Derivative-Free Optimization of High-Dimensional Non-Convex Functions by Sequential Random Embeddings*

Hong Qian, Yi-Qi Hu, and **Yang Yu**
National Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210023, China
{qianh,huyq,yuy}@lamda.nju.edu.cn

Abstract

Derivative-free optimization methods are suitable for sophisticated optimization problems, while are hard to scale to high dimensionality (e.g., larger than 1,000). Previously, the random embedding technique has been shown successful for solving high-dimensional problems with low effective dimensions. However, it is unrealistic to assume a low effective dimension in many applications. This paper turns to study high-dimensional problems with low optimal ε-effective dimensions, which allow all dimensions to be effective but many of them only have a small bounded effect. We characterize the properties of random embedding for this kind of problems, and propose the sequential random embeddings (SRE) to reduce the embedding gap while running optimization algorithms in the low-dimensional spaces. We apply SRE to several state-of-the-art derivative-free optimization methods, and conduct experiments on synthetic functions as well as non-convex classification tasks with up to 100,000 variables. Experiment results verify the effectiveness of SRE.

1 Introduction

Solving sophisticated optimization problems plays an important role in artificial intelligence. Let $f\colon \mathbb{R}^D \to \mathbb{R}$ be a function of which we assume that a global minimizer $\boldsymbol{x}^*$ always exists. An optimization problem can be formally written as

$$\boldsymbol{x}^* = \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^D} f(\boldsymbol{x}).$$

We assume that the optimization problems discussed in this paper are *deterministic*, i.e., every call of $f(\boldsymbol{x})$ returns the same value for the same $\boldsymbol{x}$.

In this paper, we focus on derivative-free optimization methods, which regard f as a black-box function that can only be evaluated point-wisely, i.e., they perform optimization based on the function values $f(\boldsymbol{x})$ for the sampled solutions and other information like gradient is not used. Because these methods do not rely on derivatives, they are suitable for optimization problems that are, e.g., with many local optima, non-differentiable, and discontinuous, which are often encountered in a wide range of applications. The performance of a derivative-free optimization algorithm can be evaluated by the *simple regret* [Bubeck *et al.*, 2009], i.e., given n function evaluations, for minimization,

$$r_n = f(\boldsymbol{x}(n)) - \min_{\boldsymbol{x}\in\mathbb{R}^D} f(\boldsymbol{x}),$$

where $\boldsymbol{x}(n) \in \mathbb{R}^D$ is the solution with the lowest function value found by the algorithm when the budget of n function evaluations is used up. The simple regret measures the difference between the best function value found by the algorithm and the minimum of f.

Many derivative-free optimization methods have been designed under various principles. They can be roughly categorized into three kinds: model-based methods, deterministic Lipschitz optimization methods and meta-heuristic search. Model-based methods, such as Bayesian optimization methods [Brochu *et al.*, 2010; Snoek *et al.*, 2012; Kawaguchi *et al.*, 2015] and classification-based methods [Yu *et al.*, 2016], learn a model from the solutions and the model is then applied to guide sampling of solutions for the next round. Deterministic Lipschitz optimization methods need Lipschitz continuity assumption on f, such as [Jones *et al.*, 1993; Pintér, 1996; Bubeck *et al.*, 2011; Munos, 2014]. Meta-heuristic search is designed with inspired heuristics, such as evolutionary strategies [Hansen *et al.*, 2003].

Problem. Almost all derivative-free methods are effective and efficient in low-dimensional problems (usually less than 100 dimensions), but are hard to scale to high dimensionality (say, larger than 1,000 dimensions). This is mainly due to either the low convergence rate in high-dimensional space, thus unbearably many iterations are required; or the per-iteration computational cost is very high in high-dimensional space, thus it is unbearable for finishing a few iterations; or even both of the reasons. The unsatisfactory scalability is one of the main bottlenecks of these methods.

Related Work. Recently, there are some studies focusing on improving the scalability of derivative-free methods. The two major directions are *decomposition* and *embedding*.

Decomposition methods extract sub-problems from the original optimization problem, and by solving the sub-problems the original problem will be solved. In [Kandasamy

*This research was supported by the NSFC (61375061, 61223003), Foundation for the Author of National Excellent Doctoral Dissertation of China (201451), and 2015 Microsoft Research Asia Collaborative Research Program.

et al., 2015], additive functions were considered, i.e., the function value $f(\boldsymbol{x})$ is the sum of several sub-functions with smaller dimensions, and there is no variable overlaps between any two sub-functions. In [Kandasamy *et al.*, 2015], via employing a Bayesian optimization method, it was shown that using the additive structure can effectively accelerate the Bayesian optimization method. In [Friesen and Domingos, 2015], a recursive decomposition method was proposed for approximately locally decomposable problems. These methods, however, rely on the (mostly axis-parallel) decomposability, which may restrict their applications.

Embedding methods assume that in the high-dimensional space, only a small subspace effects the function value. Therefore, optimization only in the effective subspace can save a lot of efforts. In [Carpentier and Munos, 2012], compressed sensing was employed to deal with linear bandit problems with low-dimensional effective subspaces. In [Chen *et al.*, 2012], a variable selection method was proposed to identify the effective axis-parallel subspace. In [Djolonga *et al.*, 2013], a low-rank matrix recovery technique was employed to learn the effective subspace. In [Wang *et al.*, 2013; Qian and Yu, 2016], the random embedding based on the random matrix theory was employed to identify the underlying linear effective subspace. However, real-world problems may not have a clear effective subspace, also it is hard to verify the existence of the effective subspace.

Our Contributions. In this paper, we study high-dimensional problems with low optimal ε-effective dimensions (see Definition 1). In these problems, any (linear transformed) variable is allowed to effect the function value, however, only a small linear subspace has a large impact on the function value, and the orthogonal complement subspace makes only a small bounded effect.

