Bachelor Thesis

Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

# Evolutionary Optimization Using Random Embeddings

Dominik Nguyen

Supervisor: Ing. Petr Pošík, Ph.D.
May 2024

# BACHELOR'S THESIS ASSIGNMENT

## I. Personal and study details

Student's name: **Nguyen Dominik** Personal ID number: **503223**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Specialisation: **Artificial Intelligence and Computer Science**

## II. Bachelor's thesis details

Bachelor's thesis title in English:

**Evolutionary Optimization Using Random Embeddings**

Bachelor's thesis title in Czech:

**Evoluční optimalizace v náhodných lineárních podprostorech**

Guidelines:

1) Survey the usage of random embeddings in black-box optimization algorithms.
2) Find or design a test suite of benchmark functions with varying type, dimensionality, intrinsic dimensionality, and other features, that shall be used to assess the efficiency of the optimization algorithm.
3) Choose a variant of evolutionary algorithm and implement an interface that will allow it to use (random) embeddings.
4) Compare the performance of the optimization algoritm with and without random embedding on the benchmark problems. Compare random embeddings also to some deterministically chosen embeddings (PCA, …).
5) Propose and implement a restarting scheme for the algorithm and assess its effects to the optimization process.

Bibliography / sources:

[1] Letham, Benjamin et al. "Re-Examining Linear Embeddings for High-Dimensional Bayesian Optimization", NeurIPS 2020
[2] Sanyang, Momodou L. and Ata Kabán. "REMEDA: Random Embedding EDA for Optimising Functions with Intrinsic Dimension". In: Parallel Problem Solving from Nature – PPSN XIV, 2016
[3] Cartis, Coralia and Adilet Otemissov. "A dimensionality reduction technique for unconstrained global optimization of functions with low effective dimensionality". In: Information and Inference: A Journal of the IMA 11.1, pp. 167–201, 2022

Name and workplace of bachelor's thesis supervisor:

**Ing. Petr Pošík, Ph.D. Department of Cybernetics FEE**

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **30.01.2023** Deadline for bachelor thesis submission: **24.05.2024**

Assignment valid until: **22.09.2024**

| Ing. Petr Pošík, Ph.D. | prof. Ing. Tomáš Svoboda, Ph.D. | prof. Mgr. Petr Páta, Ph.D. |
|---|---|---|
| Supervisor's signature | Head of department's signature | Dean's signature |

## III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

.

| Date of assignment receipt | Student's signature |
|---|---|

# Acknowledgements

I would like to express my deepest gratitude to my supervisor, Ing. Petr Pošík, Ph.D., for his invaluable guidance, especially in the early stages of the research. Special thanks to my family for their unwavering support. I would like to also thank my long-time friend David for his encouragement and help with the proofreading.

# Declaration

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 24.5.2024

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 24.5.2024

Dominik Nguyen

# Abstract

This thesis explores the use of random embeddings in evolutionary optimization algorithms to address high-dimensional black-box optimization problems. By embedding high-dimensional spaces into lower-dimensional subspaces, we aim to enhance the efficiency and effectiveness of optimization algorithms. Our experiments demonstrate the potential of random embeddings to improve the performance of Evolutionary Algorithms.



**Keywords:** optimization, evolutionary algorithms, random embeddings, dimensionality reduction, covariance matrix adaptation evolution strategy (CMA-ES), machine learning, hyperparameter optimization


**Supervisor:** Ing. Petr Pošík, Ph.D.


# Abstrakt

Tato práce zkoumá použití náhodných vnoření v evolučních optimalizačních algoritmech k řešení problémů vysokodimenzionálních optimalizací typu černé skříňky. Za pomocí vnoření vysokodimenzionálních prostorů do prostorů s nižší dimenzí chceme zlepšit výkonnost a efektivitu optimalizačních algoritmů. Naše obsáhlé experimenty demonstrují dosud neprozkoumaný potenciál zlepšení výkonnosti evoluční strategie CMA-ES v kombinaci s metodou náhodného vnoření pro specifické případy.



**Klíčová slova:** optimalizace, evoluční algoritmy, náhodné lineární podprostory, vnoření, redukce dimenzionality, evoluční strategie s adaptací kovarianční matice (CMA-ES), strojové učení, optimalizace hyperparametrů


**Překlad názvu:** Evoluční optimalizace v náhodných lineárních podprostorech

# Contents

# Figures



# Tables

# Part I

# Introduction

# Chapter 1

## Introduction

In the era of big data and complex modeling, high-dimensional data has become increasingly present across various domains, including construction engineering chemistry and most notably machine learning in recent years. High-dimensional datasets, characterized by a large number of features, pose significant challenges for optimization algorithms. Particularly black-box optimization, also known as derivative-free optimization, refers to the optimization of objective functions that are not in the form of closed-form expression. Such functions are treated as “black boxes” where the internal workings are unknown, and only input-output evaluations via querying are available. Those functions have a potentially very high dimensionality and complex structures, making them challenging to optimize.

The scalability of optimization algorithms has thus become an increasingly important research topic. These challenges, often referred to as the “curse of dimensionality,“ include increased computational complexity, difficulty in exploring the solution space, and the risk of overfitting. This issue is particularly pronounced in recent advancements such as large language models (LLMs), which require the optimization of numerous hyperparameters across large feature spaces (Liu et al. (2024)). Consequently, there is a growing need for effective techniques to address these challenges.

There are several classical approaches to reducing the dimensionality of data using linear dimensionality reduction techniques. One of the classical methods is Principal Component Analysis (PCA), which performs a linear mapping from a higher-dimensional space to a lower-dimensional space by identifying the principal components that maximize the variance in the data. Another approach is Random Projection, a stochastic method where dimensionality reduction is achieved using a random matrix. Both PCA and Random Projection are used when dealing with large datasets, as they can simplify the data while preserving features. One such use is face recognition (Hou et al. (2023)).

However, dimensionality reduction is used differently in the context of Evolutionary Algorithms (EAs). The data samples are generated and updated while the EA is running (Hou et al. (2023)). The main approaches how to scale optimization in conjunction with EAs can be categorized into the following groups:

1. Problem decomposition
2. Problem reduction

Both approaches involve the utilization of optimization algorithms within a lower-dimensional subspace. The problem decomposition makes use of the inner structure of the problem, this is not always feasible as the detection of the structure can be computationally expensive and the objective function may not always be decomposed into low-dimension subproblems. The second approach does not rely on gaining knowledge about the problem structure but it relies on the fact that in some cases it was shown that the objective function has low effective dimensionality (in some literature referred to as intrinsic dimension). In this thesis, the aim is to explore how to exploit this property via random embedding.(Omidvar, X. Li, and Yao (2022a), Omidvar, X. Li, and Yao (2022b), Sanyang and Kabán (2016), Z. Wang et al. (2016), James Bergstra and Yoshua Bengio (n.d.)).

Dimensionality reduction is crucial in addressing the challenges posed by high-dimensional black-box optimization. This study aims to investigate the use of random embedding in conjunction with EAs to exploit the underlying low effect dimensionality of the problem. By embedding the high-dimensional space onto a lower-dimensional subspace, dimensionality reduction can significantly enhance the efficiency and effectiveness of optimization algorithms.

