Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Embedded Bandits for Large-Scale Black-Box Optimization

Abdullah Al-Dujaili,[1] **S. Suresh**[1,2]
[1] School of Computer Science and Engineering, NTU, Singapore
[2] ST Engineering - NTU Corporate Lab, Singapore
aldujail001@e.ntu.edu.sg, ssundaram@ntu.edu.sg

Abstract

Random embedding has been applied with empirical success to large-scale black-box optimization problems with low effective dimensions. This paper proposes the EMBEDDEDHUNTER algorithm, which incorporates the technique in a hierarchical stochastic bandit setting, following the *optimism in the face of uncertainty* principle and breaking away from the *multiple-run* framework in which random embedding has been conventionally applied similar to stochastic black-box optimization solvers. Our proposition is motivated by the bounded mean variation in the objective value for a low-dimensional point projected randomly into the decision space of Lipschitz-continuous problems. In essence, the EMBEDDEDHUNTER algorithm expands optimistically a partitioning tree over a low-dimensional—equal to the effective dimension of the problem—search space based on a bounded number of random embeddings of sampled points from the low-dimensional space. In contrast to the probabilistic theoretical guarantees of multiple-run random-embedding algorithms, the finite-time analysis of the proposed algorithm presents a theoretical upper bound on the regret as a function of the algorithm's number of iterations. Furthermore, numerical experiments were conducted to validate its performance. The results show a clear performance gain over recently proposed random embedding methods for large-scale problems, provided the intrinsic dimensionality is low.

Introduction

Problem. This paper is concerned with the *large-scale black-box* optimization problem given a finite number of function evaluations. Mathematically, the problem has the form:

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) \\ &\text{subject to} && \mathbf{x} \in \mathcal{X} \ , \end{aligned} \tag{1}$$

where $f : \mathcal{X} \subseteq \mathbb{R}^n \to \mathbb{R}$ and $n \gg 10^2$. Without loss of generality, it is assumed that $\mathcal{X} = [-1, 1]^n$, and there exists at least one global optimizer $\mathbf{x}^*$ whose objective value is denoted by f^*, i.e., $\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) = f(\mathbf{x}^*) = f^*$. Solving the optimization problem (1) is notoriously difficult as the sole source of information about its objective function f is available through a *black-box* or an *oracle*, which one can query for the value of f at a specific solution (point) $\mathbf{x} \in \mathcal{X}$. High-order information (e.g., derivatives) are unavailable symbolically nor numerically or are tedious to compute compared to zero-order information—i.e., point-wise function evaluations. Thus, the task is to find the (or one) optimal solution $\mathbf{x}^* \in \mathcal{X}$ to (1) or a good approximation using a finite number v of function evaluations, which is commonly referred to as the evaluation budget. The quality of the returned solution $\mathbf{x}(v) \in \mathcal{X}$ after v function evaluations, denoted by f_v^*, is assessed by the *regret*,

$$r(v) = f_v^* - f^* \ . \tag{2}$$

Besides the aforementioned challenging nature of black-box problems, the high dimensionality n of the decision space $\mathcal{X}$ poses another challenge towards finding the global optimum. Despite their witnessed success, the effectiveness of most black-box optimization algorithms is restricted to moderate dimensions (typically, $n < 100$) and they do not scale well to high-dimensional (say, $n \gg 10^2$) problems. As the dimensionality increases, the number of evaluations (sampled points) required to cover $\mathcal{X}$ increases *exponentially*.

Despite the curse of dimensionality, it has been noted that for artificial intelligence (AI) applications, most dimensions of certain classes of the associated optimization problems do not affect the objective function significantly. In other words, such problems have *low effective dimensionality*, e.g., hyper-parameter optimization for neural and deep belief networks (Bergstra and Bengio 2012).

Related Work. The literature on black-box optimization is huge and we only highlight here works that are closely related to the paper's contribution. The bulk of algorithmic work on large-scale black-box optimization has been following one of two approaches: *decomposition* and *embedding*.

Decomposition algorithms break the problem into several subproblems, and solutions for the original problem are recognized in a coordinated manner. In (Kandasamy, Schneider, and Póczos 2015), Bayesian optimization was scaled to high-dimensional problems whose objectives have an additive structure. i.e., the function f is the sum of several sub-functions with smaller dimensions, such that no two sub-functions share one or more variables. On the other hand, Friesen and Domingos (2015) proposed to decompose the function into *approximately locally independent* sub-functions and optimize them separately. Chen et

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

al. (2010) addressed *interdependent* sub-functions and proposed to consider all entries of the decision vector $\mathbf{x}$ independent and discover their relations gradually. In general, decomposition methods employ axis-aligned decomposability, which may limit their applicability.

Embedding algorithms exploit the assumption/empirical observation of *low effective dimensionality*. Chen, Krause, and Castro (2012) presented a variable selection method to discover the effective axis-aligned subspace, while Djolonga, Krause, and Cevher (2013) sought to learn the effective subspace using a low-rank matrix recovery technique. In (Carpentier, Munos, and others 2012), compressed sensing was applied to deal with linear-bandit problems with a high degree of sparsity. Recent works—motivated by the empirical success of random search in leveraging low effective dimensionality without knowing which variables are important (Bergstra and Bengio 2012)—presented *random embedding* techniques based on the random matrix theory (Wang et al. 2013; Kaban, Bootkrajang, and Durrant 2013) and provided probabilistic theoretical guarantees. In (Qian and Yu 2016), the Simultaneous Optimistic Optimization (SOO) algorithm (Munos 2011) was scaled via random embedding. Problems, whose all dimensions are effective but many of them have a small bounded effect, were addressed in (Qian, Hu, and Yu 2016) where the random embedding technique was incorporated in a sequential framework. In general, random embedding methods employ multiple runs to substantiate the probabilistic theoretical performance.