Firstly, we characterize the property of random embedding for this kind of problems. We find that, given optimal ε-effective dimension, single random embedding bears 2ε embedding gap. Note that this embedding gap cannot be compensated by the optimization algorithm.

We then propose the sequential random embeddings (SRE) to overcome the embedding gap. SRE applies the random embedding several times sequentially, and in each subspace, SRE employs an optimization algorithm to reduce the residue of the previous solution. Therefore, SRE can also be viewed as a combination of decomposition and embedding, as each random embedding defines a sub-problem. We also disclose the condition under which SRE could improve the optimization quality for a large class of problems.

In experiments, we apply SRE to several state-of-the-art derivative-free optimization methods, and conduct experiments on synthetic functions as well as classification tasks using the non-convex Ramp loss. Experiment results show that SRE can significantly improve the performance of the optimization methods in high-dimensional problems. Moreover, comparing with previous related studies where testing functions are mostly up to 1,000 variables, the derivative-free methods with SRE are tested for up to 100,000 variables, in real-world data sets.

The consequent sections respectively introduce the optimal ε-effective dimension problems and present the property of random embedding, describe the proposed SRE technique as well as its theoretical property, present the empirical results, and finally conclude the paper.

2 Optimal ε-Effective Dimension and Random Embedding

Optimal ε-Effective Dimension

Effective dimension defined in [Wang *et al.*, 2013] requires the existence of a non-effective linear subspace, which has exactly zero effect on the function value. It is often unrealistic to make such an assumption. We thus make a relaxation to this assumption as the optimal ε-effective dimension in Definition 1.

Note that a function with optimal ε-effective dimension can have no low-dimensional effective subspace according to the definition given in [Wang *et al.*, 2013], i.e., no linear subspace that has exactly zero effect on the function value. Instead, it has a linear subspace that makes at most ε contribution to the function value. Therefore, this kind of problems may still be efficiently tackled when ε is not so large.

DEFINITION 1 (Optimal ε-Effective Dimension)
*For any $\varepsilon > 0$, a function $f\colon \mathbb{R}^D \to \mathbb{R}$ is said to have an **ε-effective subspace** $\mathcal{V}_\varepsilon$, if there exists a linear subspace $\mathcal{V}_\varepsilon \subseteq \mathbb{R}^D$ s.t. for all $\boldsymbol{x} \in \mathbb{R}^D$, we have $|f(\boldsymbol{x}) - f(\boldsymbol{x}_\varepsilon)| \leq \varepsilon$, where $\boldsymbol{x}_\varepsilon \in \mathcal{V}_\varepsilon$ is the orthogonal projection of $\boldsymbol{x}$ onto $\mathcal{V}_\varepsilon$. Let $\mathbb{V}_\varepsilon$ denote the collection of all the ε-effective subspaces of f, and $dim(\mathcal{V})$ denote the dimension of a linear subspace $\mathcal{V}$. We define the **optimal ε-effective dimension** of f as $d_\varepsilon = \min_{\mathcal{V}_\varepsilon \in \mathbb{V}_\varepsilon} dim(\mathcal{V}_\varepsilon)$.*

In the definition above, it should be noted that ε and d_ε are related variables, commonly, a small d_ε implies a large ε, while a small ε implies a large d_ε.

Random Embedding

Given the definition of optimal ε-effective dimension, Lemma 1 below shows the effect of random embedding for such functions. For simplicity, let $\mathcal{N}$ denote the Gaussian distribution with zero mean and σ^2 variance.

LEMMA 1
Given a function $f\colon \mathbb{R}^D \to \mathbb{R}$ with optimal ε-effective dimension d_ε, and any random matrix $\boldsymbol{A} \in \mathbb{R}^{D\times d}$ ($d \geq d_\varepsilon$) with independent entries sampled from $\mathcal{N}$, then, with probability 1, for any $\boldsymbol{x} \in \mathbb{R}^D$, there exists $\boldsymbol{y} \in \mathbb{R}^d$ s.t. $|f(\boldsymbol{x}) - f(\boldsymbol{A}\boldsymbol{y})| \leq 2\varepsilon$.

Proof. We borrow the idea of constructing such $\boldsymbol{y}$ as in [Wang *et al.*, 2013]. Since f has the optimal ε-effective dimension d_ε, there exists an ε-effective subspace $\mathcal{V}_\varepsilon \subseteq \mathbb{R}^D$ s.t. $\dim(\mathcal{V}_\varepsilon) = d_\varepsilon$. Besides, any $\boldsymbol{x} \in \mathbb{R}^D$ can be decomposed as $\boldsymbol{x} = \boldsymbol{x}_\varepsilon + \boldsymbol{x}_\varepsilon^\perp$, where $\boldsymbol{x}_\varepsilon \in \mathcal{V}_\varepsilon$, $\boldsymbol{x}_\varepsilon^\perp \in \mathcal{V}_\varepsilon^\perp$ and $\mathcal{V}_\varepsilon^\perp$ is the orthogonal complement of $\mathcal{V}_\varepsilon$. By the definition of ε-effective subspace, we have $|f(\boldsymbol{x}) - f(\boldsymbol{x}_\varepsilon)| \leq \varepsilon$. Hence, it suffices to show that, for any $\boldsymbol{x}_\varepsilon \in \mathcal{V}_\varepsilon$, there exists $\boldsymbol{y} \in \mathbb{R}^d$ s.t. $|f(\boldsymbol{x}_\varepsilon) - f(\boldsymbol{A}\boldsymbol{y})| \leq \varepsilon$.