# Chapter 2

## Literature review

This work was originally motivated by Sanyang and Kabán (2016), proposing an extension to the Estimation of Distribution Algorithm (EDA) with Random Embedding (REM). EDA is a stochastic optimization method belonging to the class of evolutionary algorithms, which performs well in low-dimensional problems but its performance deteriorates rapidly as the dimensionality increases. Kaban, Bootkrajang, and Durrant (2013) devised a divide-and-conquer approach to mitigate the curse of dimensionality by applying a Random Matrix Theory making use of a series of Random projections. If the dimension is high the random projection algorithm preserves the local separation of the data which is guaranteed by the Johnson-Lindenstrauss lemma (J. Wang 2012). This low local distortion property ensures that the projection approximately preserves the relationship between variables making it possible for EDA to be used on the problem that was initially impossible to solve with EDA.

Unlike the divide-and-conquer approach in Kaban, Bootkrajang, and Durrant (2013), which does not make any assumption about the structure of the problem, the approach suggested by Sanyang and Kabán (2016) assumes low effective dimensionality of the given function. This assumption is based on the observation that for certain problems, only a subset of dimensions significantly affects the objective function, while others remain relatively constant.

One of the suggested approaches (Z. Wang et al. 2016), which was also used in Sanyang and Kabán (2016), is to deal with the curse of dimensionality by applying the dimensionality reduction technique before deploying the chosen optimization algorithm. One such possible technique is random embedding, where ‘embedding’ refers to a structure-preserving mapping and ‘random’ indicates that the mapping is conducted using a random matrix.

The authors in Z. Wang et al. (2016) note that many researchers noticed that for a particular subset of problems, only some of the dimensions affect the objective function. This means that search space can be split into two subspaces, one being effective with respect to the objective function and the other being constant in this regard. To exploit this feature REMBO (Random EMbedding Bayesian Optimization) is proposed as a novel algorithm where the original search space can be embedded into a low-dimensional subspace using Gaussian random embedding. For this method is not necessary to identify the effective subspace, unlike other methods that put effort into learning the effective subspace. So instead of optimizing $f : \mathbb{R}^D \to \mathbb{R}$ we can optimize reduced problem $g(y) = f(Ay)$ where $A \in \mathbb{R}^{D \times d}$ is a random embedding matrix with entries iid $N(0,1)$ and $y \in \mathbb{R}^d$.

The authors note that obtaining values for the black-box function outside of certain constraints is often impractical for many real-world problems. Consequently, while REMBO is designed to solve global optimization problems, the optimization is constrained to a compact set. In some cases, the function being optimized may project outside of the specified box bounds. To address this issue, a convex projection is applied to the projected $y$ values, ensuring they are mapped to the closest point within the box bounds based on the L2 norm. This approach effectively maintains feasibility within the optimization space while accommodating the constraints imposed by the problem

The original REMBO has several shortcomings, which led to attempts to overcome them. In the case of $y$ being projected outside of the box bound it is projected to the closest point in the box bound, which causes a nonlinear mapping and a subsequent nonlinear distortion. To mitigate this, hash-enhanced Subspace Bayesian optimization (HeSBO) was introduced by Nayebi, Munteanu, and Poloczek (2019). It addresses the problem with distortion by modifying the projection to use a hashing matrix to ensure that the points from $\mathbb{R}^D$ fall into $[-1,1]^D$.

While the technique proposed in Z. Wang et al. (2016) is called random embedding Bayesian optimization (REMBO), the authors suggested that the principle behind REMBO can be applied to any arbitrary optimization procedure. The idea of generalizing random embedding technique for the use with arbitrary solvers was presented in Cartis and Otemissov (2022) where the authors investigated and further analyzed the properties of random embedding extending the original REMBO algorithm to Random Embeddings for Global Optimization framework (REGO).

For solvers DIRECT, KNITRO and Baron Cartis and Otemissov (2022) demonstrated that solving reduced problems in lower dimensions is effective, and generally outperforming solvers that did not apply embedding before optimizations. The original dimension is not essential to the success of optimization, but it is sensitive to setting the embedding dimension $d$ and the parameter $\delta : [-\delta, \delta]^d$.

The main difference between REMBO and REGO lies in the constraints. In the REGO the objective function is not subject to any constraints so the projection of the points falling outside of the box bounds is not necessary effectively eliminating the issues with distortions. This however comes at the cost of being unable to solve problems with constraints.

The authors suggest that possible future research using the results of REGO could be done in estimating of effective dimension and the embedding dimension. Another direction of research could go into investigating different techniques and how to generate the random embedding matrix.

## 2.1 Motivation and aim of the thesis

Our research focus is on enhancing the performance of the covariance matrix adaptation evolution strategy (CMA-ES), similar to the experiments of Sanyang and Kabán (2016) where they extend EDA by adding random embedding. CMA-ES already has properties that are sought after in search algorithms – the property of invariance to various linear transformations such as translation, reflection, rotation, or scaling invariance and invariance to scaling of variables (Hansen (2016)). Additionally, it is also empirically very successful in applications involving black-box optimization.

However, this method suffers from the curse of dimensionality as it constructs and updates its covariance matrix of parameters to learn pair-wise dependencies between parameters leading to the complexity of at least $O(n^2)$. Then sampling from the multivariate distribution, which has a complexity of $n^3$, but as it is not performed for every generation, the complexity is again $O(n^2)$(Ros and Hansen (2008)).

There were several attempts to overcome this issue of dimensionality in CMA-ES by modifying it internally to improve its performance in large-scale optimization problems (Ros and Hansen (2008); Jin, Yang, and Zhang (2020) or Tong, Yuan, and B. Li (2019)). Their ideas are examined below.

One technique called sep-CMA-ES, suggested by Ros and Hansen (2008) revolves around reducing the degrees of freedom in the covariance matrix by adding two simple modifications. The first is constraining the covariance matrix $C$ to be diagonal, and the second is increasing the learning rate. Due to those changes, the time complexity of steps in CMA-ES becomes linear as degrees of freedom in the covariance matrix are reduced to $n$.

A different attempt to reduce the degrees of freedom of the covariance matrix was proposed by Tong, Yuan, and B. Li (2019), the so-called CCG-MMC-CMAES.There the trade-off between having more degrees of freedom is pointed out. While it allows for more effective correlation modeling between variables, having fewer degrees of freedom has the advantage of a lower time and space complexity. To control for this dilemma, a combination of two techniques is proposed to extend CMA-ES. Firstly, the correlation-based grouping (CCG) strategy is used to divide variables into two groups –

correlated and not correlated. Then the search is performed with CMA-ES under the model complexity control, which reduces the degrees of freedom.

Another approach to reducing the time complexity of CMA-ES is to replace the covariance matrix entirely. Besides the issues with time complexity, one other shortcoming the authors (Jin, Yang, and Zhang (2020)) mention is that the feature aimed at improving convergence of CMA-ES is also its major drawback. As CMA-ES assesses only some of the best individuals in the population, some if not most, information is lost. On the other hand, the proposed improvement, gradient information ES –*GI-ES*, to the CMA-ES here involves keeping complete information about each generation. The evolution strategy is then guided by gradient information obtained with the stochastic gradient descent algorithm.

Currently, there is no existing research on extending CMA-ES with random embedding, which will be the focus of this thesis. The closest research was done by authors of Sanyang and Kabán (2016) or in their other work where they used multiple random projections for EDA (Kaban, Bootkrajang, and Durrant 2013).