Our Contributions. This paper aims to tackle large-scale black-box optimization (1) based on the random embedding technique. Previous propositions put the technique in a framework of multiple runs—be it parallel (Qian and Yu 2016) or sequential (Qian, Hu, and Yu 2016)—to maximize the performance guarantee. In this paper, we seek to break away from the *multiple-run* framework and follow the *optimism in the face of uncertainty* principle, or so-called *optimistic optimization*. To this end, we incorporate the random embedding technique in a stochastic hierarchical bandit setting and present EMBEDDEDHUNTER: an algorithmic instance of the sought approach. Similar to other optimistic methods, EMBEDDEDHUNTER iteratively expands a partitioning tree over a low-dimensional space $\mathcal{Y}$ based on randomly projecting sampled points to the original high-dimensional space $\mathcal{X}$ once or more times. This approach is motivated by the proof that the mean variation in the objective function f value for a point $\mathbf{y} \in \mathcal{Y}$ projected randomly to f's decision space $\mathcal{X}$ is bounded for objective functions that are Lipschitz-continuous. EMBEDDEDHUNTER's regret (2) is upper bounded in terms of the number of iterations required to expand near-optimal nodes in the (effective) low-dimensional space based on the Lipschitz continuity assumption and that random embedding can preserve local distance.

The rest of the paper is organized as follows. First, a formal motivation is presented, followed by an introduction to EMBEDDEDHUNTER. Then, the algorithm's finite-time performance is studied and complemented by an empirical validation. Towards the end, the paper is concluded.

Optimistic Optimization Meets Random Embeddings

Optimistic methods, i.e., methods that implement the *optimism in the face of uncertainty* principle have proved to be viable for black-box optimization. Such a principle finds its foundations in the machine learning field addressing the exploration-vs.-exploitation dilemma, known as the multi-armed bandit problem. Within the context of function optimization, optimistic approaches formulate the complex problem of optimization (1) over the space $\mathcal{X}$ as a hierarchy of simple bandit problems (Kocsis and Szepesvári 2006) in the form of space-partitioning tree search. At step t, the algorithm optimistically expands a leaf node (partitions the corresponding subspace) that may contain the global optimum. Previous empirical studies have shown that optimistic methods—e.g., SOO (Munos 2011) and NMSO (Al-Dujaili and Suresh 2016)—are not suitable for problems with high dimensionality.

Random embedding has emerged as a practical tool for large-scale optimization with an experimental success and probabilistic theoretical guarantees. It assumes the problem (1) has an implicit low effective dimension d much lower than the explicit (original) dimension n. In essence, for an optimizer $\mathbf{x}^* \in \mathcal{X} = [-1, 1]^n$ and a random matrix $A \in \mathbb{R}^{n \times d}$ whose entries are sampled independently from a normal distribution, there exists a point $\mathbf{y}^* \in \mathcal{Y} = [-d/\eta, d/\eta]^d$ such that its Euclidean random projection to $\mathcal{X}$, $\mathcal{P}_{\mathcal{X}}(A\mathbf{y}^*)$, is $\mathbf{x}^*$ with a probability at least $1 - \eta$ where $\eta \in (0, 1)$. That is to say, $f(\mathcal{P}_{\mathcal{X}}(A\mathbf{y}^*)) = f(\mathbf{x}^*) = f^*$. The Euclidean random projection of the ith coordinate $[\mathbf{y}]_i$ to $[\mathcal{X}]_i$ is defined as follows.

$$[\mathcal{P}_{\mathcal{X}}(A\mathbf{y})]_i = \begin{cases} 1, & \text{if } [\mathbf{y}]_i \geq 1; \\ -1, & \text{if } [\mathbf{y}]_i \leq -1; \\ [A\mathbf{y}]_i & \text{otherwise.} \end{cases} \tag{3}$$

The reader can refer to (Wang et al. 2013; Qian and Yu 2016) for more details and a formal treatment of the above.

It was shown that is possible to scale up optimistic methods via random embedding (Qian and Yu 2016; Qian, Hu, and Yu 2016). Nevertheless, one can observe that it has been applied in a *multiple-run* framework, where (multiple) M random embeddings are applied on the *same* low-dimensional search space $\mathcal{Y}$ in parallel or sequentially to increase the success rate $1 - \eta^M$, ignoring the relationship among the function f values at the multiple projections in $\mathcal{X}$ of a single point $\mathbf{y} \in \mathcal{Y}$. In Theorem 1, we show that a relationship can be established for Lipschitz functions. Prior to that, let us introduce some notation and state the Lipschitz condition formally.

Notation. Let $\mathcal{N}$ denote the Gaussian distribution with zero mean and $1/n$ variance, and $\{A_p\}_p \subseteq \mathbb{R}^{n \times d}$, with $d \ll n$, be a sequence of realization matrices of the random matrix $\mathbf{A}$ whose entries are sampled independently from $\mathcal{N}$. Furthermore, let $g_P(\mathbf{y})$ be a random (stochastic) function such that $g_P(\mathbf{y}) \stackrel{\text{def}}{=} f(\mathcal{P}_{\mathcal{X}}(\mathbf{A}\mathbf{y}))$ and $g_p(\mathbf{y}) = f(\mathcal{P}_{\mathcal{X}}(A_p\mathbf{y}))$ is a realization (deterministic) function, where $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^d$. On the other hand, let us define $\ell(\mathbf{x}_1, \mathbf{x}_2)$ as $L \cdot ||\mathbf{x}_1 - \mathbf{x}_2||$,

where $||\cdot||$ denotes the L_2-norm. The expectation of a random variable X is denoted by $\mathbb{E}[X]$.