Let $\boldsymbol{\Phi} \in \mathbb{R}^{D\times d_\varepsilon}$ be a matrix whose columns form a standard orthonormal basis for $\mathcal{V}_\varepsilon$. For any $\boldsymbol{x}_\varepsilon \in \mathcal{V}_\varepsilon$, there exists $\boldsymbol{c} \in \mathbb{R}^{d_\varepsilon}$ s.t. $\boldsymbol{x}_\varepsilon = \boldsymbol{\Phi}\boldsymbol{c}$. Let us for now assume that

$\boldsymbol{\Phi}^\top \boldsymbol{A}$ has rank d_ε. If $\text{rank}(\boldsymbol{\Phi}^\top \boldsymbol{A}) = d_\varepsilon$, there must exist $\boldsymbol{y} \in \mathbb{R}^d$ s.t. $(\boldsymbol{\Phi}^\top \boldsymbol{A})\boldsymbol{y} = \boldsymbol{c}$, because $\text{rank}(\boldsymbol{\Phi}^\top \boldsymbol{A}) = \text{rank}([\boldsymbol{\Phi}^\top \boldsymbol{A}, \boldsymbol{c}])$. The orthonormal projection of $\boldsymbol{Ay}$ onto $\mathcal{V}_\varepsilon$ is given by $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top \boldsymbol{Ay} = \boldsymbol{\Phi c} = \boldsymbol{x}_\varepsilon$. Thus, $\boldsymbol{Ay} = \boldsymbol{x}_\varepsilon + \tilde{\boldsymbol{x}}$ where $\tilde{\boldsymbol{x}} \in \mathcal{V}_\varepsilon^\perp$ for $\boldsymbol{x}_\varepsilon$ is the orthonormal projection of $\boldsymbol{Ay}$ onto $\mathcal{V}_\varepsilon$. Since $\boldsymbol{Ay} \in \mathbb{R}^D$, by the definition of ε-effective subspace, we have $|f(\boldsymbol{x}_\varepsilon) - f(\boldsymbol{Ay})| \leq \varepsilon$, and thus $|f(\boldsymbol{x}) - f(\boldsymbol{Ay})| \leq 2\varepsilon$.

Similar to the proof of Theorem 2 in [Wang *et al.*, 2013], we have that $\text{rank}(\boldsymbol{\Phi}^\top \boldsymbol{A}) = d_\varepsilon$ with probability 1. ■

Lemma 1 implies that, given a random embedding matrix $\boldsymbol{A} \in \mathbb{R}^{D\times d}$, for any minimizer $\boldsymbol{x}^* \in \mathbb{R}^D$, there exists $\tilde{\boldsymbol{y}} \in \mathbb{R}^d$ such that $f(\boldsymbol{A}\tilde{\boldsymbol{y}}) - f(\boldsymbol{x}^*) \leq 2\varepsilon$. Note that this *embedding gap* grows twice as fast as ε.

Optimization with Random Embedding

Via random embedding, current derivative-free optimization algorithms can be applied to solve high-dimensional optimization problems by running in a low-dimensional solution space. Given a high-dimensional function f and a random matrix $\boldsymbol{A} \in \mathbb{R}^{D\times d}$, we construct a new optimization function $g(\boldsymbol{y}) = f(\boldsymbol{Ay})$, where the solution space for g only has a dimension d. Note that every solution is evaluated by mapping back to the original high-dimensional space.

For functions with a low optimal ε-effective dimension, we can bound the gap between the optimal function values of g and f based on Lemma 1, which is stated in Theorem 1. We omit the proof since it can be verified directly.

THEOREM 1

Given a function $f\colon \mathbb{R}^D \to \mathbb{R}$ with optimal ε-effective dimension d_ε, and a random matrix $\boldsymbol{A} \in \mathbb{R}^{D\times d}$ ($d \geq d_\varepsilon$) with independent entries sampled from $\mathcal{N}$, let $\boldsymbol{x}^*$ be a global minimizer of f and $g(\boldsymbol{y}) = f(\boldsymbol{Ay})$ where $\boldsymbol{y} \in \mathbb{R}^d$, then, with probability 1*, we have*

$$\inf_{\boldsymbol{y}\in\mathbb{R}^d} g(\boldsymbol{y}) - f(\boldsymbol{x}^*) \leq 2\varepsilon.$$

We then look into the simple regret of a derivative-free optimization algorithm using a random embedding. The optimization algorithm works in the low-dimensional space, and usually, it can only approximate the optimal solution, thus there is a gap between the found solution $\tilde{\boldsymbol{y}}$ and the optimal solution in the low-dimensional space $\inf_{\boldsymbol{y}\in\mathbb{R}^d} g(\boldsymbol{y})$. This *approximation gap* is related to dimension size, function complexity, and optimization budget. We assume that the approximation gap is upper bounded by θ. Furthermore, as disclosed by Theorem 1, there exists an *embedding gap* 2ε, which cannot be compensated by the optimization algorithm. We then have that the simple regret of the algorithm is upper bounded by the approximation gap and the embedding gap,

$$\begin{aligned} g(\tilde{\boldsymbol{y}}) - f(\boldsymbol{x}^*) &= g(\tilde{\boldsymbol{y}}) - \inf_{\boldsymbol{y}\in\mathbb{R}^d} g(\boldsymbol{y}) + \inf_{\boldsymbol{y}\in\mathbb{R}^d} g(\boldsymbol{y}) - f(\boldsymbol{x}^*) \\ &\leq \theta + 2\varepsilon. \end{aligned} \tag{1}$$

3 Sequential Random Embeddings

From Eq.(1), in order to improve the simple regret, i.e., $g(\tilde{\boldsymbol{y}}) - f(\boldsymbol{x}^*)$, we need to reduce the approximation gap and the embedding gap. Unfortunately, this two factors are conflicting: In order to reduce the embedding gap, we need to use a large d_ε to trade for a smaller ε, but with a large d_ε, the optimization algorithm needs to deal with more dimensions, which will badly increase the approximation gap.