This thesis explores the use of random embeddings in evolutionary optimization algorithms to address high-dimensional black-box optimization problems. By embedding high-dimensional spaces into lower-dimensional subspaces, we aim to enhance the efficiency and effectiveness of optimization algorithms. Our extensive experiments demonstrate the so far unexplored potential of random embeddings to improve the performance of CMA-ES evolutionary algorithms in specific scenarios

# Part II

# Methods and implementation

## Chapter 3

# Random Embedding

In this chapter, we describe the theoretical framework of random embedding and its properties. The aim is to integrate random embedding with Evolutionary Algorithms (EAs), specifically the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), to enhance optimization efficiency in high-dimensional spaces.

## 3.1 Theoretical framework

First, we introduce the concept of effective dimensionality which is essential for the random embedding technique. The effective dimensionality of a function is defined as the dimension of the subspace where the function value remains constant. This property can be exploited to reduce the complexity of the optimization problem by solving it in a lower-dimensional subspace.

**Definition 1.** A function f: $\mathbb{R}^D \to \mathbb{R}$ has an effective dimensionality $d_e < D$ if there exists a subspace $S \subset \mathbb{R}^D$ with dimension $d_e$ such that for all $x \in \mathbb{R}^D$, there exists $x' \in S$ such that $f(x) = f(x')$.

**corollary 2.** Definition 1 implies that besides the effective subspace $S$ there is its orthogonal complement - a constant subspace $S'$ with dimension $D - d_e$. In other words all $x \in \mathbb{R}^D$ can be decomposed so that $x = x' + x''$ and $f(x) = f(x' + x'') = f(x')$ where $x' \in \mathbb{R}^{d_e}$ and $x'' \in \mathbb{R}^{D-d_e}$.

This property suggests that it suffices to only find the optimizer in the effective subspace $S$ as the function value does not change along the coordinates from $S'$, effectively reducing the problem complexity.Cartis and Otemissov (2022) also impose restrictions on possible solutions, the optimizer must be non-trivial. This will be important in designing our experiments.

Corollary 1 can be exploited to reduce the complexity of the given problem by solving it in some d-dimensional subspace where $d_e \leq d \leq D$. See appendix in Z. Wang et al. (2016) for the proof of the following theorem.

**Theorem 3.** *Let a function f:* $\mathbb{R}^D \to \mathbb{R}$ *have an effective dimensionality* $d_e < D$ *and random matrix* $\mathbf{A}^{D\times d}$ *with i.i.d entries* $\mathcal{N}(0,1)$ *where* $d \geq d_e$*. Then, with probability 1, for any* $x \in \mathbb{R}^D$*, there exists* $y \in \mathbb{R}^d$ *such that* $f(x) = f(\mathbf{A}y)$*.*

Theorem 3 gives us the theoretical guarantee that if there is optimizer $x^*$, with probability 1, there exists $y^* \in \mathbb{R}^d$ such that $f(\mathbf{A}y) = f(x^*)$. It can be found by optimizing $g(y) = f(\mathbf{A}y) =$. It is worth noting, as highlighted in Sanyang and Kabán (2016), that the proof was given for the case where the dimension $d$ in *theorem*3 is equal to the effective dimensionality $d_e$, but Sanyang and Kabán (2016) shown that case $d > d_e$ also holds for a search box that is bounded by $\delta$ satisfying certain conditions derived in their work. Cartis and Otemissov (2022) extended it even further allowing any arbitrary $d > d_e$.

While the existence of effective subspace is crucial for the random embedding there are also some guarantees regarding the role of ambient dimension $D$. The *Theorem 4* in Z. Wang et al. (2016) states that the random embedding is invariant to the addition of unimportant dimensions, that is along the axis in the constant space. This invariance means adding dimensions that do not change the function's value and subsequently, it does not affect the performance of the optimization. The theorem is followed by a proof. However, Cartis and Otemissov (2022) also demonstrated the invariance numerically in experiments testing the quality of bounds for the success of solving the reduced problem.

The experiments revealed that ambient dimension $D$ is not as important as the difference between the dimensions $d_e$ and $d$. Specifically the difference between them, as the difference $d - d_e$ increases, the success also goes up. So when choosing the dimension $d$ it is a good idea to choose $d \gg d_e$.

Furthermore, the size of the search box is another crucial factor for the success of optimizing the reduced problem. When appropriately chosen, it ensures that the reduced search space contains the global minimizers with high probability, thus significantly improving the likelihood of finding the optimal solution. However, if it is chosen too large, it decreases the efficiency of optimization algorithms due to increased complexity. In the constrained optimization scenario, where methods like REMBO or REMEDA are used, there is added complexity of choosing the constraints carefully as solutions might lie outside of the original search space. This requires additional transformations such as convex projection (as discussed in Z. Wang et al. (2016)), hash-encoding in Nayebi, Munteanu, and Poloczek (2019) or orthogonal projection in Hou et al. (2023). While the literature provides theoretical guidance, on how to set the search box in low dimension $d$.

Practically, this theoretical guidance is not useful due to the of lack knowledge about the effective dimensionality $d_e$. It is suggested by Z. Wang et al. (2016) and by Sanyang and Kabán (2016) to set the box constraint $[-\delta, \delta]^d$ to $[-\sqrt{d}, \sqrt{d}]^d$. Since the focus of this thesis is on unconstrained optimization, it is the only consideration we need to make regarding the search space. There is no need to choose a method for projecting solutions that fall outside of the box bounds of the original search space. This however will limit the scope of application of the proposed method to a subset of optimization problems.

## 3.2 Proposed Approach

In this section, we describe the implementation of random embeddings within the framework of evolutionary algorithms, specifically focusing on the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). The goal is to leverage random embeddings to reduce the dimensionality of high-dimensional optimization problems, thereby improving the efficiency and effectiveness of the optimization process. Now consider a black-box problem that is given by an objective function $f : \mathbb{R}^D \to \mathbb{R}$ that is to be minimized:

$$\min_{x \in \mathbb{R}^D} f(x) \tag{3.1}$$

Theorem 3 in conjunction with Corollary 2 guarantees that if $f : \mathbb{R}^D \to \mathbb{R}$ has effective dimensionality $d_e < D$, a random matrix $\mathbf{A} \in \mathbb{R}^{D \times d}$ where $d_e \leq d$ such that with probability 1, for any $x \in \mathbb{R}^D$, there exists $y \in \mathbb{R}^d$ such that $f(x) = f(\mathbf{A}y)$. This also holds for optimizer $f^* = f(x^*) = f(\mathbf{A}y^*)$.

$$\begin{aligned} &\min_{y \in \mathbb{R}^d} g(y) = f(Ay) \\ &\text{subject to} \quad y \in [-\delta, \delta]^d \subseteq \mathbb{R}^d \end{aligned} \tag{3.2}$$

Let $f : \mathbb{R}^D \to \mathbb{R}$ be the objective function in a high-dimension $D$ and random projection matrix $A \in \mathbb{R}^{D \times d}$ where $d_i < d < D, d_i$ being the dimension of the effective subspace. Then using the random embedding technique, we optimize $g(y) = f(Ay), y \in \mathbb{R}^d$ with CMA-ES, which has reduced complexity compared to optimizing $f$. The Pseudocode is shown below.

