ES Tackling Single-and Multi-Objective Mixed-Integer Black-Box Optimization. *ACM Transactions on Evolutionary Learning*, 2024.

[31] R. Hamano, S. Shirakawa, and M. Nomura. Natural Gradient Interpretation of Rank-One Update in CMA-ES. In *International Conference on Parallel Problem Solving from Nature*, pages 252–267. Springer, 2024.

[32] N. Hansen. Benchmarking a BI-Population CMA-ES on the BBOB-2009 Function Testbed. In *Proceedings of the Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pages 2389–2396, 2009.

[33] N. Hansen. The CMA Evolution Strategy: A Tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

[34] N. Hansen, D. V. Arnold, and A. Auger. Evolution Strategies. *Springer handbook of computational intelligence*, pages 871–898, 2015.

[35] N. Hansen and A. Auger. Principled Design of Continuous Stochastic Search: From Theory to Practice. In *Theory and Principled Methods for the Design of Metaheuristics*, pages 145–180. Springer, 2014.

[36] N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, and D. Brockhoff. COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting. *Optimization Methods and Software*, 36(1):114–144, 2021.

[37] N. Hansen and S. Kern. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In *International Conference on Parallel Problem Solving from Nature*, pages 282–291, 2004.

[38] N. Hansen, S. D. Müller, and P. Koumoutsakos. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary computation*, 11(1):1–18, 2003.

[39] N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[40] K. Hasselmann, A. Ligot, J. Ruddick, and M. Birattari. Empirical Assessment and Comparison of Neuro-Evolutionary Methods for the Automatic Off-Line Design of Robot Swarms. *Nature communications*, 12(1):4345, 2021.

[41] M. Hellwig and H.-G. Beyer. Evolution Under Strong Noise: A Self-Adaptive Evolution Strategy Can Reach the Lower Performance Bound - The pcCMSA-ES. In *International Conference on Parallel Problem Solving from Nature*, pages 26–36, 2016.

[42] Y. Hu, G. Chen, Z. Li, and A. Knoll. Robot Policy Improvement with Natural Evolution Strategies for Stable Nonlinear Dynamical System. *IEEE Transactions on Cybernetics*, 53(6):4002–4014, 2022.

[43] B. Huang, C. Lu, L. Leqi, J. M. Hernández-Lobato, C. Glymour, B. Schölkopf, and K. Zhang. Action-Sufficient State Representation Learning for Control with Structural Constraints. In *International Conference on Machine Learning*, pages 9260–9279, 2022.

[44] H. Ishida, N. Hiraoka, K. Okada, and M. Inaba. CoverLib: Classifiers-equipped Experience Library by Iterative Problem Distribution Coverage Maximization for Domain-tuned Motion Planning. *arXiv preprint arXiv:2405.02968*, 2024.

[45] M. Ishige, Y. Yoshimura, and R. Yonetani. Opt-in Camera: Person Identification in Video via UWB Localization and Its Application to Opt-in Systems. *arXiv preprint arXiv:2409.19891*, 2024.

[46] S. Jastrzębski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey. Three Factors Influencing Minima in SGD. *arXiv preprint arXiv:1711.04623*, 2017.

[47] M. Kegeleirs, D. G. Ramos, K. Hasselmann, L. Garattoni, G. Francesca, and M. Birattari. Transferability in the Automatic Off-Line Design of Robot Swarms: from Sim-to-Real to Embodiment and Design-Method Transfer across Different Platforms. *IEEE Robotics and Automation Letters*, 2024.

[48] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *arXiv preprint arXiv:1609.04836*, 2016.

[49] K. Kikuchi, M. Otani, K. Yamaguchi, and E. Simo-Serra. Modeling Visual Containment for Web Page Layout Optimization. In *Computer Graphics Forum*, volume 40, pages 33–44, 2021.

[50] K. Kikuchi, E. Simo-Serra, M. Otani, and K. Yamaguchi. Constrained Graphic Layout Generation via Latent Optimization. In *Proceedings of the ACM International Conference on Multimedia*, pages 88–96, 2021.

[51] K. Kimura and I. Ono. A Reinforcement Learning Method Based on Natural Evolution Strategies. In *2024 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2024.

[52] P. E. Kloeden and E. Platen. *Stochastic Differential Equations*. 1992.

[53] Y. Kobayashi and I. Ono. Sequential Estimation of States and Parameters of Nonlinear State Space Models using Particle Filter and Natural Evolution Strategy. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2020.

[54] O. Krause. Large-Scale Noise-Resilient Evolution-Strategies. In *Proceedings of the Genetic and Evolutionary Computation Conference*, page 682–690, 2019.

[55] I. Loshchilov. CMA-ES with Restarts for Solving CEC 2013 Benchmark Problems. In *2013 IEEE Congress on Evolutionary Computation*, pages 369–376. Ieee, 2013.

[56] I. Loshchilov and F. Hutter. CMA-ES for Hyperparameter Optimization of Deep Neural Networks. *arXiv preprint arXiv:1604.07269*, 2016.

[57] I. Loshchilov, M. Schoenauer, M. Sebag, and N. Hansen. Maximum Likelihood-based Online Adaptation of Hyper-parameters in CMA-ES. In *International Conference on Parallel Problem Solving from Nature*, pages 70–79, 2014.

[58] A. Maki, N. Sakamoto, Y. Akimoto, H. Nishikawa, and N. Umeda. Application of optimal control theory based on the evolution strategy (CMA-ES) to

automatic berthing. *Journal of Marine Science and Technology*, 25:221–233, 2020.

[59] H. Miyazawa and Y. Akimoto. Effect of the Mean Vector Learning Rate in CMA-ES. In *Proceedings of the Genetic and Evolutionary Computation Conference*, page 721–728, 2017.

[60] N. Müller and T. Glasmachers. Non-Local Optimization: Imposing Structure on Optimization Problems by Relaxation. In *Proceedings of the 16th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*, pages 1–10, 2021.

[61] D. M. Nguyen and N. Hansen. Benchmarking CMAES-APOP on the BBOB Noiseless Testbed. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, page 1756–1763, 2017.

[62] K. Nishida and Y. Akimoto. 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*, page 237–244, 2016.

[63] K. Nishida and Y. Akimoto. Benchmarking the PSA-CMA-ES on the BBOB Noiseless Testbed. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1529–1536, 2018.

[64] K. Nishida and Y. Akimoto. PSA-CMA-ES: CMA-ES with Population Size Adaptation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, page 865–872, 2018.

[65] M. Nomura, Y. Akimoto, and I. 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*, page 839–847, 2023.

[66] M. Nomura, Y. Akimoto, and I. Ono. CMA-ES with Learning Rate Adaptation. *ACM Transactions on Evolutionary Learning*, 2024.

[67] M. Nomura and I. Ono. Natural Evolution Strategy for Unconstrained and Implicitly Constrained Problems with Ridge Structure. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7. IEEE, 2021.

[68] M. Nomura and I. Ono. Fast Moving Natural Evolution Strategy for High-Dimensional Problems. In *2022 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2022.

[69] M. Nomura and I. Ono. Towards a Principled Learning Rate Adaptation for Natural Evolution Strategies. In *Applications of Evolutionary Computation*, pages 721–737, 2022.