Let $\mathcal{S}_i = \{\boldsymbol{A}^{(i)}\boldsymbol{y} \mid \boldsymbol{y} \in \mathbb{R}^d\}$ denote the subspace defined by the random matrix, where $i = 1, \ldots, m$. In spired by [Zhang *et al.*, 2013], we propose the technique of sequentially using multiple random embeddings to reduce the embedding gap while keeping the approximation gap almost unaffected:

- In the first step, it generates a random matrix $\boldsymbol{A}^{(1)}$ which defines a subspace $\mathcal{S}_1$, and applies the chosen optimization algorithm to find a near-optimal solution in the subspace, i.e., $\tilde{\boldsymbol{y}}_1 = \text{argmin}_{\boldsymbol{y}} f(\boldsymbol{A}^{(1)}\boldsymbol{y})$. Let $\tilde{\boldsymbol{x}}_2 = \boldsymbol{A}^{(1)}\tilde{\boldsymbol{y}}_1$ be the high-dimensional solution.
- In the second step, it generates another random matrix $\boldsymbol{A}^{(2)}$ which defines a subspace $\mathcal{S}_2$, and applies the algorithm to optimize the *residue* of the current solution $\tilde{\boldsymbol{x}}_2$ in the subspace, i.e., $\tilde{\boldsymbol{y}}_2 = \text{argmin}_{\boldsymbol{y}} f(\tilde{\boldsymbol{x}}_2 + \boldsymbol{A}^{(2)}\boldsymbol{y})$. Then update the current solution $\tilde{\boldsymbol{x}}_3 = \tilde{\boldsymbol{x}}_2 + \boldsymbol{A}^{(2)}\tilde{\boldsymbol{y}}_2$.
- In the following steps, it acts like the second step that performs the optimization to reduce the residue.

By this simple sequence of applying random embeddings, we show that, for a large class of functions with optimal ε-effective dimension, an upper bound of the solution gap $f(\tilde{\boldsymbol{x}}_i) - f(\boldsymbol{x}^*)$ reduces as the step i increases.

Theoretical Property

Let $\tilde{\boldsymbol{x}}_1 = \boldsymbol{0}$, and $\|\cdot\|$ represent $\|\cdot\|_2$ for simplicity. In the step i of the sequential random embeddings, the residue solution to be approximated is $\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i$, let $\hat{\boldsymbol{x}}_i$ be the orthonormal projection of $\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i$ onto the subspace $\mathcal{S}_i$. Thus, we call $\|\hat{\boldsymbol{x}}_i\|/\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i\|$ as the *embedding ratio* which is smaller than 1, and $\|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\|/\|\hat{\boldsymbol{x}}_i\|$ as the *optimization ratio* for this step. We show in Proposition 1 that, when the optimization ratio is less than a fraction of the embedding ratio, the solution is closer to the optimal solution.

PROPOSITION 1

Given $f \in \mathcal{F}$, and a sequence of random matrices $\{\boldsymbol{A}^{(i)}\}_i \subseteq \mathbb{R}^{D\times d}$ ($d \geq d_\varepsilon$) each with independent entries sampled from $\mathcal{N}$, for all $i = 1, \ldots, m$, if

$$\|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\|/\|\hat{\boldsymbol{x}}_i\| \leq (1/5)\cdot\|\hat{\boldsymbol{x}}_i\|/\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i\|,$$

it holds that $\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i\| > \|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_{i+1}\|$.

Proof. For any $i = 1, \ldots, m$, since $\hat{\boldsymbol{x}}_i$ is the orthonormal projection of $\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i$ onto $\mathcal{S}_i$, we have that

$$\begin{aligned} &\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i\|^2 = \|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_i\|^2 + \|\hat{\boldsymbol{x}}_i\|^2 \\ &\geq (\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\| - \|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\|)^2 + \|\hat{\boldsymbol{x}}_i\|^2 \\ &= \|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_{i+1}\|^2 + \|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\|^2 + \|\hat{\boldsymbol{x}}_i\|^2 \\ &\quad - 2\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\|\cdot\|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\| \\ &\geq \|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_{i+1}\|^2 + (\|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\| - \|\hat{\boldsymbol{x}}_i\|)^2 \\ &\quad - 2(\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i\| + \|\boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\|)\cdot\|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\| \\ &\quad + 2\|\hat{\boldsymbol{x}}_i\|\cdot\|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\| \\ &\geq \|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_{i+1}\|^2 + (\|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\| - \|\hat{\boldsymbol{x}}_i\|)^2 \end{aligned}$$

$$- 2\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i\| \cdot \|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\| - 2\|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\|^2,$$

where the last inequality is by $\|\boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\| - \|\hat{\boldsymbol{x}}_i\| \leq \|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\|$.

Since $\|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\| \cdot \|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i\| / \|\hat{\boldsymbol{x}}_i\|^2 \leq 1/5$ and $\|\hat{\boldsymbol{x}}_i\| \leq \|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i\|$, we have that $(\|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\| - \|\hat{\boldsymbol{x}}_i\|)^2 - 2\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i\| \cdot \|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\| - 2\|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\|^2 > 0$. Therefore, $\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i\| > \|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_{i+1}\|$ for all $i = 1, \ldots, m$. ■

We then consider a very general class of problems, i.e., functions with local Hölder continuity as in Definition 2.

DEFINITION 2 (Local Hölder Continuity)
There exists $L, \alpha > 0$ s.t., for all $\boldsymbol{x} \in \mathbb{R}^D$, $f(\boldsymbol{x}) - f(\boldsymbol{x}^*) \leq L \cdot \|\boldsymbol{x} - \boldsymbol{x}^*\|_2^\alpha$, where $\boldsymbol{x}^*$ is a global minimizer of f.

Intuitively, Definition 2 means that the rate of increase of f around $\boldsymbol{x}^*$ is bounded. Note that a function with local Hölder continuity can have many local optima, or non-differentiable.