This algorithm and the CMA-ES-PCA (see 2) used for comparison purposes were implemented using the Python package *cmaes* Nomura and Shibata (2024). This package provides a simple implementation of the CMA-ES algorithm and tts API provides the ask-tell interface, which facilitates the implementation of random embedding for CMA-ES and its PCA counterpart.

The CMA-ES-PCA algorithm was chosen for its ability to efficiently handle high-dimensional optimization problems by reducing the dimensionality of

**Algorithm 1** CMA-ES with Random Embedding

**Require:** $D$: Original high-dimensional space dimension
**Require:** $d$: Reduced lower-dimensional space dimension
**Require:** $f$: Objective function in $\mathbb{R}^D$
**Ensure:** Optimized solution in $\mathbb{R}^D$

**Initialize:**
$A \leftarrow$ Generate random matrix $A \in \mathbb{R}^{D\times d}$ with entries i.i.d. $N(0, 1)$
$\delta \leftarrow$ size of search box
$\mathcal{B} \leftarrow$ set search domain to $[-\delta, \delta]^d$
$\mu \leftarrow$ initial mean vector in $\mathbb{R}^d$
$\sigma \leftarrow$ initial standard deviation of covariance matrix
$sigma \leftarrow$ initial standard deviation of covariance matrix
**Define reduced objective function:**
$g(y) = f(Ay)$ for $y \in \mathbb{R}^d$
**Optimization loop:**
**for** $t = 1$ to $populationSize$ **do**
  /* *Evaluate individuals from population* */
  Ask for an individual
  Evaluate the individual
**end for**
**Map the best solution back to $\mathbb{R}^D$:**
$x^* \leftarrow Ay^*$ where $y^*$ is the best solution in $\mathbb{R}^d$
**return** optimized solution $x^*$ in $\mathbb{R}^D$

the search space through Principal Component Analysis (PCA). Its deterministic nature makes it a suitable candidate for comparison with the random embedding technique.

# Part III

# Experiments

# Chapter 4

# Experimental setup

## Replication of numerical experiment

We begin our experimental setup by partially replicating the numerical experiment conducted by Cartis and Otemissov (2022). The primary objective of this replication is to evaluate and confirm the impact of ambient dimensionality $D$ on the performance of the random embedding approach when combined with the CMA-ES.

In this experiment, we fix the difference between the true intrinsic dimensionality $d$ and the effective dimensionality $d_e$ to values from the set $\{0, 5, 10, 20\}$. While holding $d - d_e$ fixed, we vary the ambient dimension D across the set $\{100, 150, 200, 250\}$. This allows us to systematically investigate how changes in ambient dimensionality influence the optimization performance of the random embedding technique.

It is important to note that two important aspects of using the random embedding technique were discussed in the section on theoretical framework but were not to test directly: the selection of the dimension for random embedding and the choice of the box constraint. Both of these parameters were taken from other works. In the case of box constraints $[-\delta, \delta]^d$ is likely that it is not the smallest possible chosen constraint, meaning some computation time is wasted unnecessarily due to larger search space. Regarding the case of selecting dimension $d$ we used the knowledge directly from the setup to observe its behavior with respect to $d_e$ be made iteratively Sanyang and Kabán (2016), spending part of the evaluation budget starting with very low $d$ and increasing it when the convergence stops, or some prior knowledge in case of many real-world problems or techniques to learn the effect space can be used Cartis and Otemissov (2022).

And lastly, in this experimental setup, the selection of the population size was omitted. While this simplification allowed us to focus on the primary objective of comparing the performance of CMA-ES in multiple scenarios of changing embedding dimension, it is important to acknowledge that population size can significantly affect the optimization process. If chosen too large the evaluation budget might be wasted. Future experiments could consider systematically varying the population size to fully understand its impact on the optimization performance and to identify the optimal settings for each algorithm. In our case, we let CMA-ES choose the size, it is initialized this way $pop_s ize = 4 + \lfloor 3 \rfloor \cdot \log(d)$, where $d$ is the dimension given to CMA-ES, in which the optimization is performed.

### Performance Evaluation Against Pure CMA-ES

Following the initial replication experiment, we conduct a comparative evaluation to assess the performance of the random embedding approach with CMA-ES against the baseline performance of pure CMA-ES. The goal here is to determine the effectiveness and potential benefits of incorporating random embeddings into the CMA-ES framework. By contrasting the performance metrics of the two approaches we see potential to improve the optimization process.

### Carry over to other EAs

In this experimental setup, we aim to investigate if random embedding as a dimensionality-reducing technique can be used with other Evolutionary Algorithms besides CMA-ES. The chosen algorithm is Differential Evolution (DE).

### Restarting CMA-ES

Restarting by generating a new random matrix $A$ the starting parameters might be viable way to improve the performance of the proposed combination Random Embedding + CMA-ES even further, first we try to restart at fixed intervals, then in the second scenario the restart occurs when the convergence stops (as indicated by the optimizer itself).

### Comparison with CMA-ES-PCA

In the final stage of our experimental setup, we extend our analysis by comparing the random embedding approach with a deterministic alternative, specifically the PCA-based dimensionality reduction method, integrated into CMA-ES (CMA-ES-PCA) as shown in Mei and H. Wang (2021). This comparison aims to highlight the differences between stochastic and deterministic dimensionality reduction techniques. By evaluating their respective performances, we seek to understand how the inherent randomness in the embedding affects the optimization process compared to a more structured and deterministic approach provided by PCA. The pseudocode for this technique that is going to be used for testing purposes is shown in Algorithm 2.

**Algorithm 2** CMA-ES with PCA

**Require:** Learning rates:$\alpha_\mu$, $\alpha_\sigma$, $\alpha_{cp}$, $\alpha_{c1}$, $\alpha_{c\lambda}$
**Require:** Generation Count: $t = 0$
**Require:** Attenuation Factor: $d_\sigma$
**Require:** Evolutionary Paths $p_\sigma^{(0)} = 0$, $p_c^{(0)} = 0$
**Require:** Default Covariance Matrix: $C^{(0)} = I$
**Ensure:** $\mu(t), \sigma(t), C(t)$

```
while not stopping criteria do
    if iteration t is a multiple of 20 then
        Calculate PCA matrix P using the last generation's population
    end if
    Transform C^(t) to C^(t)_ΛΘ using PCA matrix P
    Sample x_i^(t+1) = μ^(t) + σ^(t) y_i^(t+1) where y_i ~ N(0, C^(t)_Θ)
    Re-map x_i^(t+1) = P^T x_i^(t+1)
    Evaluate fitness of each candidate x_i^(t+1), i = 1, ..., λ
    Select top λ samples based on fitness
    Update μ, σ, C using standard CMA-ES updates
    t ← t + 1
end while
return μ(t), σ(t), C(t)
```

## 4.1 Evaluation Methodology

For evaluation purposes, the tested objective function $f$ which has effective dimension $d_e$ is, as proposed in Cartis and Otemissov (2022) and Z. Wang et al. (2016), first lifted to an arbitrarily high dimension $D$ by adding $D - d_e$ dimensions with zero weights, so we obtain new function $\tilde{g}$ 4.1. Then to obtain a function with a non-trivial constant subspace, the function has to be rotated by applying an orthogonal matrix $\mathbb{R}^{D\times D}$ to $x$. The final function is then as shown in 4.2. Since the random embedding is invariant to the ordering of parameters (Letham et al. (2020)) the test functions are defined in such a way that they have the first $d_e$ parameters as relevant and disregard the remaining $D - d_e$ parameters as they do not affect function evaluation.

$$\tilde{g}(x) = f(x_1, x_2, \cdots, x_{d_e}) + 0 \cdot x_{d_e+1} + 0 \cdot x_{d_e+2} + \cdots + 0 \cdot x_D \tag{4.1}$$

$$g(x) = \tilde{g}(\mathbb{R}x) \tag{4.2}$$

In our experiments, the evaluation budget is crucial due to the high cost of black-box function evaluations. To ensure reliable results, we perform ten repeated runs for each experiment, reducing the impact of randomness.