[70] M. Nomura, N. Sakai, N. Fukushima, and I. Ono. Distance-weighted Exponential Natural Evolution Strategy for Implicitly Constrained Black-Box Function Optimization. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 1099–1106. IEEE, 2021.

[71] M. Nomura and M. Shibata. cmaes : A Simple yet Practical Python Library for CMA-ES. *arXiv preprint arXiv:2402.01373*, 2024.

[72] M. Nomura, S. Watanabe, Y. Akimoto, Y. Ozaki, and M. Onishi. Warm Starting CMA-ES for Hyperparameter Optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9188–9196, 2021.

[73] Y. Ollivier, L. Arnold, A. Auger, and N. Hansen. Information-Geometric Optimization Algorithms: A Unifying Picture via Invariance Principles. *Journal of Machine Learning Research*, 18(18):1–65, 2017.

[74] A. Piergiovanni, A. Angelova, and M. S. Ryoo. Evolving Losses for Unsupervised Video Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 133–142, 2020.

[75] L. O. Purucker and J. Beel. CMA-ES for Post Hoc Ensembling in AutoML: A Great Success and Salvageable Failure. In *AutoML Conference 2023*, 2023.

[76] I. Rechenberg. Evolutionsstrategie. *Frommann-Holzboog Verlag*, 1994.

[77] T. Schaul. *Studies in Continuous Black-Box Optimization*. PhD thesis, Technische Universität München, 2011.

[78] T. Schaul. Benchmarking Exponential Natural Evolution Strategies on the Noiseless and Noisy Black-Box Optimization Testbeds. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, pages 213–220, 2012.

[79] T. Schaul. Investigating the Impact of Adaptation Sampling in Natural Evolution Strategies on Black-Box Optimization Testbeds. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, pages 221–228, 2012.

[80] T. Schaul. Natural Evolution Strategies Converge on Sphere Functions. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pages 329–336, 2012.

[81] T. Schaul, T. Glasmachers, and J. Schmidhuber. High Dimensions and Heavy Tails for Natural Evolution Strategies. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 845–852, 2011.

[82] L. Schönenberger and H.-G. Beyer. On a Population Sizing Model for Evolution Strategies Optimizing the Highly Multimodal Rastrigin Function. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 848–855, 2023.

[83] H.-P. P. Schwefel. *Evolution and Optimum Seeking*. John Wiley & Sons, Inc., 1993.

[84] S. Shirakawa, Y. Akimoto, K. Ouchi, and K. Ohara. Sample Reuse via Importance Sampling in Information Geometric Optimization. *arXiv preprint arXiv:1805.12388*, 2018.

[85] D. Slepian. Estimation of signal parameters in the presence of noise. *Transactions of the IRE Professional Group on Information Theory*, 3(3):68–89, 1954.

[86] M. Stollenga, L. Pape, M. Frank, J. Leitner, A. Förster, and J. Schmidhuber. Task-Relevant Roadmaps: A Framework for Humanoid Motion Planning. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5772–5778. IEEE, 2013.

[87] M. F. Stollenga, J. Masci, F. Gomez, and J. Schmidhuber. Deep Networks with Internal Selective Attention through Feedback Connections. *Advances in neural information processing systems*, 27, 2014.

[88] Y. Sun, D. Wierstra, T. Schaul, and J. Schmidhuber. Efficient Natural Evolution Strategies. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 539–546. ACM, 2009.

[89] T. Tanabe, K. Fukuchi, J. Sakuma, and Y. Akimoto. Level Generation for Angry Birds with Sequential VAE and Latent Variable Evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1052–1060, 2021.

[90] S. Tian, Y. Cai, H.-X. Yu, S. Zakharov, K. Liu, A. Gaidon, Y. Li, and J. Wu. Multi-Object Manipulation via Object-Centric Neural Scattering Functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9021–9031, 2023.

[91] K. Uchida, R. Hamano, M. Nomura, S. Saito, and S. Shirakawa. CMA-ES for Discrete and Mixed-Variable Optimization on Sets of Points. In *International Conference on Parallel Problem Solving from Nature*, pages 236–251. Springer, 2024.

[92] K. Uchida, K. Nishihara, and S. Shirakawa. CMA-ES with Adaptive Reevaluation for Multiplicative Noise. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 731–739, 2024.

[93] K. Uchida, S. Shirakawa, and Y. Akimoto. Finite-Sample Analysis of Information Geometric Optimization with Isotropic Gaussian Distribution on Convex Quadratic Functions. *IEEE Transactions on Evolutionary Computation*, 24(6):1035–1049, 2019.

[94] V. Volz, J. Schrum, J. Liu, S. M. Lucas, A. Smith, and S. Risi. Evolving Mario Levels in the Latent Space of a Deep Convolutional Generative Adversarial Network. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 221–228, 2018.

[95] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber. Natural Evolution Strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

[96] J. C. Wong, A. Gupta, and Y.-S. Ong. Can Transfer Neuroevolution Tractably Solve Your Differential Equations? *IEEE Computational Intelligence Magazine*, 16(2):14–30, 2021.

[97] T. Yamaguchi and Y. Akimoto. Benchmarking the Novel CMA-ES Restart Strategy Using the Search History on the BBOB Noiseless Testbed. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1780–1787, 2017.

[98] D. Yamamoto, H. Yoshida, Y. Kobayashi, and I. Ono. Sequential Estimation of States and Parameters of Non-Linear State Space Models Taking Account of Ensembles Not Covering True States. In *2023 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2023.

[99] S. Yi, D. Wierstra, T. Schaul, and J. Schmidhuber. Stochastic Search Using the Natural Gradient. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1161–1168. ACM, 2009.

[100] Z. Zhang and M. Chen. Introducing Learning Rate Adaptation CMA-ES into Rigid 2D/3D Registration for Robotic Navigation in Spine Surgery. *arXiv preprint arXiv:2405.10186*, 2024.

List of Publications

Publications Related to Doctoral Dissertation

Peer-Reviewed Journal Papers

1. Masahiro Nomura, Youhei Akimoto, Isao Ono : CMA-ES with Learning Rate Adaptation, ACM Transactions on Evolutionary Learning, 2024.

Peer-Reviewed International Conference Papers

2. Masahiro Nomura, Isao Ono : Towards a Principled Learning Rate Adaptation for Natural Evolution Strategies, International Conference on the Applications of Evolutionary Computation (Part of EvoStar), pp.721-737, 2022.

3. Masahiro Nomura, Youhei Akimoto, Isao Ono : CMA-ES with Learning Rate Adaptation: Can CMA-ES with Default Population Size Solve Multimodal and Noisy Problems?, Proceedings of the Genetic and Evolutionary Computation Conference, pp.839-847, 2023. (Best Paper Nomination at ENUM Track)

Other Publications

Peer-Reviewed Journal Papers

4. Ryoki Hamano, Shota Saito, Masahiro Nomura, Shinichi Shirakawa : Marginal Probability-Based Integer Handling for CMA-ES Tackling Single-and Multi-Objective Mixed-Integer Black-Box Optimization, ACM Transactions on Evolutionary Learning, 10, pp.1-26, 2024.