For any function with local Hölder continuity, since

$$f(\tilde{\boldsymbol{x}}_i) - f(\boldsymbol{x}^*) \leq L \cdot \|\tilde{\boldsymbol{x}}_i - \boldsymbol{x}^*\|_2^\alpha,$$

Proposition 1 implies that the sequential random embeddings can reduce this upper bound in a mild condition.

Less Greedy

By the method of sequential random embeddings (SRE), each sub-problem in the subspace is solved greedily. However, a perfect solution for one sub-problem may not be good globally, and once an unsatisfied solution is found, it is hard to be corrected by the later solutions because of the greedy process. Therefore, we further introduce a withdraw variable α to the previous solution such that it is possible to eliminate the previous solution if necessary, and the optimization problem in each step becomes

$$\min_{\boldsymbol{y},\alpha} f(\alpha\tilde{\boldsymbol{x}}_i + \boldsymbol{A}^{(i)}\boldsymbol{y}).$$

Since derivative-free optimization methods are employed, which make few requirements on the optimization problems, we can simply let the algorithm optimize α together with $\boldsymbol{y}$.

The full algorithm named SRE is depicted in Algorithm 1. Given a derivative-free optimization algorithm $\mathcal{M}$, in each step of SRE, $\mathcal{M}$ optimizes a low-dimensional function $g_i(\boldsymbol{y})$ as defined in line 4, and returns the best solution as well as the withdraw variable found by $\mathcal{M}$ (line 5). Finally, the best solution throughout the procedure is returned as the output.

4 Experiments

For the practical issues in experiments, we set the random matrix $\boldsymbol{A} \in \mathbb{R}^{D\times d}$ with independent entries sampled from $\mathcal{N}(0, 1/d)$, and we consider the high-dimensional solution space $\mathcal{X} = [-u, u]^D$ instead of $\mathbb{R}^D$ and low-dimensional solution space $\mathcal{Y} = [-l, l]^d$ instead of $\mathbb{R}^d$ with $u, l > 0$ for simplicity. To implement the derivative-free algorithm in $\mathcal{Y}$, note that there may exist $\boldsymbol{y}' \in \mathcal{Y}$ s.t. $\boldsymbol{A}\boldsymbol{y}' \notin \mathcal{X}$ and thus f cannot be evaluated at point $\boldsymbol{A}\boldsymbol{y}'$. To tackle this problem, we employ Euclidean projection, i.e., $\boldsymbol{A}\boldsymbol{y}'$ is projected to $\mathcal{X}$ when it is outside $\mathcal{X}$ by $P_{\mathcal{X}}(\boldsymbol{A}\boldsymbol{y}') = \operatorname{argmin}_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{x} - \boldsymbol{A}\boldsymbol{y}'\|_2$. Let $f(P_{\mathcal{X}}(\boldsymbol{A}\boldsymbol{y}')) + \|P_{\mathcal{X}}(\boldsymbol{A}\boldsymbol{y}') - \boldsymbol{A}\boldsymbol{y}'\|_1$ be the function value of

Algorithm 1 Sequential Random Embeddings (SRE)

Input:
Objective function f;
Derivative-free optimization algorithm $\mathcal{M}$;
Number of function evaluation budget n;
Upper bound of the optimal ε-effective dimension d;
Number of sequential random embeddings m.

Procedure:
1: $\tilde{\boldsymbol{x}}_1 = \boldsymbol{0}$.
2: **for** $i = 1$ to m **do**
3: Sample a random matrix $\boldsymbol{A}^{(i)} \in \mathbb{R}^{D\times d}$ with $\boldsymbol{A}_{i,j} \sim \mathcal{N}$.
4: Apply $\mathcal{M}$ to optimize the low-dimensional function $g_i(\boldsymbol{y}) = f(\alpha\tilde{\boldsymbol{x}}_i + \boldsymbol{A}^{(i)}\boldsymbol{y})$ with $\frac{n}{m}$ function evaluations.
5: Obtain the solution $\tilde{\boldsymbol{y}}_i$ and α_i for $g_i(\boldsymbol{y})$ by $\mathcal{M}$.
6: $\tilde{\boldsymbol{x}}_{i+1} = \alpha_i\tilde{\boldsymbol{x}}_i + \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i$.
7: **end for**
8: **return** $\operatorname{argmin}_{i=2,\ldots,m+1} f(\tilde{\boldsymbol{x}}_i)$.

$\boldsymbol{A}\boldsymbol{y}'$ where $\boldsymbol{A}\boldsymbol{y}' \notin \mathcal{X}$. Let $[\boldsymbol{x}]_i$ denote the i-th coordinate of $\boldsymbol{x}$. Since $\mathcal{X} = [-u, u]^D$, the closed form of Euclidean projection is $[P_{\mathcal{X}}(\boldsymbol{A}\boldsymbol{y})]_i = -u$ if $[\boldsymbol{A}\boldsymbol{y}]_i < -u$; $[P_{\mathcal{X}}(\boldsymbol{A}\boldsymbol{y})]_i = [\boldsymbol{A}\boldsymbol{y}]_i$ if $-u \leq [\boldsymbol{A}\boldsymbol{y}]_i \leq u$; and $[P_{\mathcal{X}}(\boldsymbol{A}\boldsymbol{y})]_i = u$ otherwise.

We employ three state-of-the-art derivative-free optimization methods: IMGPO [Kawaguchi *et al.*, 2015], which is a combination of Bayesian optimization and optimistic optimization, CMAES [Hansen *et al.*, 2003], which is a representative of evolutionary algorithms, and RACOS [Yu *et al.*, 2016], which is a model-based optimization algorithm proposed recently. All implementations of them are by their authors. We also employ random search as a reference baseline.

When we apply the single random embedding with these optimization algorithms, we denote by the prefix "RE-", and when we apply the SRE, we denote by the prefix "SRE-". Thus we have combinations including RE-IMGPO, RE-CMAES, RE-RACOS, SRE-IMGPO, SRE-CMAES, and SRE-RACOS.