Assumption 1. *f is Lipschitz-continuous, i.e., $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$,*

$$|f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq L \cdot ||\mathbf{x}_1 - \mathbf{x}_2|| \ , \tag{4}$$

where $L > 0$ is the Lipschitz constant.

Theorem 1 (Mean absolute difference for $g_P(\mathbf{y})$)**.** *$\forall y \in \mathcal{Y} \subseteq \mathbb{R}^d$, we have $E[|g_p(\mathbf{y}) - g_q(\mathbf{y})|] \leq \sqrt{8} \cdot L \cdot ||\mathbf{y}||$.*

Proof. From Assumption 1, we have

$$\begin{aligned}\mathbb{E}[|g_p(\mathbf{y}) - g_q(\mathbf{y})|] =& \mathbb{E}[|f(\mathcal{P}_{\mathcal{X}}(A_p\mathbf{y})) - f(\mathcal{P}_{\mathcal{X}}(A_q\mathbf{y}))|] \\ \leq& L \cdot \mathbb{E}[||\mathcal{P}_{\mathcal{X}}(A_p\mathbf{y}) - \mathcal{P}_{\mathcal{X}}(A_q\mathbf{y})||] \ .\end{aligned}$$

From the definition of the Euclidean projection (3):

$$\mathbb{E}[|g_p(\mathbf{y}) - g_q(\mathbf{y})|] \leq L \cdot \mathbb{E}[||A_p\mathbf{y} - A_q\mathbf{y}||]$$

Thus, from Cauchy's inequality, we have

$$\begin{aligned}\mathbb{E}[|g_p(\mathbf{y}) - g_q(\mathbf{y})|] \leq& L \cdot ||\mathbf{y}|| \cdot \mathbb{E}[||A_p - A_q||] \\ \leq& L \cdot ||\mathbf{y}|| \cdot \sqrt{\frac{8}{n}} \cdot \sqrt{\max(n, d)} \qquad (5) \\ \leq& \sqrt{8} \cdot L \cdot ||\mathbf{y}|| \ ,\end{aligned}$$

where (5) is derived from (Hansen 1988). □

Theorem 1 says that the variation in the function f values at points in $\mathcal{X}$ projected randomly from the same low-dimensional point $\mathbf{y} \in \mathcal{Y}$ is bounded on the order of the point's norm $||\mathbf{y}||$. Indeed, the d-dimensional zero vector (center of $\mathcal{Y}$) will always give the same function value (zero variation), regardless of the random matrix used. As a result, one is motivated to project a point $\mathbf{y}$ multiple times proportional to its norm in search for the optimal solution $\mathbf{x}^*$. Next, we provide EMBEDDEDHUNTER: a novel scalable optimistic algorithm that exploits the above result.

EMBEDDEDHUNTER

EMBEDDEDHUNTER is a space-partitioning tree-search algorithm that constructs iteratively finer and finer partitions of the (effective) low-dimensional space $\mathcal{Y}$ in a hierarchical fashion looking for the global optimum. The hierarchical partitioning can be represented by a K-ary tree $\mathcal{T}$, where nodes of the same depth h correspond to a partition of K^h subspaces/cells. i.e., the ith node at depth h, denoted by (h, i), corresponds to the subspace/cell $\mathcal{Y}_{h,i}$ such that $\mathcal{Y} = \cup_{0 \leq i < K^h} \mathcal{Y}_{h,i}$. To each node (h, i), a base point $\mathbf{y}_{h,i}$ (center of $\mathcal{Y}_{h,i}$) is assigned at which f is evaluated once or more times. That is to say, for every new evaluation of the node (h, i), $\mathbf{y}_{h,i}$ is randomly projected to f's decision space $\mathcal{X}$ via a random matrix (3) and gets evaluated. To expand/split a node, the corresponding cell is partitioned into K subscells along one of $\mathcal{Y}$'s coordinates, one coordinate at a depth in a sequential manner. Moreover, let the set of leaf nodes of $\mathcal{T}$ be denoted as $\mathcal{L}$. Furthermore, one can denote the algorithm's tree $\mathcal{T}$ at step t by $\mathcal{T}_t$. Towards the detailed aspects of the algorithm, some assumptions are made about the hierarchical partitioning in line with Assumption 1 and Theorem 1, relating variations in function f values using the same and different projection matrices, respectively.

Function values within the same projection. Based on the Johnson-Lindenstrauss Lemma (Achlioptas 2003; Vempala 2004); for a set of m points $\{\mathbf{y}^i\}_{0 \leq i \leq m} \subset \mathcal{Y}$ and their projections $\{\mathbf{x}^i\}_{0 \leq i \leq m} \subset \mathcal{X}$ via the same matrix, we have $\ell(\mathbf{x}^i, \mathbf{x}^j) \leq (1+\epsilon)^{1/2} \cdot \ell(\mathbf{y}^i, \mathbf{y}^j)$, where $\epsilon \in (0, 1/2]$ and $n > 9 \ln m/(\epsilon^2 - \epsilon^3)$—see (Qian and Yu 2016, Lemma 3).

In other words, the random embedding can probably preserve local distance. Thus, from Assumption 1, the difference between the function f values of two points in the low-dimensional space $\mathcal{Y}$—using the same projection matrix—is on the order of their distance in $\mathcal{Y}$. The next assumption exploits the above observation with respect to the optimal cell (node), which is defined as follows.