Performance is measured using the fitness gap, which indicates the difference between the best solution and the known global optimum. A smaller fitness gap signifies better optimization performance. We track the number of function evaluations and the fitness gap to evaluate and compare the convergence of different algorithms under the given evaluation budget.

## 4.2 Test functions

The functions are taken from (Surjanovic and Bingham (2023)). Functions 4.3 and 4.2 were used with the modification to increase the dimensionality of the function as described in Oh, Gavves, and Welling (2019), functions 4.7 and 4.6 did not need such a change as they can scale themselves. Additionally, increased dimensionality, some functions were also modified so they don't have a trivial solution.

**Repeated Branin function.**

$$f(x_1, x_2) = \left(x_2 - \frac{5.1}{4\pi^2}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10 \tag{4.3}$$

$$f_{rep}(\mathbf{x}) = \lfloor\frac{D}{2}\rfloor \sum_{i=1}^{\lfloor D/2\rfloor} f(x_{2i-1}, x_{2i}) \tag{4.4}$$

**Repeated Hartmann6 function.**

$$f(\mathbf{x}) = -\sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{6} A_{ij}(x_j - P_{ij})^2\right), \tag{4.5}$$

where,

$$\mathbf{A} = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix},$$

$$\mathbf{P} = 10^{-4} \times \begin{pmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{pmatrix},$$

and

$$\boldsymbol{\alpha} = \begin{pmatrix} 1.0 & 1.2 & 3.0 & 3.2 \end{pmatrix}.$$

**Sphere function.**

$$f(x) = \sum_{i=1}^{D}(x_i - 1)^2 \tag{4.6}$$

**Rosenbrock function.**

$$f(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \tag{4.7}$$

## 4.3 Results

### 4.3.1 Invariance to the ambient dimensionality

We replicated the numerical experiments conducted by fixing the $d - d_e$ and testing the performance of the random embedding in various ambient dimensions $D$, functions used in the experiment were Replicated Branin 4.3 and 4.6. Experiments showed that the optimization performance of the random embedding approach is indeed invariant to the increase in ambient dimensionality, as long as $d \leq d_e$. Our experiments successfully replicated the findings in the literature, confirming the invariance of change in constant subspace and setting the $d$ to a value slightly above the effective dimensionality is better. This ensures that the optimization process remains efficient and effective, regardless of the dimensionality of the original problem space. This also serves as a basic validation that the implementation of the random embedding technique is at least partially correct. Results for Sphere function are shown in Figure 4.1.

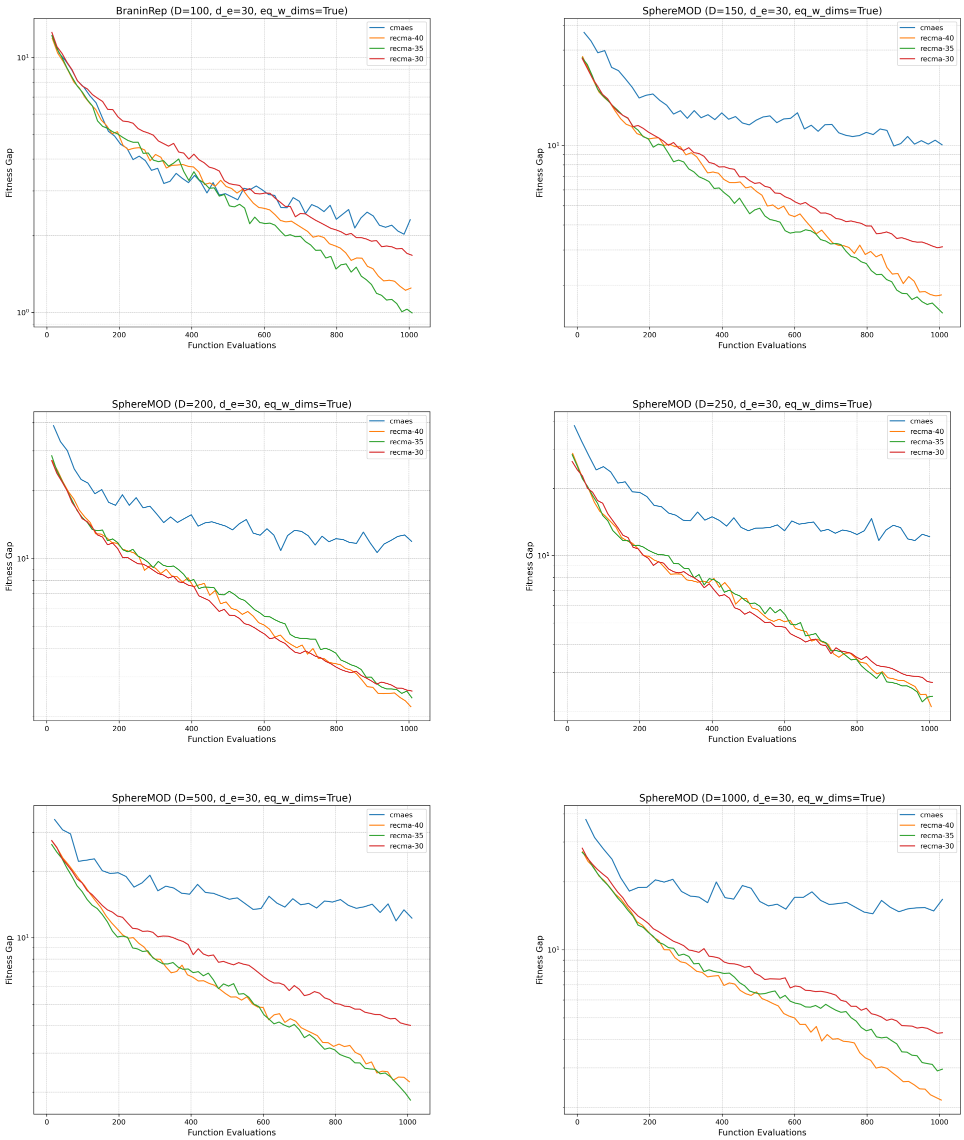


**Figure 4.1:** Experiment with Sphere function4.6 with fixed $d = d_e + c, c = \{0, 5, 10\}$ and with ambient dimension $D \in \{100, 150, 200, 250, 500, 1000\}$.

## 4.4 Comparison with pure CMA-ES

The results of our experiments indicate that the random embedding CMA-ES significantly outperformed the pure CMA-ES across all tested functions. This performance boost is attributed to the reduced complexity and improved search efficiency in the lower-dimensional subspace. The embedding to $d = 100$ serves as a sanity check, to make sure that it behaves similarly to the pure CMA-ES. Instead of a random matrix $\mathbf{A}^{D\cdot\ d}$, identity matrix $\mathbf{I}^{D\cdot\ D}$ was used to “reduce” the dimensionality of the problem to $d = 100$, so nothing should change. The results are shown in Figure 4.2.