5. Yoshihiko Ozaki, Yuki Tanigaki, Shuhei Watanabe, Masahiro Nomura, Masaki Onishi : Multiobjective Tree-Structured Parzen Estimator, Journal of Artificial Intelligence Research, 73, pp.1209-1250, 2022.

6. Yoshihiko Ozaki, Masahiro Nomura, Masaki Onishi : Hyperparameter Optimization Methods: Overview and Characteristics, IEICE TRANS. INF. & SYST., Vol.J103-D No.9 SEPTEMBER 2020 (Best Paper Award), in Japanese.

Peer-Reviewed International Conference Papers

7. Tatsuhiro Shimizu*, Koichi Tanaka*, Ren Kishimoto, Haruka Kiyohara, Masahiro Nomura, Yuta Saito : Effective Off-Policy Evaluation and Learning in Contextual Combinatorial Bandits, In Proceedings of the ACM Conference on Recommender Systems, pp.733-741, 2024.

8. Ryoki Hamano, Shinichi Shirakawa, Masahiro Nomura : Natural Gradient Interpretation of Rank-One Update in CMA-ES, In International Conference on Parallel Problem Solving from Nature, pp.252-267 2024.

9. Kento Uchida, Ryoki Hamano, Masahiro Nomura, Shota Saito, Shinichi Shirakawa : CMA-ES for Discrete and Mixed-Variable Optimization on Sets of Points, In International Conference on Parallel Problem Solving from Nature, pp.236-251, 2024.

10. Yuta Saito, Masahiro Nomura : Hyperparameter Optimization Can Even be Harmful in Off-Policy Learning and How to Deal with It, In Proceedings of the International Joint Conference on Artificial Intelligence, pp.4860-4867, 2024.

11. Kento Uchida, Ryoki Hamano, Masahiro Nomura, Shota Saito, Shinichi Shirakawa : CMA-ES for Safe Optimization, In Proceedings of the Genetic and Evolutionary Computation Conference, pp.722-730, 2024.

12. Ryoki Hamano, Shota Saito, Masahiro Nomura, Kento Uchida, Shinichi Shirakawa : CatCMA: Stochastic Optimization for Mixed-Category Problems, In Proceedings of the Genetic and Evolutionary Computation Conference, pp.656-664, 2024.

13. Haruka Kiyohara, Masahiro Nomura, Yuta Saito : Off-Policy Evaluation of Slate Bandit Policies via Optimizing Abstraction, In Proceedings of the ACM on Web Conference, pp.3150-3161, 2024.

14. Yohei Watanabe, Kento Uchida, Ryoki Hamano, Shota Saito, Masahiro Nomura, Shinichi Shirakawa : (1+1)-CMA-ES with Margin for Discrete and Mixed-Integer Problems, In Proceedings of the Genetic and Evolutionary Computation Conference, pp.882-890, 2023.

15. Shion Takeno, Masahiro Nomura, Masayuki Karasuyama : Towards Practical Preferential Bayesian Optimization with Skew Gaussian Processes, In Proceedings of the International Conference on Machine Learning, pp.33516-33533, 2023.

16. Masahiro Nomura, Isao Ono : Fast Moving Natural Evolution Strategy for High-Dimensional Problems, In IEEE Congress on Evolutionary Computation, pp.1–8, 2022.

17. Ryoki Hamano, Shota Saito, Masahiro Nomura, Shinichi Shirakawa : Benchmarking CMA-ES with Margin on the bbob-mixint Testbed, In Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp.1708-1716, 2022.

18. Ryoki Hamano, Shota Saito, Masahiro Nomura, Shinichi Shirakawa : CMA-ES with Margin: Lower-Bounding Marginal Probability for Mixed-Integer Black-Box Optimization, In Proceedings of the Genetic and Evolutionary Computation Conference, pp.639-647, 2022. (Best Paper Nomination at ENUM Track)

19. Yuta Saito, Masahiro Nomura : Towards Resolving Propensity Contradiction in Offline Recommender Learning, In Proceedings of the International Joint Conference on Artificial Intelligence, pp.2211-2217, 2022.

20. Masahiro Nomura, Isao Ono : Natural Evolution Strategy for Unconstrained and Implicitly Constrained Problems with Ridge Structure, IEEE Symposium Series on Computational Intelligence, pp.1-7, 2021.

21. Masahiro Nomura*, Yuta Saito* : Efficient Hyperparameter Optimization under Multi-Source Covariate Shift, In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp.1376-1385, 2021.

22. Masahiro Nomura, Nobuyuki Sakai, Nobusumi Fukushima, Isao Ono : Distance-weighted Exponential Natural Evolution Strategy for Implicitly Constrained

Black-Box Function Optimization, IEEE Congress on Evolutionary Computation, pp.1099-1106, 2021.

23. Masahiro Nomura*, Shuhei Watanabe*, Youhei Akimoto, Yoshihiko Ozaki, Masaki Onishi : Warm Starting CMA-ES for Hyperparameter Optimization, In Proceedings of the AAAI Conference on Artificial Intelligence, pp.9188-9196, 2021.

24. Shintaro Takenaga, Shuhei Watanabe, Masahiro Nomura, Yoshihiko Ozaki, Masaki Onishi, Hitoshi Habe : Evaluating Initialization of Nelder-Mead Method for Hyperparameter Optimization in Deep Learning, International Conference on Pattern Recognition, pp.3372-3379, 2020.

Appendix A

Additional Details in Chapter 3

A.1 Landscape with stochastic relaxation for Sphere Function

Figure A.1 illustrates the landscape of the Sphere function with stochastic relaxation. Note that the expectation of the objective function value on the Sphere function is $\mathbb{E}[x] = m^2 + v$. It can be observed that the landscape is globally unimodal, making the optimization easier compared to the Rastrigin function, as expected.

A.2 Sensitivity Analysis of Hyperparameters

The proposed learning rate adaptation method in Chapter 3 has two hyperparameters, α and β. In this section, we present an empirical analysis of the hyperparameters sensitivity. We evaluate the sensitivity to α over the range $\alpha \in \{1.2, 1.3, 1.4, 1.5\}$ for several values of λ, with β fixed at 0.2. Next, we evaluate the sensitivity to β over the range $\beta \in \{0.1, 0.2, 0.3, 0.4\}$ for several values of λ, with α fixed at 1.3. The other experimental settings are the same as those in Chapter 3.