Definition 1 (Optimal cell)**.** *A cell $\mathcal{Y}_{h,i}$ at depth $h \geq 0$ is optimal if there exists a random matrix A_p whose entries are sampled independently from $\mathcal{N}$ such that $\min_{\mathbf{y} \in \mathcal{Y}_{h,i}} g_p(\mathbf{y}) = f(\mathbf{x}^*)$, where $\mathbf{x}^*$ is a global optimizer of f. We denote such a cell by $\mathcal{Y}_{h,i_p^*}$ and its node by (h, i_p^*).*

Assumption 2 (Bounded intra-variation)**.** *There exists a decreasing sequence δ in $h \geq 0$ such that for one (or more) optimal cell(s) $\mathcal{Y}_{h,i_p^*}$ at depth h, we have*

$$0 \leq \sup_{q, \mathbf{y} \in \mathcal{Y}_{h,i_p^*}} |g_q(\mathbf{y}_{h,i}) - g_q(\mathbf{y})| \leq \delta(h) \ .$$

As the hierachical partitioning is performed coordinate-wise in a sequential manner, let us link the fact that the cell's shapes are not skewed in some dimensions with Assumption 2 through the next assumption.

Assumption 3 (Well-shaped cells)**.** *$\exists\ m > 0$ such that $\forall (h, i) \in \mathcal{T}$, $\mathcal{Y}_{h,i}$ contains an ℓ-ball of radius $m\delta(h)$ centered in $\mathbf{y}_{h,i}$.*

Function values among different projections. Now, we state another assumption about the optimal cell $\mathcal{Y}_{h,i_p^*}$ in line with Theorem 1.

Assumption 4 (Bounded inter-variation)**.** *There exists two non-decreasing sequences λ and τ in $\mathbf{y}$ and h, respectively, such that for any depth $h \geq 0$, for any optimal cell $\mathcal{Y}_{h,i_p^*}$,*

$$0 \leq \sup_{s,t} |g_s(\mathbf{y}_{h,i_p^*}) - g_t(\mathbf{y}_{h,i_p^*})| \leq \lambda(\mathbf{y}_{h,i_p^*}) \ ,$$

and $\sup_{i_p^*} \lambda(\mathbf{y}_{h,i_p^*}) \leq \tau(h)$.

Note that λ being bounded by τ in h is due to the nature of the hierarchical partitioning: the maximum norm of base points at depth h is smaller than or equal to those at depth $h+1$. e.g., at depth $h = 0$, there is a single node $(0, 0)$ whose base point is the d-dimensional zero vector centered in $\mathcal{Y}$. As the tree $\mathcal{T}$ goes deeper, more base points farther away from the center—and hence greater norms—are sampled.

Combining Assumptions 2 and 4 implies that the values of function f, which the optimal node's base point $\mathbf{y}_{h,i_p^*}$ can have, are within $\tau(h) + \delta(h)$ from the global optimum f^*.

One can therefore establish a lower confidence bound (commonly referred to as the b-value) on the f values within a cell. Let $f^*_{h,i}$ be the best function f value achieved among $\mathbf{y}_{h,i}$ evaluations. Then, the b-value for (h,i) can be written as $b_{h,i} \stackrel{\text{def}}{=} f^*_{h,i} - \tau(h) - \delta(h)$. With the knowledge of the sequences τ and δ, we can expand nodes whose b-values lower bound f^*, discarding other nodes and striking an efficient balance in exploration-vs-exploitation based on the lowest $\tau(h) + \delta(h)$ portion of the function space.

However, the knowledge of such sequences (δ, λ, τ) is not available/known in practice. Thus, we follow an optimistic approach and propose to simulate the knowledge of these sequences via two realization aspects of the algorithm. First, the tree $\mathcal{T}$'s nodes are visited based on their depths and their base points' norms: relating the depth-wise and norm-wise visits to the notion of intra-(same projection) and inter-(different projections) exploration-vs.-exploitation dilemmas, respectively. Second, as the tree $\mathcal{T}$ is swept across multiple depths and norms, a node (h,i) is expanded only if its $f^*_{h,i}$ is strictly smaller than those of nodes of higher depths and those of nodes at the same depth but of greater or equal base points' norms.

Besides motivating the norm-wise traversal described above, Theorem 1 implies evaluating f at the nodes' base points multiple times—each with a new random projection—in proportion to their norms as larger improvement over the current f value is probable at points with greater norms. A tree $\mathcal{T}$ with an odd-numbered partition factor K can seamlessly accommodate this observation because the center child node $(h+1, j)$ of a node (h,i) shares the same base point as its parent (h,i). Therefore, one can decide whether to evaluate a newly created center child node based on the number of function evaluations that its base point had in its ancestor nodes in relation to its norm.

In summary, Algorithm 1 describes EMBEDDEDHUNTER. The algorithm takes four parameters: i). the maximum depth h_{max} up to which nodes can be expanded, it can be a function of the evaluation budget or the number of iterations similar to other optimistic methods; ii). the partition factor K, which has to be an odd number; iii). $\eta \in (0,1)$ to specify the bounds of the search space $\mathcal{Y}$ from the random projection theory; and iv). a multiplicative factor M to bound the number of past function evaluations at a base point $\mathbf{y}_{h,i}$ such that it is not greater than $M \cdot ||\mathbf{y}_{h,i}||$, simulating Theorem 1's bound, $\sqrt{8} \cdot L \cdot ||\mathbf{y}_{h,i}||$. Otherwise, no more evaluation is performed and the node retains its parent's best achieved function value (performed at Line 9 of the algorithm). For the sake of readability, the following definitions were used in Algorithm 1.