### 4.4.1 Sphere function

The first tested function was Sphere. The results show that the random embedding works well in reducing the dimensionality of the problem space, leading to a improvement in the optimization process. The results 4.2 are mostly as expected, with the $d = 30$ converging to the optimum the fastest, it is as was already shown in Cartis and Otemissov (2022), showing that higher $d$ than the effective dimensionality $d_e$ improves the performance, but there is drop and adding more dimensions starts to degrade the performance. Conversely, if the dimension $d$ is too small it does not converge at all. It is however unexpected that the $d = 20$ after 5000 function evaluations start to converge faster.

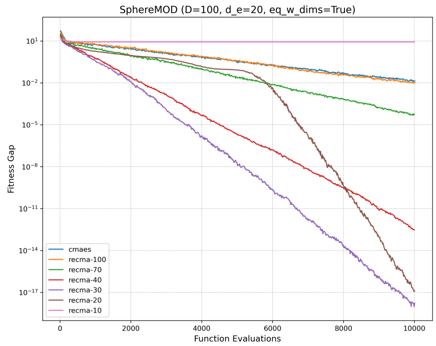


**Figure 4.2:** SphereMOD function with $D = 100$ and effective dimension $d_e = 20$, CMA-ES enhanced by random embedding with $d \in \{10, 20, 30, 40, 70, 100\}$ using pure CMA-ES.

### 4.4.2 Repeated Branin function

The situation here is almost identical to the Sphere function, the results we obtained are within our expectation.

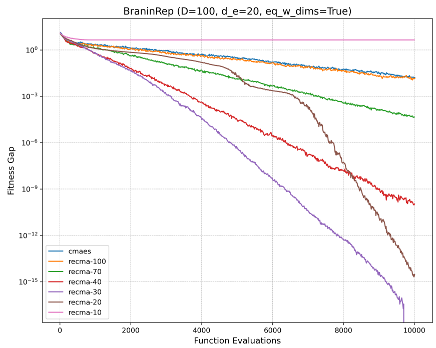


**Figure 4.3:** Branin function with $D = 100$ and $d_i = 20$ using pure CMA-ES.

### 4.4.3 Rosenbrock function

The most noticeable difference in this testing case is how lower the dimension $d$ is, faster it starts to converge, but then it stops very early on. After about 4000 evaluations it starts to behave as observed in previous cases.

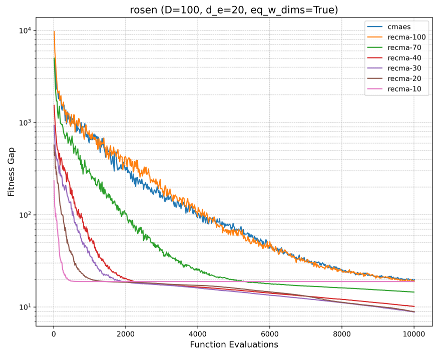


**Figure 4.4:** Rosenbrock function with $D = 100$ and effective dimension $d_e = 20$ using pure CMA-ES.

### 4.4.4 Repeated Hartmann6 function

The early results indicated similar behavior to Rosenbrock function, so we rerun the experiment with a higher evaluation budget 50000. The results (see 4.5) is interesting as it shows that the setting $d = d_e = 20$ works best. Otherwise for $d > d_e$ the performance is as in previous cases.

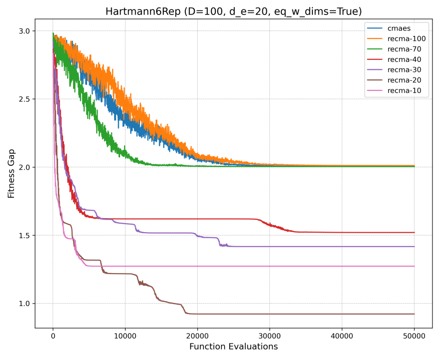


**Figure 4.5:** Hartmann6 function with $D = 100$ and effective dimension $d_e = 20$, CMA-ES enhanced by random embedding with $d \in \{10, 20, 30, 40, 70, 100\}$ using pure CMA-ES.

## 4.5 Random Embedding with other EA

The results of the experiments with Differential Evolution indicate that the random embedding is also effective when used with other evolutionary algorithms. The tests were again conducted on the same functions as in the previous experiment. The fitness gap is generally larger than in the case of CMA-ES but the ranking of how each embedding dimension affects the performance is generally the same showing the possibility of using random embedding as a versatile way to improve the performance of optimization algorithms. Existing EA can be easily modified to accommodate using the random embedding technique.

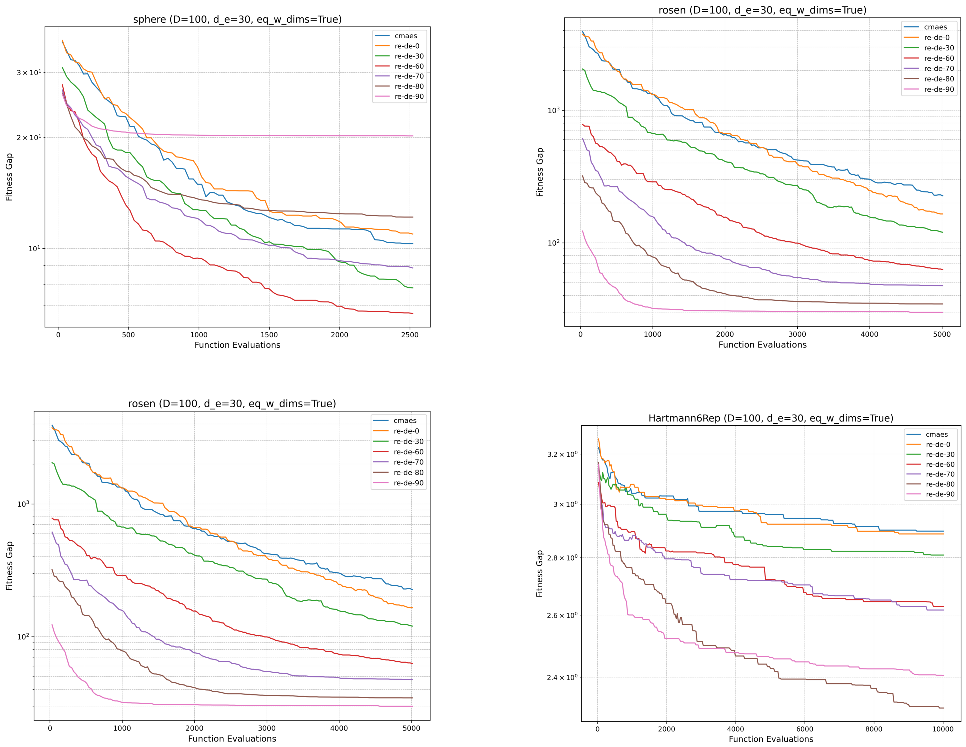


**Figure 4.6:** Experiment with Differential Evolution with Sphere, Rosenbrock, BraninRep and Hartmann6Rep functions.

## 4.6 Restarting CMA-ES

### 4.6.1 Fixed interval restart

Restarting the embedding matrix at fixed intervals (measured in function evaluations) does not improve the convergence in the case of Sphere and Repeated Branin function. On the other hand, in the case of the Rosenbrock function the situation is different, convergence starts slow but improves over time and doesn't stop as early as in the case of the normal version of random embedding. In the case of repeated Hartmann situation is similar as it converges faster. It it interesting to see that setting the dimension here to $d = 100$ is not same as the no embedding version of CMA-ES. So while there is no dimensional reduction, the restarts apparently help with finding the optimum.