On Synthetic Functions

We first test the algorithms on two synthetic testing functions. Based on the convex Sphere function and the highly non-convex Ackley function, we construct the high-dimensional Sphere function and Ackley function to meet the optimal ε-effective dimension assumption.

The high-dimensional Sphere function is constructed as

$$f_1(\boldsymbol{x}) = \sum_{i=1}^{10} ([\boldsymbol{x}]_i - 0.2)^2 + \frac{1}{D}\sum_{i=11}^{D} ([\boldsymbol{x}]_i - 0.2)^2,$$

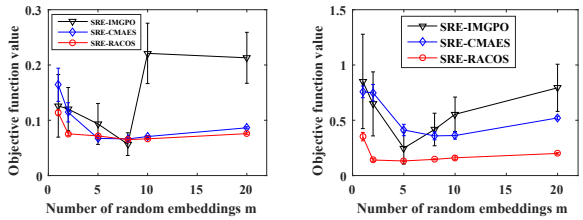
where the dimensions except the first 10 ones have limited impact on the function value. The high-dimensional Ackley function is similarly constructed as

$$\begin{aligned} f_2(\boldsymbol{x}) = &-20\exp\left(-\frac{1}{5}\sqrt{\frac{1}{10}\sum_{i=1}^{10}([\boldsymbol{x}]_i - 0.2)^2}\right) \\ &- \exp\left(\frac{1}{10}\sum_{i=1}^{10}\cos 2\pi([\boldsymbol{x}]_i - 0.2)\right) + \exp(1) + 20 \\ &+ \frac{1}{D}\sum_{i=11}^{D}([\boldsymbol{x}]_i - 0.2)^2. \end{aligned}$$

Noted that the optimal solutions of these functions are $(0.2, \ldots, 0.2)$, in order to avoid the all-zero optimal solution which is in all the linear subspaces. These functions are

minimized within the solution space $\mathcal{X} = [-1, 1]^D$. We set $\mathcal{Y} = [-1, 1]^d, \alpha \in [-1, 1]$. Each algorithm is run 30 times independently, and the average performance is reported.

On the number of random embeddings m. We first study the effect of SRE iteration number m, i.e., the number of sequential random embeddings. We choose $D = 10000$, set the total number of function evaluations $n = 10000$ and the subspace size $d = 10$, and choose the number of sequential random embeddings $m = \{1, 2, 5, 8, 10, 20\}$. When $m = 1$, algorithms with SRE degenerate into algorithms with RE. The achieved objective function values are shown in Figure 1.

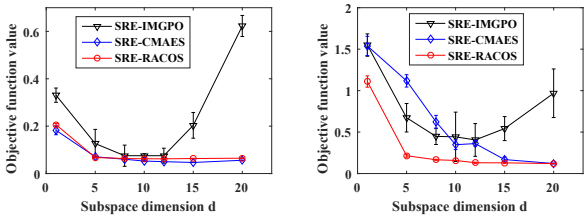


(a) on Sphere function (b) on Ackley function

Figure 1: On the effect of the number of random embeddings m.

Figure 1 (a) and (b) show that, if the total number of function evaluations is fixed, we should choose a compromised value for m. Because if m is large then the budget for each step of SRE is limited, and if m is small then the steps in SRE is limited, and both of them can lead to unsatisfied optimization performance.

On subspace dimension d. To study how low-dimensional size d affects optimization performance, we only adopt the algorithms with SRE (SRE-IMGPO, SRE-CMAES, SRE-RACOS). We choose $D = 10000$, set the total number of function evaluations $n = 10000$, choose $d = \{1, 5, 8, 10, 12, 15, 20\}$, and set the number of sequential random embeddings $m = 5$. The achieved objective function values are shown in Figure 2.

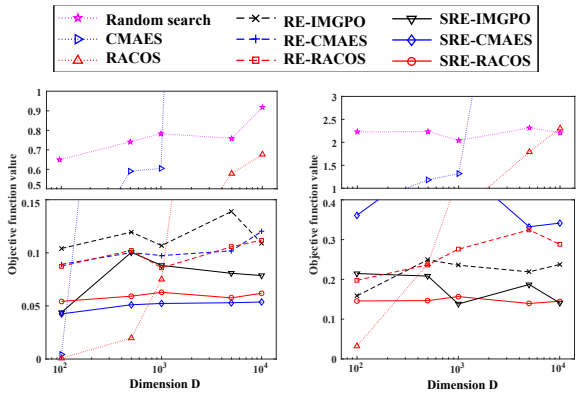


(a) on Sphere function (b) on Ackley function

Figure 2: On the effect of the subspace dimension d.

Figure 2 (a) and (b) show that, for algorithms with SRE, in most cases the closer the d to d_ε the better the optimization performance, indicating that a good estimate of the optimal ε-effective dimension is desirable. Besides, we can observe that, even if $d < d_\varepsilon = 10$ but close to d_ε, the performances of algorithms with SRE are still satisfied.

On scalability. We then study the scalability w.r.t. the solution space dimensions D, we choose $D = \{100, 500, 1000, 5000, 10000\}$, set the total number of function evaluations $n = 10000$ for all algorithms, set $d = 10$ for algorithms with RE and SRE, and set the number of sequential random embeddings $m = 5$ for algorithms with SRE. The achieved objective function values are shown in Figure 3.

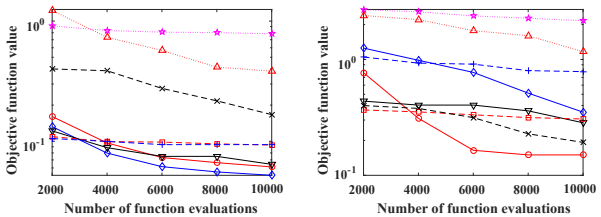


(a) on Sphere function (b) on Ackley function

Figure 3: Comparing the scalability with $n = 10000$ function evaluations.