$$
\begin{aligned}
\mathcal{L}_{t,h} &\stackrel{\text{def}}{=} \{(h,i) \mid 0 \le i < K^h,\ (h,i) \in \mathcal{L}_t\} \\
\Gamma_{h,t} &\stackrel{\text{def}}{=} \{\gamma \in \mathbb{R}^+_0 \mid \exists (h,i) \in \mathcal{L}_{t,h} \text{ such that } \gamma = ||\mathbf{y}_{h,i}||\} \\
\mathcal{L}^j_{t,h} &\stackrel{\text{def}}{=} \{(h,i) \mid (h,i) \in \mathcal{L}_{t,h}, \\
&\qquad ||\mathbf{y}_{h,i}|| = \text{the } j\text{th largest element} \in \Gamma_{h,t}\} \qquad (6)
\end{aligned}
$$

Algorithm 1 The EMBEDDEDHUNTER Algorithm

Input:
stochastic function g_P,
search space $\mathcal{Y} = [-d/\eta, d/\eta]^d$,
evaluation budget v.
Initialization:
$t \leftarrow 1$, $\mathcal{T}_1 = \{(0,0)\}$, Evaluate $g_P(\mathbf{y}_{0,0})$.
1: **while** evaluation budget is not exhausted **do**
2: $\quad \nu_{\min} \leftarrow \infty$
3: $\quad$ **for** $l = 0$ **to** $\min\{\text{depth}(\mathcal{T}_t), h_{\max}\}$ **do**
4: $\quad\quad$ **for** $j = 1$ **to** $|\Gamma_{l,t}|$ **do**
5: $\quad\quad\quad$ Select $(l,o) = \arg\min_{(h,i) \in \mathcal{L}^j_{t,l}} f^*_{h,i}$
6: $\quad\quad\quad$ **if** $f^*_{l,o} < \nu_{\min}$ **then**
7: $\quad\quad\quad\quad \nu_{\min} \leftarrow f^*_{l,o}$
8: $\quad\quad\quad\quad$ Expand (l,o) into its child nodes
9: $\quad\quad\quad\quad$ Evaluate (l,o)'s child nodes by g_P
10: $\quad\quad\quad\quad$ Add (l,o)'s child nodes to $\mathcal{T}_t$
11: $\quad\quad\quad$ **end if**
12: $\quad\quad$ **end for**
13: $\quad\quad \mathcal{T}_{t+1} \leftarrow \mathcal{T}_t$
14: $\quad\quad t \leftarrow t+1$
15: $\quad$ **end for**
16: **end while**
17: **return** $f^*_v = \min_{(h,i) \in \mathcal{T}_t} f^*_{h,i}$

Theoretical Analysis

In this section, we analyze the performance of the EMBEDDEDHUNTER algorithm and upper-bound its regret (2). To derive a bound on the regret, a measure of the quantity of near-optimal points is used, which is closely similar to those in (Bubeck et al. 2009; Al-Dujaili, Suresh, and Sundararajan 2016) and defined after introducing some terminology. For any $\epsilon > 0$, define the set of ϵ-optimal points as $\mathcal{Y}_\epsilon \stackrel{\text{def}}{=} \{\mathbf{y} \in \mathcal{Y} \mid \min_p g_p(\mathbf{y}) \le f^* + \epsilon\}$ and let $g(\mathcal{Y}_\epsilon) \stackrel{\text{def}}{=} \{\min_p g_p(\mathbf{y}) \mid \mathbf{y} \in \mathcal{Y}_\epsilon\} = [f^*, f^* + \epsilon]$. Likewise, denote the set of *ϵ-optimal nodes* at depth h whose base points are in $\mathcal{Y}_\epsilon$ by $\mathcal{I}^\epsilon_h \stackrel{\text{def}}{=} \{(h,i) \in \mathcal{T} \mid 0 \le i < K^h, \mathbf{y}_{h,i} \in \mathcal{Y}_\epsilon\}$. After t iterations, one can denote the depth of the *deepest expanded optimal* node by h^*_t, where one iteration represents *executing the lines 4–14 of Algorithm 1, once*.

Definition 2. *The m-**near-optimality dimension** is the smallest $d_m \ge 0$ such that there exists $C > 0$ such that for any $\epsilon > 0$, the maximum number of disjoint ℓ-balls of radius $m\epsilon$ and center in $\mathcal{Y}_\epsilon$ is less than $C\epsilon^{-d_m}$.*

Let the considered depth after $t-1$ iterations be h and the depth of the deepest expanded optimal node h^*_{t-1} be $h-1$. At iteration t, EMBEDDEDHUNTER would expand at most $|\Gamma_{h,t}|$ nodes. As the (any) optimal node at depth h is in one of the $\{\mathcal{L}^j_{t,h}\}_{1 \le j \le |\Gamma_{h,t}|}$ sets, the optimal node at depth h is not expanded at iteration t if $\nu_{\min} \le f^*_{h,i^*_p}$ or if there exists a node $(h,i) \in \mathcal{L}_{h,t}$ such that $||\mathbf{y}_{h,i}|| \ge ||\mathbf{y}_{h,i^*_p}||$ and $f^*_{h,i} \le f^*_{h,i^*_p}$. The latter condition implies $f^*_{h,i} - f^* \le f^*_{h,i^*_p} - f^*$ and by triangular inequality, we have $f^*_{h,i} - f^* \le |f^*_{h,i^*_p} - g_p(\mathbf{y}_{h,i^*_p})| + |g_p(\mathbf{y}_{h,i^*_p}) - f^*|$. Hence, from Assumption 4 and