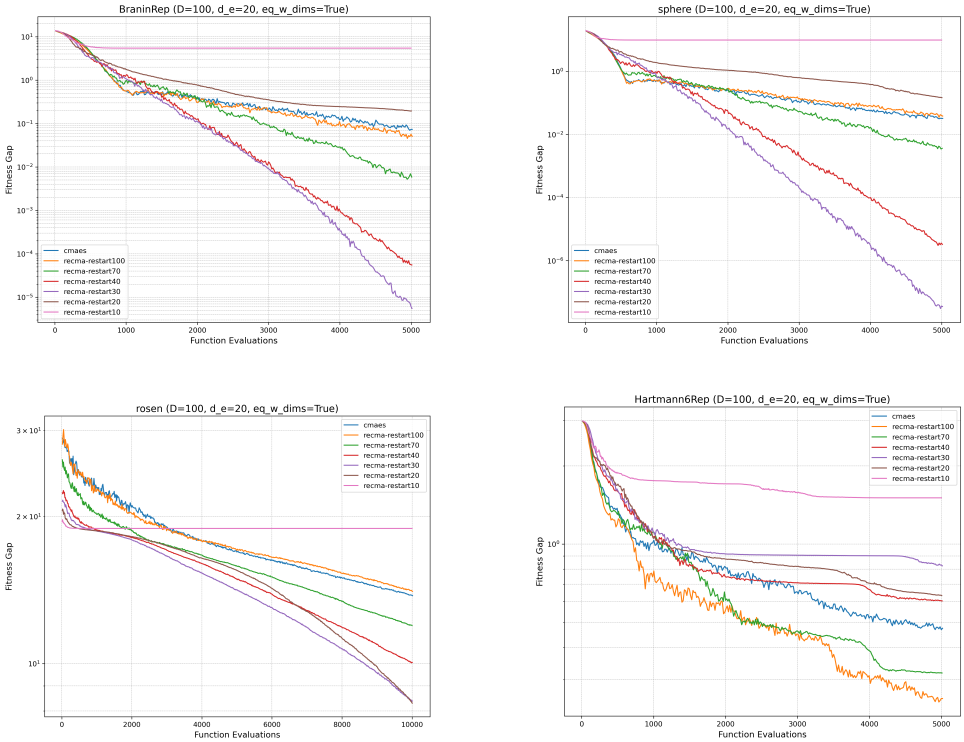


**Figure 4.7:** Fixed-interval restart experiment for all functions from the test set. Embedding dimensions tested are kept same for comparison purpoes

### 4.6.2 Convergence-based restart

The situation here is almost identical as in the fixed interval restart, the observation about the convergence of Hartmann6 is greatly improved for restarted CMA-ES.

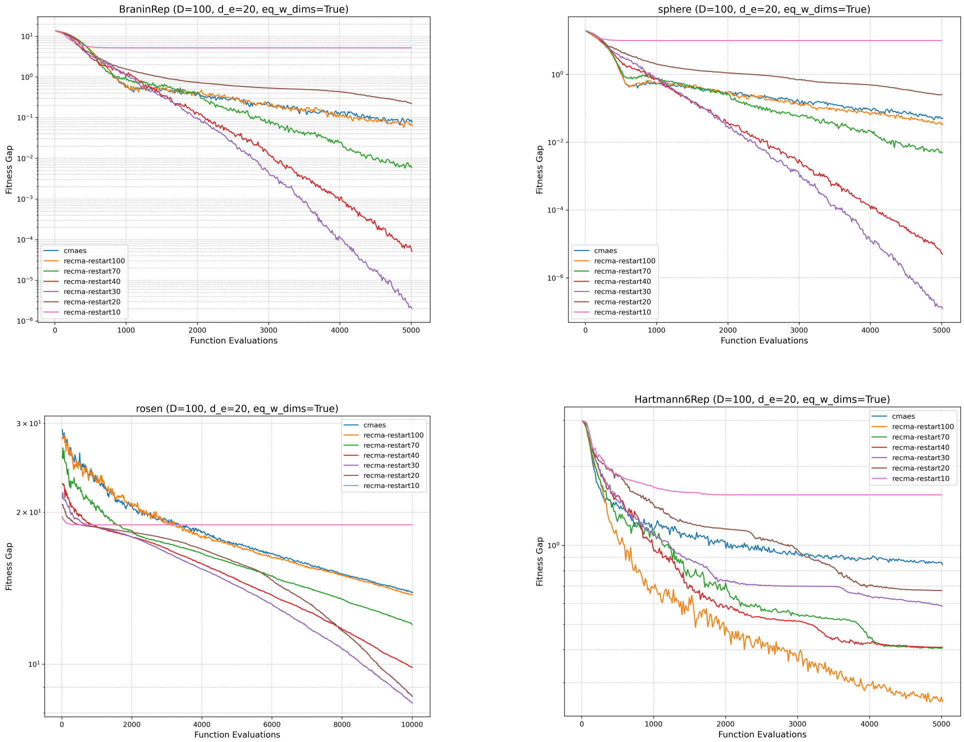


**Figure 4.8:** Convergence-based restart experiment for all functions from the test set. Embedding dimensions tested are kept same for comparison purpoes

## 4.7 Comparision with CMA-ES-PCA

As part of the experimental setup, an attempt was made to compare the performance of the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) enhanced with Principal Component Analysis (PCA) against our optimization technique. The objective was to evaluate the performance of incorporating PCA into CMA-ES to reduce the dimensionality of the optimization problem, thereby potentially improving performance in high-dimensional spaces.

Ensuring that the solutions sampled in the reduced-dimensional space could be accurately transformed back to the original space for evaluation by the objective function proved to be complex. The transformation between the original high-dimensional space and the reduced-dimensional space introduced inconsistencies in the solution vectors. This experiment was thus not carried out and subsequently, no result was obtained, to confirm the effectiveness of using PCA against the Random Embedding.

# Chapter 5

# Conclusion

The experimental results for combining CMA-ES with random embedding are promising when applied to synthetic benchmark functions. These initial findings suggest that random embedding can significantly enhance the performance of optimization algorithms, particularly in high-dimensional settings.

We first numerically demonstrated the invariance to ambient dimensionality. Then we showed that the random embedding technique can effectively reduce the search to help an Evolutionary Algorithm converge faster, be it the in-depth studied CMA-ES or other EA such as DE. Restarting CMA-ES was also tested, but the results are inconclusive. It might be suited for some types of functions but that would require further investigation. These are the main contributions of our work that have not been experimented upon in earlier research. The shown optimization experiments have the potential to impact many machine learning topics where hyperparameter optimization is costly for big data applications.

To gain further insights and validate these results more comprehensively, it is robust to adapt and test the proposed methods within the COCO framework using the BBOB test suite for large-scale optimization. This can be done in future research. The BBOB test suite offers a standardized and rigorous evaluation environment, allowing for a detailed comparison of optimization algorithms across a wide range of challenging problem instances.