Figure A.2 and A.3 shows the SP1 values and success rates with respect to α for the benchmark problems. In terms of the SP1 values, the different α values mildly impacted the overall performance. However, in terms of the success rates, an excessively small α leads to failure on the Ellipsoid function. Therefore, further investigation into how to appropriately set the rationale α is required to

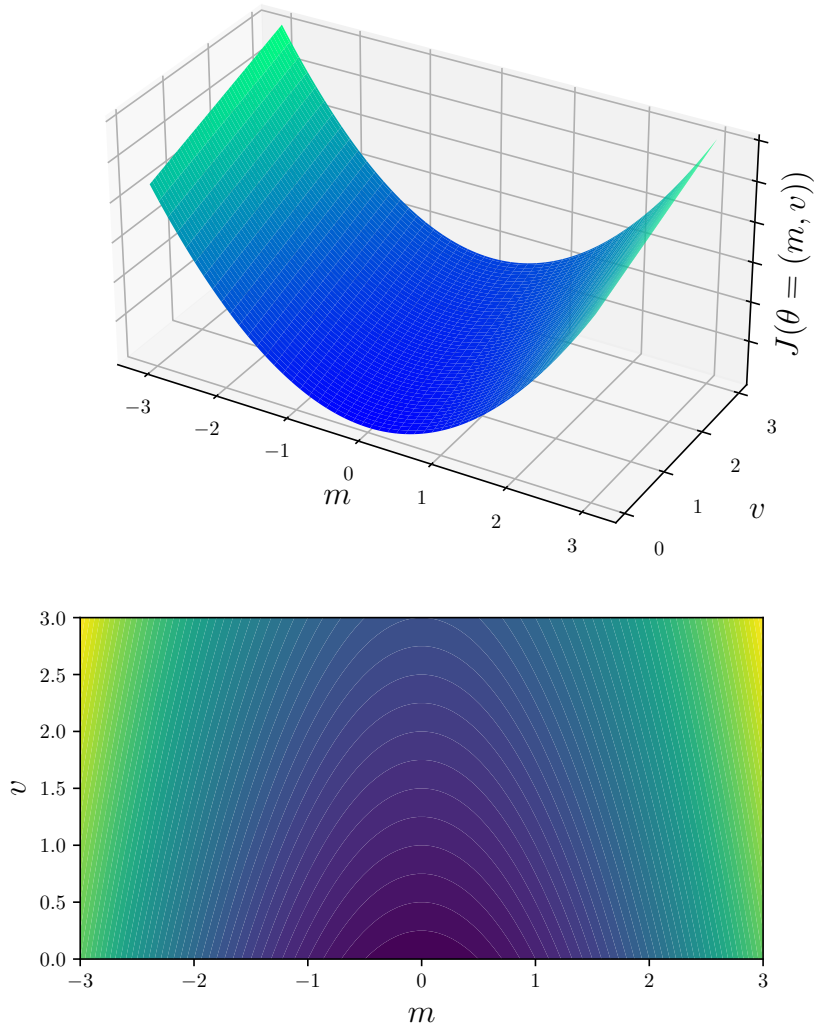


Figure A.1: Landscape of the Sphere function with stochastic relaxation (θ-space). The upper figure shows the 3D plot, while the lower figure presents the 2D plot with contour lines.

achieve stable optimization.

Figure A.4 and A.5 shows the SP1 values and success rates with respect to β for the benchmark problems. We can observe that β has a greater impact on the

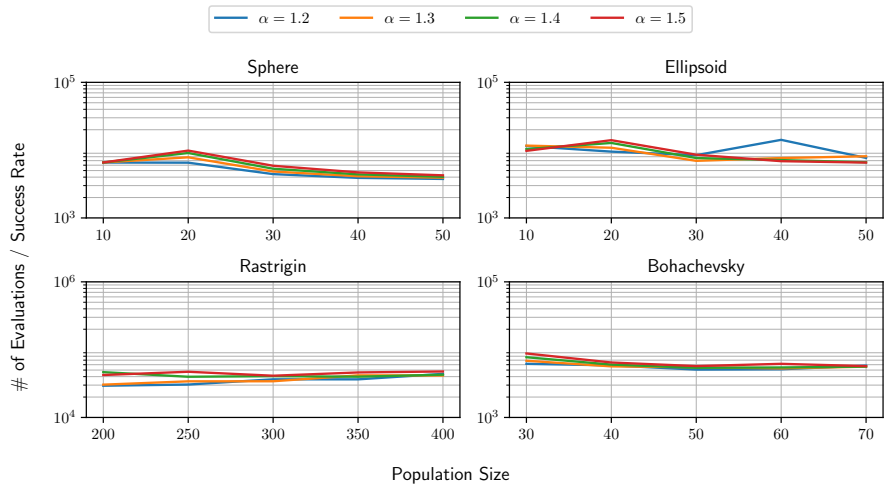


Figure A.2: SP1 values with hyperparameter α for 10-D problems (20 trials). In the experiments, we set $\beta = 0.2$.

performance than α. For example, an excessively small β tends to be an significant failure on the Ellipsoid and Bohachevsky functions. While the success rates are not primary goal of this study, it is important to find a safer configuration of the hyperparameters.

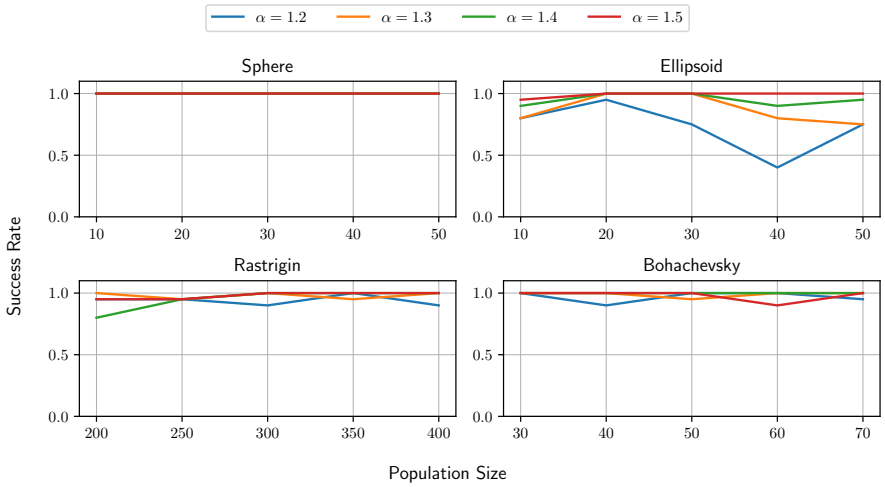


Figure A.3: Success rates with hyperparameter α for 10-D problems (20 trials). In the experiments, we set $\beta = 0.2$.

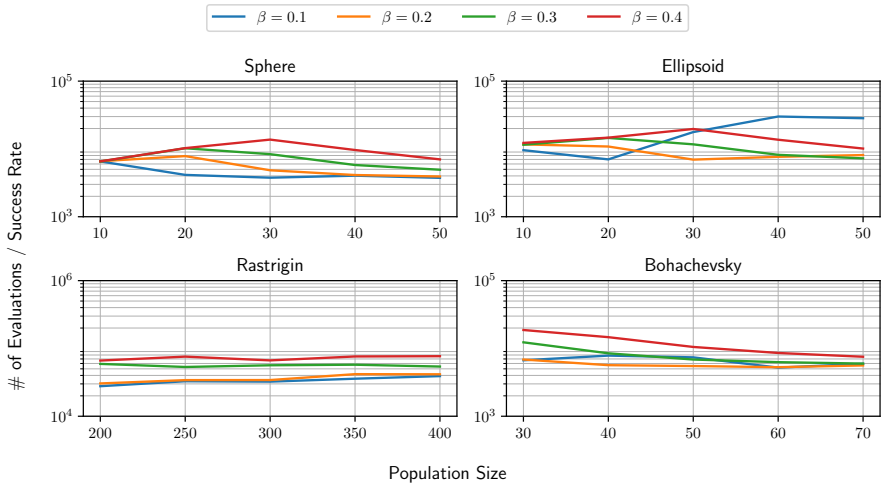


Figure A.4: SP1 values with hyperparameter β for 10-D problems (20 trials). In the experiments, we set $\alpha = 1.3$.