Figure 3 (a) and (b) show that the algorithms with SRE have the lowest growing rate, while the algorithms without RE have the highest growing rate as the dimension increases, indicating that SRE can scale the derivative-free algorithms to high-dimensional problems better than the compared algorithms.

On convergence rate. To study the convergence rate w.r.t. the number of function evaluations, we set $D = 10000$, and set the total number of function evaluations $n = \{2000, 4000, 6000, 8000, 10000\}$, set low-dimensional size $d = 10$ for RE as well as SRE, and set $m = 5$ for SRE. The achieved objective function values are shown in Figure 4.



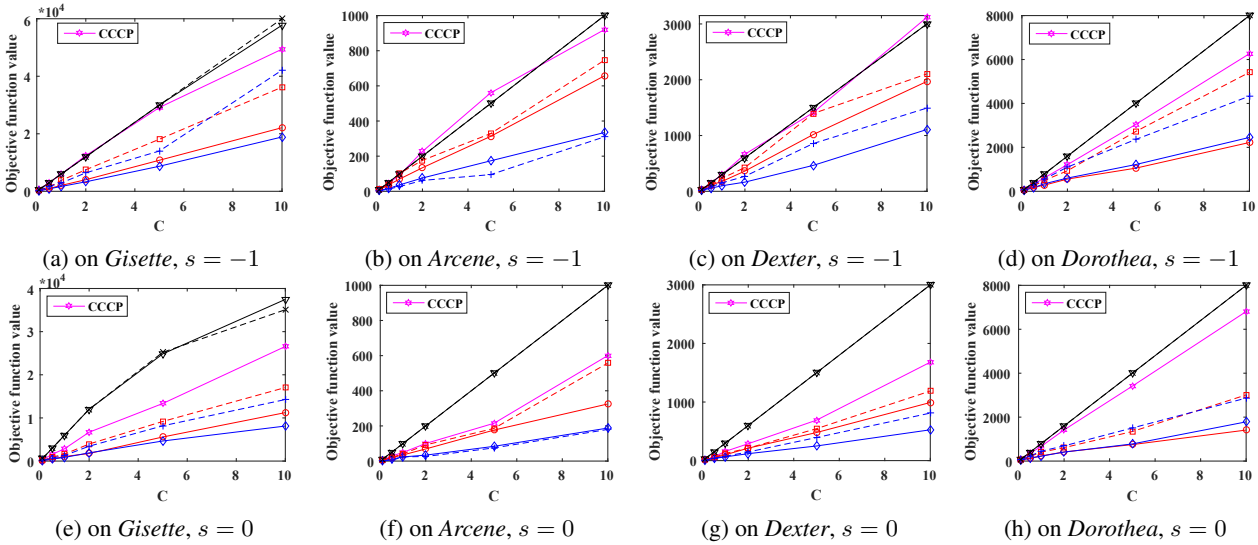
(a) on Sphere function (b) on Ackley function

Figure 4: Comparing the convergence rate with $D = 10000$. The Y-axis is in log-scale. The legend is shared with Figure 3.

Figure 4 (a) and (b) show that, in most cases, algorithms with SRE reduce the objective function value with the highest rate, while the algorithms without RE reduce the objective function value with the lowest rate, indicating that algorithms with SRE converge faster in general than the others.

On Classification with Ramp Loss

We finally study on a classification task with Ramp loss [Collobert *et al.*, 2006]. The Ramp loss is defined as $R_s(z) =$



(a) on *Gisette*, $s = -1$ (b) on *Arcene*, $s = -1$ (c) on *Dexter*, $s = -1$ (d) on *Dorothea*, $s = -1$

(e) on *Gisette*, $s = 0$ (f) on *Arcene*, $s = 0$ (g) on *Dexter*, $s = 0$ (h) on *Dorothea*, $s = 0$

Figure 5: Comparing the achieved objective function values against the parameter C of the classification with Ramp loss. The legend is shared with Figure 3 except CCCP.

$H_1(z) - H_s(z)$ with $s < 1$, where $H_s(z) = \max\{0, s - z\}$ is the Hinge loss with s being the Hinge point. The task is to find a vector $\boldsymbol{w}$ and a scalar b to minimize

$f(\boldsymbol{w}, b) = \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{\ell=1}^{L} R_s\big(y_\ell(\boldsymbol{w}^\top \boldsymbol{v}_\ell + b)\big)$,

where $\boldsymbol{v}_\ell$ is the training instance and $y_\ell \in \{-1, +1\}$ is its label. This objective function is similar to that of support vector machines (SVM) [Vapnik, 2000], but the loss function of SVM is the Hinge loss. Due to the convexity of the Hinge loss, the number of support vectors increases linearly with the number of training instances in SVM, which is undesired with respect to scalability. While this problem can be relieved by employing the Ramp loss [Collobert *et al.*, 2006].

The applied algorithms here are RE-IMGPO, RE-CMAES, RE-RACOS, SRE-IMGPO, SRE-CMAES, SRE-RACOS and the concave-convex procedure (CCCP) [Yuille and Rangarajan, 2001] which is a gradient-based non-convex optimization approach for objective functions that can be decomposed into convex sub-function plus concave sub-function. We employ four binary class UCI datasets [Blake *et al.*, 1998], *Gisette*, *Arcene*, *Dexter* and *Dorothea*. The feature dimension (D) of which are $5 \times 10^3, 10^4, 2 \times 10^4$, and 10^5, respectively. Since there are two hyper-parameters in the optimization formulation, i.e., C and s, to study the effectiveness of the compared algorithms under different hyper-parameters, we test $s \in \{-1, 0\}$ and $C \in \{0.1, 0.5, 1, 2, 5, 10\}$. We set $d = 20$, the number of function evaluations $n = 3D$, and $\mathcal{X} = [-10, 10]^D, \mathcal{Y} = [-10, 10]^d, \alpha \in [-10, 10]$ for all applied algorithms except for CCCP. For CCCP, we set $\mathcal{X} = [-10, 10]^D$ and let CCCP run until it converges. For algorithms with SRE, we set the number of sequential random embeddings $m = 5$. Each randomized algorithm is repeated 30 times independently. Since we focus on the optimization performance, the achieved objective function values on each dataset are reported in Figure 5.