However, executing such extensive evaluations using the BBOB test suite in the COCO framework would require substantial computational resources. Running these tests would span multiple days on a high-performance computing (HPC) system. In my case, due to constraints including, but not limited to, the prohibitive cost of using HPC resources, such extensive testing was not feasible. Another way to approach testing would be to use real-world test cases such as hyperparameters optimization as shown in Letham et al. (2020).

# Bibliography

Cartis, Coralia and Adilet Otemissov (Mar. 2022). "A dimensionality reduction technique for unconstrained global optimization of functions with low effective dimensionality". en. In: *Information and Inference: A Journal of the IMA* 11.1, pp. 167–201. ISSN: 2049-8772. DOI: 10.1093/imaiai/iaab011. URL: https://academic.oup.com/imaiai/article/11/1/167/6278168 (visited on 01/14/2023).

Hansen, Nikolaus (Apr. 4, 2016). *The CMA Evolution Strategy: A Tutorial*. arXiv: 1604.00772 [cs, stat]. URL: http://arxiv.org/abs/1604.00772 (visited on 01/18/2023).

Hou, Yaqing et al. (Dec. 2023). *Multiform Evolution for High-Dimensional Problems with Low Effective Dimensionality*. arXiv: 2401.00168 [cs]. (Visited on 05/20/2024).

James Bergstra and Yoshua Bengio (n.d.). "Random Search for Hyper-Parameter Optimization". In: *The Journal of Machine Learning Research* 13 (), pp. 281–305.

Jin, Jin, Chuan Yang, and Yi Zhang (2020). "An Improved CMA-ES for Solving Large Scale Optimization Problem". en. In: *Advances in Swarm Intelligence*. Ed. by Ying Tan, Yuhui Shi, and Milan Tuba. Vol. 12145. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 386–396. ISBN: 978-3-030-53955-9. DOI: 10.1007/978-3-030-53956-6_34. URL: http://link.springer.com/10.1007/978-3-030-53956-6_34 (visited on 01/14/2023).

Kaban, Ata, Jakramate Bootkrajang, and Robert John Durrant (July 6, 2013). "Towards Large Scale Continuous EDA: A Random Matrix Theory Perspective". In: *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*. GECCO '13: Genetic and Evolutionary Computation Conference. Amsterdam The Netherlands: ACM, pp. 383–390. ISBN: 978-1-4503-1963-8. DOI: 10.1145/2463372.2463423. URL: https://dl.acm.org/doi/10.1145/2463372.2463423 (visited on 01/19/2023).

Letham, Benjamin et al. (2020). "Re-Examining Linear Embeddings for High-Dimensional Bayesian Optimization". In: *A*. Publisher: arXiv Version

Number: 2. DOI: 10.48550/ARXIV.2001.11659. URL: https://arxiv.org/abs/2001.11659 (visited on 01/14/2023).

Liu, Yang et al. (Feb. 27, 2024). *Datasets for Large Language Models: A Comprehensive Survey*. arXiv: 2402.18041 [cs]. URL: http://arxiv.org/abs/2402.18041 (visited on 05/21/2024). preprint.

Mei, Yangjie and Hao Wang (May 2021). *Covariance Matrix Adaptation Evolution Strategy Assisted by Principal Component Analysis*. arXiv: 2105.03687 [cs]. (Visited on 05/23/2024).

Nayebi, Amin, Alexander Munteanu, and Matthias Poloczek (June 2019). "A Framework for Bayesian Optimization in Embedded Subspaces". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research, pp. 4752–4761. URL: https://proceedings.mlr.press/v97/nayebi19a.html.

Nomura, Masahiro and Masashi Shibata (Feb. 2024). *Cmaes : A Simple yet Practical Python Library for CMA-ES*. arXiv: 2402.01373 [cs]. (Visited on 05/24/2024).

Oh, ChangYong, Efstratios Gavves, and Max Welling (Oct. 2019). *BOCK : Bayesian Optimization with Cylindrical Kernels*. arXiv: 1806.01619 [cs, stat]. (Visited on 10/24/2023).

Omidvar, Mohammad Nabi, Xiaodong Li, and Xin Yao (Oct. 2022a). "A Review of Population-Based Metaheuristics for Large-Scale Black-Box Global Optimization—Part I". In: *IEEE Transactions on Evolutionary Computation* 26.5, pp. 802–822. ISSN: 1089-778X, 1089-778X, 1941-0026. DOI: 10.1109/TEVC.2021.3130838. (Visited on 05/21/2024).

– (Oct. 2022b). "A Review of Population-Based Metaheuristics for Large-Scale Black-Box Global Optimization—Part II". In: *IEEE Transactions on Evolutionary Computation* 26.5, pp. 823–843. ISSN: 1089-778X, 1089-778X, 1941-0026. DOI: 10.1109/TEVC.2021.3130835. (Visited on 05/21/2024).

Ros, Raymond and Nikolaus Hansen (2008). "A Simple Modification in CMA-ES Achieving Linear Time and Space Complexity". In: *Parallel Problem Solving from Nature – PPSN X*. Ed. by Günter Rudolph et al. Vol. 5199. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 296–305. ISBN: 978-3-540-87699-1. DOI: 10.1007/978-3-540-87700-4_30. URL: http://link.springer.com/10.1007/978-3-540-87700-4_30 (visited on 01/14/2023).

Sanyang, Momodou L. and Ata Kabán (2016). "REMEDA: Random Embedding EDA for Optimising Functions with Intrinsic Dimension". In: *Parallel Problem Solving from Nature – PPSN XIV*. Ed. by Julia Handl et al. Vol. 9921. Series Title: Lecture Notes in Computer Science. Cham, pp. 859–868. ISBN: 978-3-319-45822-9. DOI: 10.1007/978-3-319-45823-6_80. URL: http://link.springer.com/10.1007/978-3-319-45823-6_80 (visited on 01/14/2023).

Surjanovic, S. and D. Bingham (2023). *Virtual Library of Simulation Experiments: Test Functions and Datasets*. Retrieved January 18, 2023, from http://www.sfu.ca/~ssurjano.

Tong, Xin, Bo Yuan, and Bin Li (Nov. 2019). "Model complex control CMA-ES". en. In: *Swarm and Evolutionary Computation* 50, p. 100558. ISSN: 22106502. DOI: 10.1016/j.swevo.2019.100558. URL: https://linkinghub.elsevier.com/retrieve/pii/S2210650218308253 (visited on 01/14/2023).

Wang, Jianzhong (2012). *Geometric Structure of High-Dimensional Data and Dimensionality Reduction*. Beijing: Higher Education Press. ISBN: 978-7-04-031704-6 978-3-642-27496-1.

Wang, Ziyu et al. (Feb. 2016). "Bayesian Optimization in a Billion Dimensions via Random Embeddings". In: *Journal of Artificial Intelligence Research* 55, pp. 361–387. ISSN: 1076-9757. DOI: 10.1613/jair.4806. URL: https://jair.org/index.php/jair/article/view/10983 (visited on 01/14/2023).

# Chapter 6

# Attachments

With this thesis, we provide the source code of the implementation of the random embedding with CMA-ES. The code is written in Python and uses the *cmaes* package for the CMA-ES implementation. The code is available on GitHub and were developed by Nomura and Shibata (2024).