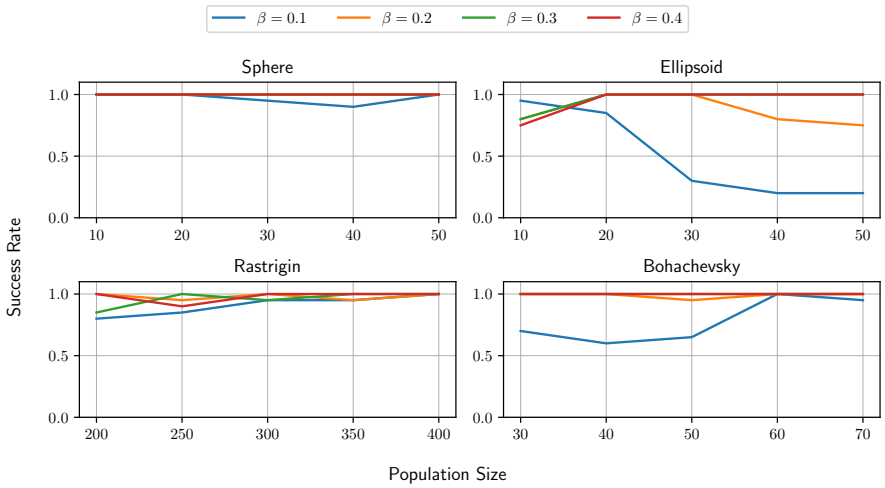


Figure A.5: Success rates with hyperparameter β for 10-D problems (20 trials). In the experiments, we set $\alpha = 1.3$.

Appendix B

Additional Details in Chapter 4

B.1 Derivation for Section 4.3.2

B.1.1 Derivations of Eq. (4.21)

This section presents the detailed derivation of Eq. (4.21). By ignoring $(1-\beta)^n$, $\mathcal{E}^{(t+n)}$ can be approximately calculated as follows:

$$\begin{aligned}
\mathcal{E}^{(t+n)} &= (1-\beta)\mathcal{E}^{(t+n-1)} + \beta\tilde{\Delta}^{(t+n-1)} \\
&= (1-\beta)\left\{(1-\beta)\mathcal{E}^{(t+n-2)} + \beta\tilde{\Delta}^{(t+n-2)}\right\} + \beta\tilde{\Delta}^{(t+n-1)} \\
&= \ldots \\
&= (1-\beta)^n\mathcal{E}^{(t)} + \sum_{i=0}^{n-1}(1-\beta)^i\beta\tilde{\Delta}^{(t+n-1-i)} \\
&\approx \sum_{i=0}^{n-1}(1-\beta)^i\beta\tilde{\Delta}^{(t+n-1-i)}.
\end{aligned}$$

Here, we assume the $\tilde{\Delta}^{(\cdot)}$ are uncorrelated with each other; this corresponds to the scenario where η is sufficiently small. In this case, we can ignore the dependence of t, that is, $\mathbb{E}[\tilde{\Delta}^{(t+n-1-i)}] =: \mathbb{E}[\tilde{\Delta}]$. Thus,

$$\mathbb{E}[\mathcal{E}^{(t+n)}] = \sum_{i=0}^{n-1}(1-\beta)^i\beta\mathbb{E}[\tilde{\Delta}].$$

where

$$\sum_{i=0}^{n-1}(1-\beta)^i = \frac{1\cdot\{1-(1-\beta)^n\}}{1-(1-\beta)} = \frac{1-(1-\beta)^n}{\beta}.$$

Subsequently, ignoring $(1-\beta)^n$, we approximate $\mathbb{E}[\mathcal{E}^{(t+n)}]$ as

$$\mathbb{E}[\mathcal{E}^{(t+n)}] = [1-(1-\beta)^n]\mathbb{E}[\tilde{\Delta}] \approx \mathbb{E}[\tilde{\Delta}].$$

Next, we consider the covariance $\mathrm{Cov}[\mathcal{E}^{(t+n)}]$:

$$\mathrm{Cov}[\mathcal{E}^{(t+n)}] = \mathbb{E}[\mathcal{E}^{(t+n)}(\mathcal{E}^{(t+n)})^\top] - \mathbb{E}[\mathcal{E}^{(t+n)}](\mathbb{E}[\mathcal{E}^{(t+n)}])^\top.$$

We first determine the exact expression for $\mathcal{E}^{(t+n)}(\mathcal{E}^{(t+n)})^\top$ as follows:

$$\begin{aligned}\mathcal{E}^{(t+n)}(\mathcal{E}^{(t+n)})^\top &= \beta^2 \sum_{i=0}^{n-1}(1-\beta)^{2i}\tilde{\Delta}^{(t+n-1-i)}(\tilde{\Delta}^{(t+n-1-i)})^\top \\ &+ \beta^2 \sum_{i,j=0:i\neq j}^{n-1}(1-\beta)^i(1-\beta)^j\tilde{\Delta}^{(t+n-1-i)}(\tilde{\Delta}^{(t+n-1-j)})^\top.\end{aligned}$$

Note that, for $i, j \in \{0,\dots,n-1\}(i \neq j)$, $\mathbb{E}[\tilde{\Delta}^{(t+n-1-i)}(\tilde{\Delta}^{(t+n-1-j)})^\top] = \mathbb{E}[\tilde{\Delta}](\mathbb{E}[\tilde{\Delta}])^\top$ because we assume that they are not correlated. For $i \in \{0,\dots,n-1\}$, $\mathbb{E}[\tilde{\Delta}^{(t+n-1-i)}(\tilde{\Delta}^{(t+n-1-i)})^\top] = \mathbb{E}[\tilde{\Delta}](\mathbb{E}[\tilde{\Delta}])^\top + \mathrm{Cov}[\tilde{\Delta}]$. Thus,

$$\begin{aligned}&\mathbb{E}[\mathcal{E}^{(t+n)}(\mathcal{E}^{(t+n)})^\top] \\ &\quad= \beta^2 \sum_{i=0}^{n-1}(1-\beta)^{2i}\left(\mathbb{E}[\tilde{\Delta}](\mathbb{E}[\tilde{\Delta}])^\top + \mathrm{Cov}[\tilde{\Delta}]\right) \\ &\qquad+ \beta^2 \sum_{i,j=0:i\neq j}^{n-1}(1-\beta)^i(1-\beta)^j\mathbb{E}[\tilde{\Delta}](\mathbb{E}[\tilde{\Delta}])^\top, \\ &\quad= \mathbb{E}[\mathcal{E}^{(t+n)}](\mathbb{E}[\mathcal{E}^{(t+n)}])^\top + \beta^2 \sum_{i=0}^{n-1}(1-\beta)^{2i}\,\mathrm{Cov}[\tilde{\Delta}].\end{aligned}$$

Therefore,

$$\begin{aligned}\mathrm{Cov}[\mathcal{E}^{(t+n)}] &= \mathbb{E}[\mathcal{E}^{(t+n)}(\mathcal{E}^{(t+n)})^\top] - \mathbb{E}[\mathcal{E}^{(t+n)}](\mathbb{E}[\mathcal{E}^{(t+n)}])^\top \\ &= \beta^2 \sum_{i=0}^{n-1}(1-\beta)^{2i}\,\mathrm{Cov}[\tilde{\Delta}].\end{aligned}$$