Assumption 2, $f^*_{h,i} - f^* \leq \tau(h) + \delta(h)$. Since τ and δ are non-decreasing and decreasing sequences in h, respectively. One can write $\tau(h)$ as a multiple of $\delta(h)$, that is, there exists $m_h \in \mathbb{Z}^+$ such that

$$f^*_{h,i} - f^* \leq \tau(h) + \delta(h) \leq m_h \cdot \delta(h) \ . \tag{7}$$

Put it differently, when $h^*_{t-1} = h - 1$, the base point of any node at depth h, that is later expanded prior to the optimal node at the same depth, is in the near-optimal space $\mathcal{Y}_{m_h\delta(h)}$. Now, if we assume that prior to any iteration at depth h, $\nu_{\min} \geq m_h\delta(h)$, then by Algorithm 1, it takes at most the next $|\mathcal{I}_h^{m_h\delta(h)}|$ iterations at depth h to expand the optimal cell $\mathcal{Y}_{h,i^*_p}$. With this observation at hand, the next question follows naturally: how many iterations at other depths $\in \{0, \ldots, h_{max}\}$ are required—at most—to expand the optimal cell $\mathcal{Y}_{h,i^*_p}$, and with no assumption on $\nu_{\min}$? In the following lemma, we show that the number of iterations required is upper bounded by the number of nodes in supersets of each of $\{\mathcal{I}_h^{m_h\delta(h)}\}_{0\leq h\leq h_{max}}$. The reader can refer to the supplemental material for a pictorial explanation.

Lemma 1. *Let depth $h \in \{0, h_{max}\}$, $\hat{m} = m_{h_{max}}$ and*

$$t_h \stackrel{\text{def}}{=} h_{max}\big(|\mathcal{I}_0^{\hat{m}\delta(0)}| + |\mathcal{I}_1^{\hat{m}\delta(1)}| + \cdots + |\mathcal{I}_h^{\hat{m}\delta(h)}|\big) \ . \tag{8}$$

After $t \geq t_h$ iterations, the depth of the deepest expanded optimal node is at least h, i.e., $h^*_t \geq h$.

Proof. Refer to the supplemental material. □

Lemma 1 quantifies the number of iterations required to expand an optimal node as a function of the number of $\hat{m}\delta(h)$-optimal nodes. Based on Assumption 3, the following lemma upper bounds the cardinality of such nodes.

Lemma 2. *Let depth $h \in \{0, h_{max}\}$, we have $|\mathcal{I}_h^{\hat{m}\delta(h)}| \leq C(\hat{m}\delta(h))^{-\hat{d}}$, where $\hat{d}$ is defined as the $m/\hat{m}$-near-optimality dimension and C the related constant.*

Proof. Refer to the supplemental material. □

With Lemmas 1 and 2 at hand, the finite-time regret (2) of the EMBEDDEDHUNTER algorithm can be linked to the number of iterations as presented in the next theorem.

Theorem 2. *Define $h(t)$ as the smallest $h \geq 0$ such that:*

$$Ch_{max}\sum_{l=0}^{h(t)}(\hat{m}\delta(l))^{-\hat{d}} \geq t \ , \tag{9}$$

where t is the number of iterations. Then EMBEDDEDHUNTER*'s regret is bounded as $r(t) \leq \min\{\tau(h) + \delta(h) \mid h \leq \min(h(t), h_{max} + 1)\}$.*

Proof. Refer to the supplemental material. □

Experiment	Setup
Convergence: performance w.r.t the number of function evaluations v	$v \in \{10, 50, 10^2, 10^3, 10^4, 5 \times 10^4, 10^5\}$
Scalability: performance w.r.t the problem's dimensionality n	$n \in \{10^2, 5 \times 10^2, 10^3, 10^4, 5 \times 10^4, 10^5\}$
Embedding Number: performance w.r.t. the number of times a random matrix is sampled M	$M \in \{1, 2, 5, 8, 10, 20\}$
Effective Dimension: performance w.r.t. the problem's implicit dimensionality d	$d \in \{2, 5, 10, 20, 50, 75\}$
Effective Dimension Knowledge: performance w.r.t the mismatch between the low dimension used and the actual effective dimension	$d = \{2, 5, 8, 25, 75, 250\}$, $\mathcal{Y} = [-d/\eta, d/\eta]^{10}$ for RESOO and EMBEDDEDHUNTER and $[-1, 1]^{10}$ for SRESOO.

Table 1: Experiments setup. Unless specified above, we set $v = 10^4$, $n = 10^4$, $d = 10$, and $M = 5$. The search space $\mathcal{Y}$ for RESOO and EMBEDDEDHUNTER was set to $[-d/\eta, d/\eta]^d$ with $\eta = 0.3$, SRESOO's $\mathcal{Y}$ was set to $[-1, 1]^{d+1}$ as suggested in (Qian and Yu 2016; Qian, Hu, and Yu 2016), respectively. h_{max} was set to the square root of number of function evaluations for each tree. i.e., for RESOO and SRESOO, $h_{max} = \sqrt{v/M}$; for EMBEDDEDHUNTER, $h_{max} = \sqrt{v}$.

Empirical Analysis

In this section, the efficacy of the proposed method is empirically validated on a set of scalable functions from the literature: the Ellipsoid, FletcherPowell, Rosenbrock, and Ackley test functions (Molga and Smutnicki 2005), each of which reflects some challenges in black-box optimization, e.g., modality, separability, and conditioning. The proposed optimistic method is also compared with the scaled SOO optimistic algorithm (Munos 2011) within two recently presented methods in the random-embedding multiple-run framework, viz. the Simultaneous Optimistic Optimization with Random Embedding (RESOO) (Qian and Yu 2016) and Simultaneous Optimistic Optimization with Sequential Random Embedding (SRESOO) (Qian, Hu, and Yu 2016) algorithms. With the aim of fully characterizing the algorithms' performance, five experiments are conducted with respect to (w.r.t) the performance as listed in Table 1.