We should note that, in Figure 5, the lines of RE-IMGPO often overlap with those of SRE-IMPGO, and in sub-figure (c), the line of CCCP overlaps with SRE-IMPGO. As shown in Figure 5, except in the dataset *Arcene*, algorithm with SRE has the best performance for any setting of s and C. And in the dataset *Arcene*, algorithm with RE achieves the best performance. This verifies the effectiveness of SRE and RE, and indicates that SRE is more effective than RE. Comparing CCCP to the derivative-free algorithms with SRE, we can observe that the objective function value of CCCP is approximately as twice as that of the best algorithm with SRE. This implies that algorithms with SRE can be significantly better than CCCP with respect to optimization performance.

5 Conclusion

This paper investigates high-dimensional problems in a more general situation compared with previous works, i.e., all variables can be effective to the function value, while many of them only have a small bounded effect. We define this kind of problems as functions with a low optimal ε-effective dimension, and find that, for this kind of problems, single random embedding makes 2ε loss that can not be compensated by the following optimization algorithm. Thus, we propose the sequential random embeddings (SRE) that employ the random embedding several times, and the condition under which SRE can reduce the embedding loss strictly in each step of SRE is disclosed. To study the effectiveness of SRE, we apply SRE to several state-of-the-art derivative-free algorithms. Experiment results on optimization testing functions and classification with the non-convex Ramp loss (where the number of variables are up to 100,000) indicate that SRE can scale these derivative-free methods to high-dimensional problems significantly.

References

[Blake *et al.*, 1998] C. L. Blake, E. Keogh, and C. J. Merz. UCI Repository of machine learning databases.

[http://www.ics.uci.edu/~mlearn/MLRepository.html], 1998.

[Brochu *et al.*, 2010] Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *CoRR*, abs/1012.2599, 2010.

[Bubeck *et al.*, 2009] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *Proceedings of the 20th International Conference on Algorithmic Learning Theory*, pages 23–37, Porto, Portugal, 2009.

[Bubeck *et al.*, 2011] Sébastien Bubeck, Gilles Stoltz, and Jia Yuan Yu. Lipschitz bandits without the lipschitz constant. In *Proceedings of the 22nd International Conference on Algorithmic Learning Theory*, pages 144–158, Espoo, Finland, 2011.

[Carpentier and Munos, 2012] Alexandra Carpentier and Rémi Munos. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, pages 190–198, La Palma, Canary Islands, 2012.

[Chen *et al.*, 2012] Bo Chen, Rui M. Castro, and Andreas Krause. Joint optimization and variable selection of high-dimensional gaussian processes. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1423–1430, Edinburgh, Scotland, 2012.

[Collobert *et al.*, 2006] Ronan Collobert, Fabian Sinz, Jason Weston, and Léon Bottou. Trading convexity for scalability. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 201–208, Pittsburgh, Pennsylvania, 2006.

[Djolonga *et al.*, 2013] Josip Djolonga, Andreas Krause, and Volkan Cevher. High-dimensional gaussian process bandits. In *Advances in Neural Information Processing Systems 26*, pages 1025–1033, Lake Tahoe, Nevada, 2013.

[Friesen and Domingos, 2015] Abram L. Friesen and Pedro M. Domingos. Recursive decomposition for nonconvex optimization. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 253–259, 2015.

[Hansen *et al.*, 2003] Nikolaus Hansen, Sibylle D. Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.

[Jones *et al.*, 1993] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[Kandasamy *et al.*, 2015] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional bayesian optimisation and bandits via additive models. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 295–304, Lille, France, 2015.

[Kawaguchi *et al.*, 2015] Kenji Kawaguchi, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Bayesian optimization with exponential convergence. In *Advances in Neural Information Processing Systems 28*, pages 2791–2799, Montreal, Canada, 2015.

[Munos, 2014] Rémi Munos. From bandits to Monte-Carlo Tree Search: The optimistic principle applied to optimization and planning. *Foundations and Trends in Machine Learning*, 7(1):1–130, 2014.

[Pintér, 1996] János D. Pintér. *Global Optimization in Action, Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications*. Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, Dordrecht, Boston, London, 1996.

[Qian and Yu, 2016] Hong Qian and Yang Yu. Scaling simultaneous optimistic optimization for high-dimensional non-convex functions with low effective dimensions. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, Phoenix, AZ, 2016.

[Snoek *et al.*, 2012] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2960–2968, Lake Tahoe, Nevada, 2012.

[Vapnik, 2000] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 2000.

[Wang *et al.*, 2013] Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, and Nando De Freitas. Bayesian optimization in high dimensions via random embeddings. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 1778–1784, Beijing, China, 2013.

[Yu *et al.*, 2016] Yang Yu, Hong Qian, and Yi-Qi Hu. Derivative-free optimization via classification. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, Phoenix, AZ, 2016.

[Yuille and Rangarajan, 2001] Alan L. Yuille and Anand Rangarajan. The concave-convex procedure (CCCP). In *Advances in Neural Information Processing Systems 14*, pages 1033–1040, Vancouver, British Columbia, Canada, 2001.

[Zhang *et al.*, 2013] Lijun Zhang, Mehrdad Mahdavi, Rong Jin, Tianbao Yang, and Shenghuo Zhu. Recovering the optimal solution by dual random projection. In *Proceedings of the 26th Conference on Learning Theory*, pages 135–157, Princeton, NJ, 2013.