Here,

$$\sum_{i=0}^{n-1}(1-\beta)^{2i} = \frac{1-(1-\beta)^{2n}}{1-(1-\beta)^2} = \frac{1-(1-\beta)^{2n}}{\beta(2-\beta)}.$$

Thus, by ignoring $(1-\beta)^{2n}$, $\mathrm{Cov}[\mathcal{E}^{(t+n)}]$ can be approximated as

$$\begin{aligned}\mathrm{Cov}[\mathcal{E}^{(t+n)}] &= [1-(1-\beta)^{2n}]\frac{\beta}{2-\beta}\,\mathrm{Cov}[\tilde{\Delta}],\\ &\approx \frac{\beta}{2-\beta}\,\mathrm{Cov}[\tilde{\Delta}].\end{aligned}$$

Therefore, $\mathcal{E}^{(t+n)}$ approximately follows the following distribution:

$$\mathcal{E}^{(t+n)} \sim \mathcal{D}\left(\mathbb{E}[\tilde{\Delta}], \frac{\beta}{2-\beta}\,\mathrm{Cov}[\tilde{\Delta}]\right).$$

Thus, the derivation of Eq. (4.21) is complete.

B.1.2 Derivation of Estimates for $\|\mathbb{E}[\tilde{\Delta}]\|_2^2$

We organized the relation between $\mathcal{E}$ and $\tilde{\Delta}$ using the following equation:

$$\begin{aligned}\mathbb{E}[\|\mathcal{E}\|_2^2] &= \mathbb{E}[\mathcal{E}]^\top I\mathbb{E}[\mathcal{E}] + \mathrm{Tr}(\mathrm{Cov}[\mathcal{E}])\\ &\approx \|\mathbb{E}[\tilde{\Delta}]\|_2^2 + \mathrm{Tr}\left(\frac{\beta}{2-\beta}\,\mathrm{Cov}[\tilde{\Delta}]\right)\\ &= \|\mathbb{E}[\tilde{\Delta}]\|_2^2 + \frac{\beta}{2-\beta}\,\mathrm{Tr}(\mathrm{Cov}[\tilde{\Delta}]).\end{aligned}$$

Now, we apply the same arguments to $\mathcal{V}$ and obtain:

$$\begin{aligned}\mathbb{E}[\mathcal{V}] &= [1-(1-\beta)^{t+1}]\mathbb{E}[\|\tilde{\Delta}\|_2^2]\\ &\approx \mathbb{E}[\|\tilde{\Delta}\|_2^2] = \|\mathbb{E}[\tilde{\Delta}]\|_2^2 + \mathrm{Tr}(\mathrm{Cov}[\tilde{\Delta}]).\end{aligned}$$

By reorganizing these arguments, we obtain

$$\|\mathbb{E}[\tilde{\Delta}]\|_2^2 \approx \frac{2-\beta}{2-2\beta}\mathbb{E}[\|\mathcal{E}\|_2^2] - \frac{\beta}{2-2\beta}\mathbb{E}[\mathcal{V}].$$

This provides the rationale for estimating $\frac{2-\beta}{2-2\beta}\|\mathcal{E}\|_2^2 - \frac{\beta}{2-2\beta}\mathcal{V}$ for $\|\mathbb{E}[\tilde{\Delta}]\|_2^2$.

B.2 On the Twice Differentiability of $J(\theta)$

In Section 4.2.3, we assumed that $J(\theta)(= E_{x\sim p(x;\theta)}[f(x)])$ is twice differentiable. In this section, we formally determine the conditions that $J(\theta)$ is twice differentiable. Assume the following conditions, where i and j are indices for distribution parameters:

1. $p(x;\theta)$ is twice differentiable with respect to θ.
2. The following integrals converge:
   - $\int |f(x)p(x;\theta)|dx < \infty$ (finite expected value)
   - $\int |f(x)\frac{\partial p(x;\theta)}{\partial \theta_i}|dx < \infty$ (1st-order partial derivative)
   - $\int |f(x)\frac{\partial^2 p(x;\theta)}{\partial \theta_i \partial \theta_j}|dx < \infty$ (2nd-order partial derivative)

Here, we prove these are necessary and sufficient conditions for the twice differentiable of $J(\theta)$.

Sufficiency: Assume the stated conditions hold. We prove that $J(\theta)$ is twice differentiable. For the first derivative, by using the definition of the derivative, we write:

$$\frac{\partial J(\theta)}{\partial \theta_i} = \lim_{\Delta\theta_i \to 0} \frac{J(\theta + \Delta\theta_i e_i) - J(\theta)}{\Delta\theta_i}, \tag{B.1}$$

where e_i is the unit vector in the i-th direction. Expanding $J(\theta)$, we have:

$$\frac{J(\theta + \Delta\theta_i e_i) - J(\theta)}{\Delta\theta_i} = \frac{1}{\Delta\theta_i} \int f(x)(p(x;\theta + \Delta\theta_i e_i) - p(x;\theta))dx. \tag{B.2}$$

Since $p(x;\theta)$ is continuously differentiable, the following pointwise limit exists:

$$\lim_{\Delta\theta_i \to 0} \frac{p(x;\theta + \Delta\theta_i e_i) - p(x;\theta)}{\Delta\theta_i} = \frac{\partial p(x;\theta)}{\partial \theta_i}. \tag{B.3}$$

Furthermore, the condition $\int |f(x)\frac{\partial p(x;\theta)}{\partial \theta_i}|dx < \infty$ ensures that the dominant convergence theorem applies, allowing the interchange of the limit and the integral:

$$\frac{\partial J(\theta)}{\partial \theta_i} = \int f(x)\frac{\partial p(x;\theta)}{\partial \theta_i}dx. \tag{B.4}$$

Next, the second derivative is computed as:

$$\frac{\partial^2 J(\theta)}{\partial \theta_i \partial \theta_j} = \frac{\partial}{\partial \theta_j} \int f(x)\frac{\partial p(x;\theta)}{\partial \theta_i}dx. \tag{B.5}$$

The partial derivative inside the integral is given by:

$$\frac{\partial}{\partial \theta_j}\left(f(x)\frac{\partial p(x;\theta)}{\partial \theta_i}\right) = f(x)\frac{\partial^2 p(x;\theta)}{\partial \theta_i \partial \theta_j}. \tag{B.6}$$

By the condition $\int |f(x)\frac{\partial^2 p(x;\theta)}{\partial \theta_i \partial \theta_j}|dx < \infty$, the integral is finite and the dominant convergence theorem again allows the interchange of the integral and the derivative:

$$\frac{\partial^2 J(\theta)}{\partial \theta_i \partial \theta_j} = \int f(x)\frac{\partial^2 p(x;\theta)}{\partial \theta_i \partial \theta_j}dx. \tag{B.7}$$

This proves that $J(\theta)$ is twice differentiable under the given conditions.