Experiment Setup. The compared algorithms were implemented in *Python* and the test functions were imported from the *Optproblems Python package* (Wessing 2016). Each algorithm is run 20 times independently per an experiment configuration and the average performance is reported. The experiments were set up as listed in Table 1. The code/data/supplemental materials of this paper will be made available at the project's website:
http://ash-aldujaili.github.io/eh-lsopt.

Results & Discussion. Results from the five experiments on the four test functions are presented in Figure 1. One can easily appreciate EMBEDDEDHUNTER's performance w.r.t the compared algorithms.

Convergence (v). Across all the tested functions, the performance gap between the algorithms grows larger with higher evaluation budget v. With more function evaluations, EMBEDDEDHUNTER is able to further refine its best achieved solution in comparison with the other algorithms.

Scalability (n). As expected, the best solution quality degrades with the problem's explicit dimensionality n. Never-

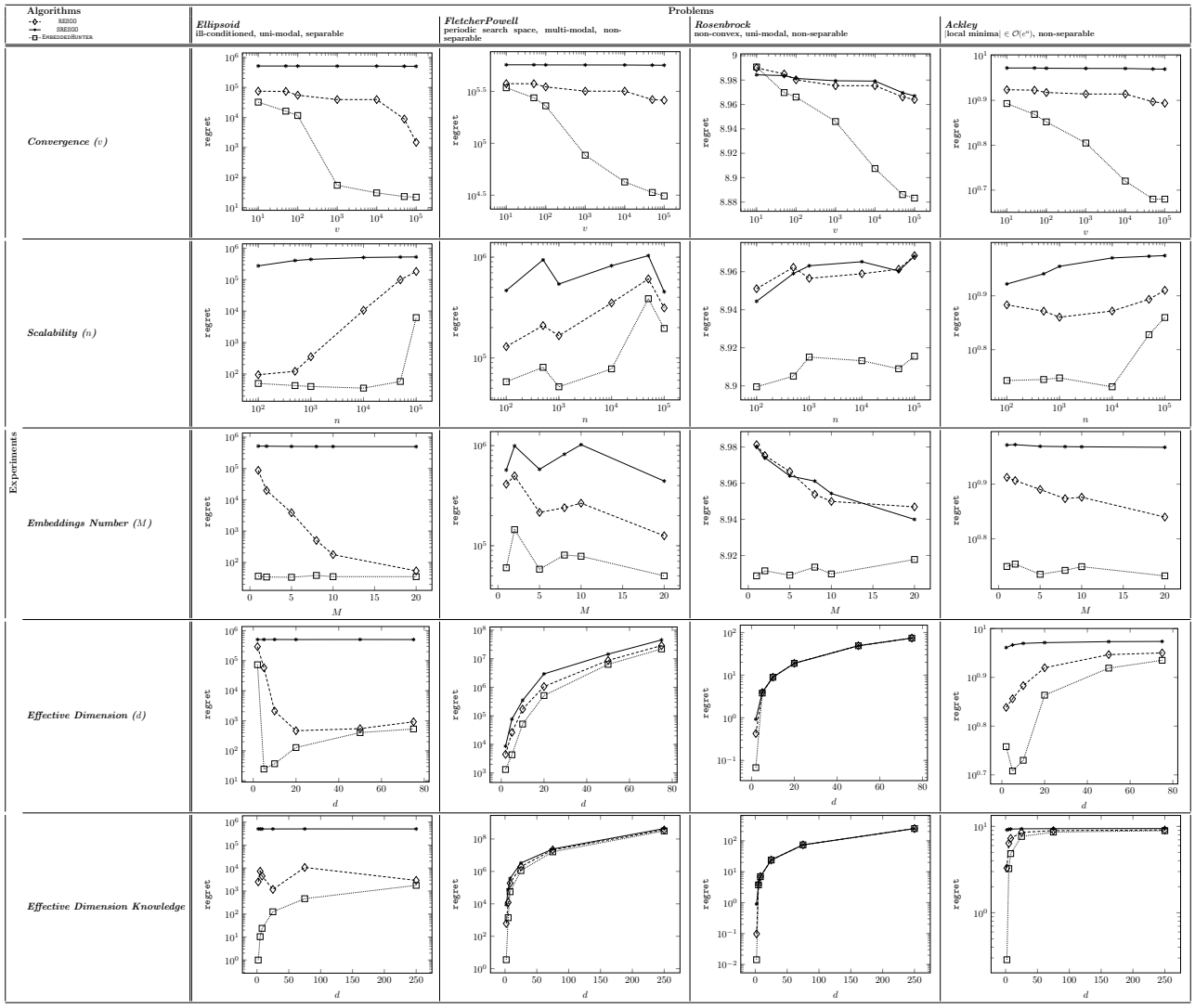

Figure 1: Empirical validation of the EMBEDDEDHUNTER algorithm on a set of commonly-used test optimization problems in comparison with recent large-scale techniques namely RESOO and SRESOO. Each data point of an algorithm's curve represents the mean performance of its 20 runs w.r.t. the experimental configuration considered.

theless, the performance of SRESOO looks robust yet poorer than that of EMBEDDEDHUNTER and RESOO.