Necessity: Assume $J(\theta)$ is twice differentiable. To ensure the validity of the derivatives, for the first derivative:

$$\frac{\partial J(\theta)}{\partial \theta_i} = \int f(x)\frac{\partial p(x;\theta)}{\partial \theta_i}dx \tag{B.8}$$

to exist, the term $\int |f(x)\frac{\partial p(x;\theta)}{\partial \theta_i}|dx$ must converge. Thus, $\int |f(x)\frac{\partial p(x;\theta)}{\partial \theta_i}|dx < \infty$ is necessary.

Similarly, for the second derivative:

$$\frac{\partial^2 J(\theta)}{\partial \theta_i \partial \theta_j} = \int f(x)\frac{\partial^2 p(x;\theta)}{\partial \theta_i \partial \theta_j}dx \tag{B.9}$$

to exist, the term $\int |f(x)\frac{\partial^2 p(x;\theta)}{\partial \theta_i \partial \theta_j}|dx$ must also converge. Thus, $\int |f(x)\frac{\partial^2 p(x;\theta)}{\partial \theta_i \partial \theta_j}|dx < \infty$ is necessary.

Therefore, the stated conditions are both sufficient and necessary for $J(\theta)$ to be twice differentiable with respect to to θ.

We then focus on the case where $p(x;\theta)$ is multivariate Gaussian distribution, as it applies to our scenario. It is clear that $p(x;\theta)$ is twice differentiable, as the multivariate Gaussian is infinitely differentiable with respect to both x and θ. Furthermore, the rapid decay of $p(x;\theta)$ at the tails ensures that $f(x)$ only contributes significantly in regions where $p(x;\theta)$ has non-negligible density. As a result, if $f(x)$ is bounded, the smoothing effect of $p(x;\theta)$ ensures that $J(\theta)$ is twice differentiable. On the other hand, if $f(x)$ is unbounded or its discontinuities induce divergence in the integrals for the derivatives, $J(\theta)$ may fail to be twice differentiable.

It is worth noting that the discontinuity of $f(x)$ does not necessarily imply that $J(\theta)$ is non-differentiable. This is because the smoothing effect of $p(x;\theta)$ mitigates discontinuities in $f(x)$ during the integration. Consider, for example, the piecewise constant function with a finite number of discontinuities, such as:

$$f(x) = \begin{cases} 1 & \text{if } x \geqslant 0, \\ 0 & \text{if } x < 0. \end{cases} \tag{B.10}$$

The product $f(x)\frac{\partial^2 p(x;\theta)}{\partial\theta_i\partial\theta_j}$ remains well-behaved because $p(x;\theta)$ is smooth, and the discontinuity in $f(x)$ does not have a critical effect for the result. Thus, $J(\theta)$ is twice differentiable.

B.3 Theoretical and Empirical Insights into SNR

In Section 4.3, we assumed that the signal-to-noise ratio (SNR) is relatively small, for example, SNR $\lessapprox 1$, which validates the approximation $1/(1+\mathrm{SNR}^{-1}) \approx \mathrm{SNR}$. In this section, we theoretically and empirically discuss the validity of SNR $\lessapprox 1$.

To obtain useful insights into this SNR from a theoretical perspective, we considered observing it in a situation wherein the objective function is the sphere function $f(x) = \|x\|^2$ and the covariance matrix is $\Sigma = \sigma^2 I$, where $\sigma = \bar{\sigma}\frac{\|m\|}{d}$ and $\bar{\sigma}$ is called the normalized step-size. The quality gain analysis [11, 3] implies that for a sufficiently large d, the distribution of the ith ranked solution among the λ candidate solutions is approximated as $X_{i:\lambda} = m + \sigma\mathcal{N}_{i:\lambda}\frac{m}{\|m\|} + \sigma\mathcal{N}_i^{\perp}$, where $\mathcal{N}_{i:\lambda}$ is the ith order statistics among λ normally distributed random variables and $\mathcal{N}_i^{\perp}$ is an independently distributed d dimensional normal random vector with covariance matrix $I - \frac{mm^{\mathrm{T}}}{\|m\|^2}$ if $m \neq 0$. Using this approximation, we obtain $\Delta_m = \sigma\left(\sum_{i=1}^{\lambda} w_i\mathcal{N}_{i:\lambda}\right)\frac{m}{\|m\|} + \sigma\left(\sum_{i=1}^{\lambda} w_i\mathcal{N}_i^{\perp}\right)$. Let $\boldsymbol{w} = (w_1, \ldots, w_\lambda)$, $\boldsymbol{n}_{(\lambda)} = (\mathbb{E}[\mathcal{N}_{1:\lambda}], \ldots, \mathbb{E}[\mathcal{N}_{\lambda:\lambda}])$. $\boldsymbol{N}_{(\lambda)}$ is a matrix whose (i,j)th element is $\mathbb{E}[\mathcal{N}_{i:\lambda}\mathcal{N}_{j:\lambda}]$. Then, we obtain

$$\mathbb{E}[\Delta_m] = \sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{n}_{(\lambda)})\frac{m}{\|m\|}, \tag{B.11a}$$

$$\mathbb{E}[\Delta_m\Delta_m^{\mathrm{T}}] = \sigma^2(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{N}_{(\lambda)}\boldsymbol{w})\frac{mm^{\mathrm{T}}}{\|m\|^2} + \sigma^2\|\boldsymbol{w}\|^2\left(I - \frac{mm^{\mathrm{T}}}{\|m\|^2}\right). \tag{B.11b}$$

Because $F_m = \sigma^{-2} I$, we obtain

$$\mathrm{SNR} = \frac{\sigma^{-2}\|\mathbb{E}[\Delta_m]\|^2}{\sigma^{-2}\,\mathrm{Tr}(\mathbb{E}[\Delta_m\Delta_m^{\mathrm{T}}]) - \sigma^{-2}\|\mathbb{E}[\Delta_m]\|^2} \tag{B.12a}$$

$$= \frac{(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{n}_{(\lambda)})^2}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{N}_{(\lambda)}\boldsymbol{w} + (d-1)\|\boldsymbol{w}\|^2 - (\boldsymbol{w}^{\mathrm{T}}\boldsymbol{n}_{(\lambda)})^2} \tag{B.12b}$$

$$\approx \frac{(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{n}_{(\lambda)})^2}{(d-1)\|\boldsymbol{w}\|^2} \tag{B.12c}$$

$$= \frac{1}{d-1}\frac{(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{n}_{(\lambda)})^2}{\|\boldsymbol{w}\|^2} \tag{B.12d}$$

$$\approx \frac{\lambda}{d-1}\frac{(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{n}_{(\lambda)})^2}{\|\boldsymbol{w}\|^2\|\boldsymbol{n}_{(\lambda)}\|^2}. \tag{B.12e}$$

Here, we used the following asymptotically true approximations for λ (See Eq. (A2) provided in [3]):

$$\frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{N}_{(\lambda)}\boldsymbol{w}}{(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{n}_{(\lambda)})^2} \approx 1 \qquad \text{and} \qquad \frac{\|\boldsymbol{n}_{(\lambda)}\|^2}{\lambda} \approx 1. \tag{B.13}$$

It should be noted that $\frac{(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{n}_{(\lambda)})^2}{\|\boldsymbol{w}\|^2\|\boldsymbol{n}\|^2}$ is upper bounded by 0.25 if only non-negative weights are used for the m-update, which aligns with our weight scheme. Therefore, we can expect that SNR $\lessapprox 1$ holds if λ is not considerably large relative to d; for example, $\lambda \leqslant 4(d-1)$. Importantly, in difficult problems, such as multimodal problems, the SNR tends to be smaller than that in the sphere functions. Therefore, it should be noted that the assumption of SNR $\lesssim 1$ becomes more easily valid for such difficult problems.

It should be noted that the aforementioned analysis only considers The main limitation of the aforementioned analysis is to the assumption that the dimension d and the population size λ are sufficiently large. To verify whether the assumption SNR $\lessapprox 1$ works in practice, we conducted experiments using the LRA-CMA-ES for 30-dimensional Sphere, Schaffer, and Rastrigin functions using the same settings as those mentioned in Section 4.4.1, and $\lambda = 14$ for $d = 30$. Figure B.1 illustrates the typical behavior of the estimated SNR where it was estimated using the method described in Section 4.3.2. It should be noted that this value includes estimation errors. Although the estimated SNR for the covariance in the Sphere function tends to be slightly larger, it often remains under 1, particularly for more *difficult* problems such as the Rastrigin function. These results

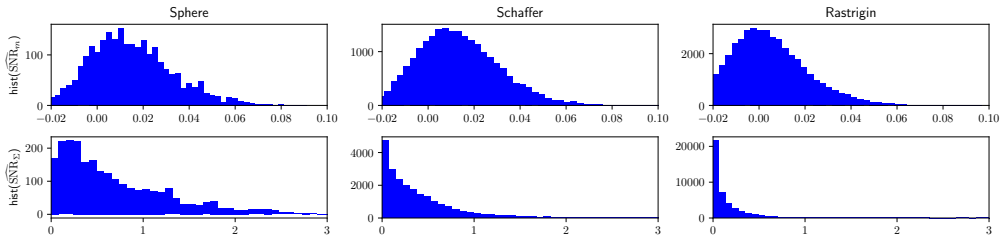


Figure B.1: Histogram of the estimated SNR in typical trials on 30-D noiseless problems. Estimated SNR with respect to (top) the mean vector m and (bottom) the covariance matrix Σ. The SNR was estimated using the method described in Section 4.3.2.

suggest that the assumption of SNR to be small, e.g., SNR $\lessapprox 1$, appears to be valid to a certain degree even under finite dimensions and population sizes.

B.4 Guidelines for Hyperparameter Settings

In this section, we discuss the guidelines for hyperparameter settings of LRA-CMA-ES. As described in the main text, a particular point of LRA-CMA-ES is that it does not basically require hyperparameter tuning of the sample size λ. The reason is that LRA-CMA-ES behaves as if it adapts the population size to maintain a constant update speed (although, in practice, it adapts the learning rate instead, leading to similar behavior). Therefore, the only rule to follow when setting the population size is straightforward: if our parallel environment has a preferred sample size (e.g., the number of workers), use it; otherwise, leave it unspecified, and the recommended default value [35] ($\lambda = 4 + \lfloor 3\log(d) \rfloor$) will be applied.

However, we can utilize the knowledge about the problem structure (e.g., multimodality) if available to accelerate the optimization. A relatively easy way to do that is to set an appropriate α, which corresponds to the target SNR. Figure 4.8 shows the success rates and SP1 values (the average number of evaluations among successful trials until achieving the target value divided by the success rate) with respect to α for the 30-dimensional (30-D) noiseless Sphere, Schaffer, and Rastrigin functions. We can observe that for the Sphere function (i.e., an unimodal problem), a better performance could be achieved with a smaller α value. However, an excessively large α results in optimization failures for multimodal problems. Based on this result, we determined the recommended value to

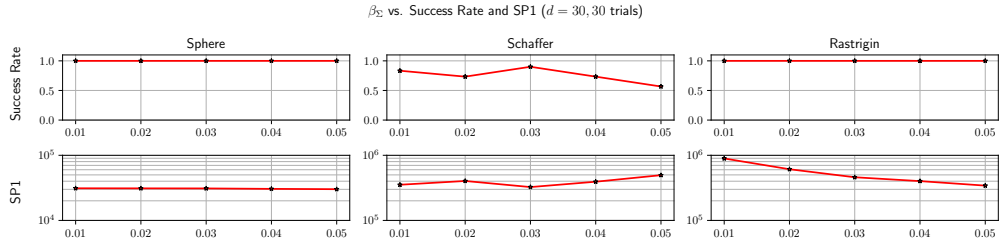


Figure B.2: Success rate and SP1 values with hyperparameter $\beta_\Sigma \in \{0.01, 0.02, ..., 0.05\}$ on 30-D noiseless problems.

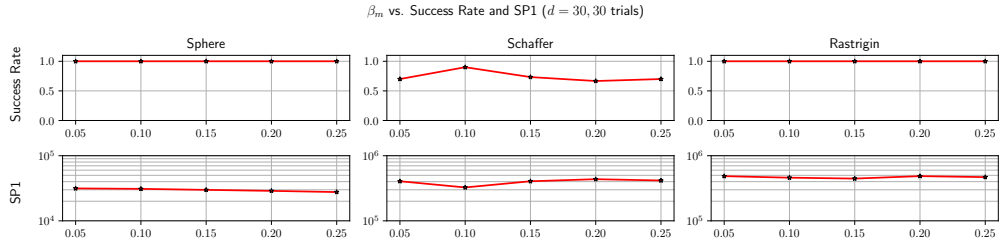


Figure B.3: Success rate and SP1 values with hyperparameter β_m for 30-D noiseless problems.

be $\alpha = 1.4$ in the paper, a conservative setting better suited for difficult problems. However, if the problem is known to be nearly unimodal, using a smaller α (e.g., $\alpha = 0.2$ or 0.6) could accelerate the optimization process.

B.5 Additional Experimental Results

Figure B.2 shows the success rate and SP1 values with respect to $\beta_\Sigma \in \{0.01, 0.02, ..., 0.05\}$ for the 30-D noiseless Sphere, Schaffer, and Rastrigin functions. Clearly, the performance is not significantly affected by β_Σ values within this range. However, as shown in Figure 4.9, an excessively small β_Σ value decelerates the convergence for the Rastrigin function.

Figures B.3 and B.4 show the success rates and SP1 values for β_m and γ, respectively. The results show that the performance is relatively stable against these hyperparameters.

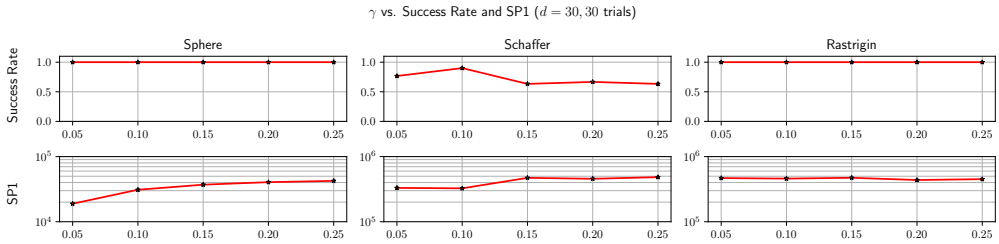


Figure B.4: Success rate and SP1 values with hyperparameter γ for 30-D noiseless problems.