Embedding Number (M). Allocating more independent runs seems to be effective for RESOO's and of lesser effect to SRESOO's performance. As M approaches v, both the algorithms act as random search optimizers. This is not the case for EMBEDDEDHUNTER, which uses M as a multiplicative factor to bound the number of evaluations a base point $\mathbf{y}_{h,i}$ can have up to $M \cdot ||\mathbf{y}_{h,i}||$. EMBEDDEDHUNTER uses M as an estimate of Theorem 1's bound factor $\sqrt{8} \cdot L$. Thus, there's a sweet-spot value for each function, which explains the regret's variations for each function in M.

Effective Dimension (d). The results validate the assumption of low effective dimensionality for the suitability of random embedding technique for large-scale problems. It does not scale well with higher effective dimensionality and the algorithms' performance gap reduces w.r.t. the same.

Knowledge of the Effective Dimension. This experiment investigated the performance of low-dimensional embedding irrespective of the effective dimension—be it higher or lower (see Table 1). As the mismatch between the two quantities increases, the performance degrades and the gap among the algorithms reduces.

Conclusion

This paper has presented the EMBEDDEDHUNTER algorithm, a different approach to random embedding for large-scale black-box optimization. While the bulk of random embedding techniques in the literature employ the multiple-run paradigm sampling a new random projection for each run to maximize the probabilistic guarantee of convergence to the optimal solution, EMBEDDEDHUNTER looks for the opti-

mal solution by building stochastic hierarchical bandits (so-called a tree) over a low-dimensional search space $\mathcal{Y}$, where stochasticity has shown to be proportional on average with the norm of the nodes' base points.

The distinctive advantage of EMBEDDEDHUNTER is that its search tree implicitly ranks $\mathcal{Y}$'s regions via its depth/norm-wise visits and allocates the evaluation budget accordingly. Indeed, other algorithms (e.g., RESOO) may evaluate $\mathcal{Y}$'s center–which is a zero vector–M times in its M independent tree searches/projections. This is inefficient as M evaluations are spent generating the same function value, whereas EMBEDDEDHUNTER evaluates the zero vector once and reserves the rest $(M-1)$ evaluations to points with greater norms, exploring more values in the function space.

The finite-time analysis of the algorithm has characterized its performance in terms of the regret as a function of the number of iterations. Besides its theoretically-proven performance, the numerical experiments have validated EMBEDDEDHUNTER's efficacy and robustness with regard to recent random-embedding methods.

Acknowledgments

The research was partially supported by the ST Engineering - NTU Corporate Lab, Singapore, through the NRF corporate lab@universty scheme.

References

Achlioptas, D. 2003. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of Computer and System Sciences* 66(4):671 – 687. Special Issue on PODS 2001.

Al-Dujaili, A., and Suresh, S. 2016. A naive multi-scale search algorithm for global optimization problems. *Information Sciences* 372:294 – 312.

Al-Dujaili, A.; Suresh, S.; and Sundararajan, N. 2016. MSO: a framework for bound-constrained black-box global optimization algorithms. *Journal of Global Optimization* 1–35.

Bergstra, J., and Bengio, Y. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* 13(Feb):281–305.

Bubeck, S.; Stoltz, G.; Szepesvári, C.; and Munos, R. 2009. Online optimization in $\mathcal{X}$-armed bandits. In *Advances in Neural Information Processing Systems*, 201–208.

Carpentier, A.; Munos, R.; et al. 2012. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. In *AISTATS*, 190–198.

Chen, W.; Weise, T.; Yang, Z.; and Tang, K. 2010. Large-scale global optimization using cooperative coevolution with variable interaction learning. In *International Conference on Parallel Problem Solving from Nature*, 300–309. Springer.

Chen, B.; Krause, A.; and Castro, R. M. 2012. Joint optimization and variable selection of high-dimensional gaussian processes. In Langford, J., and Pineau, J., eds., *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 1423–1430. New York, NY, USA: ACM.

Djolonga, J.; Krause, A.; and Cevher, V. 2013. High-dimensional gaussian process bandits. In Burges, C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems 26*. 1025–1033.

Friesen, A. L., and Domingos, P. 2015. Recursive decomposition for nonconvex optimization. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 253–259.

Hansen, P. C. 1988. The 2-norm of random matrices. *Journal of computational and applied mathematics* 23(1):117–120.

Kaban, A.; Bootkrajang, J.; and Durrant, R. J. 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, 383–390. New York, NY, USA: ACM.

Kandasamy, K.; Schneider, J.; and Póczos, B. 2015. High dimensional bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning (ICML)*.

Kocsis, L., and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*. Springer. 282–293.

Molga, M., and Smutnicki, C. 2005. Test functions for optimization needs. http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf. [Retrieved June 2013].

Munos, R. 2011. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In *Advances in Neural Information Processing Systems 24*, 783–791. Curran Associates, Inc.

Qian, H., and Yu, Y. 2016. Scaling simultaneous optimistic optimization for high-dimensional non-convex functions with low effective dimensions. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*.

Qian, H.; Hu, Y.-Q.; and Yu, Y. 2016. Derivative-free optimization of high-dimensional non-convex functions by sequential random embeddings. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*.

Vempala, S. S. 2004. *The Random Projection Method*, volume 65. Providence, Rhode Island: American Mathematical Society.

Wang, Z.; Zoghi, M.; Hutter, F.; Matheson, D.; Freitas, N.; et al. 2013. Bayesian optimization in high dimensions via random embeddings. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, 1778–1784. AAAI Press/International Joint Conferences on Artificial Intelligence.

Wessing, S. 2016. Optproblems: Infrastructure to define optimization problems and some test problems for black-box optimization. [Online; accessed 2016-09-07].