Embedded Bandits for Large-Scale Black-Box Optimization

Abdullah Al-Dujaili, Ph.D.

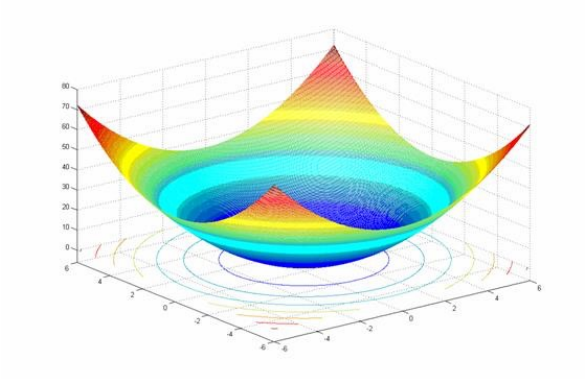
- Al-Dujaili, A., & Suresh, S. "Embedded Bandits for Large-Scale Black-Box Optimization". *AAAI Conference on Artificial Intelligence*. (2017)

Optimization

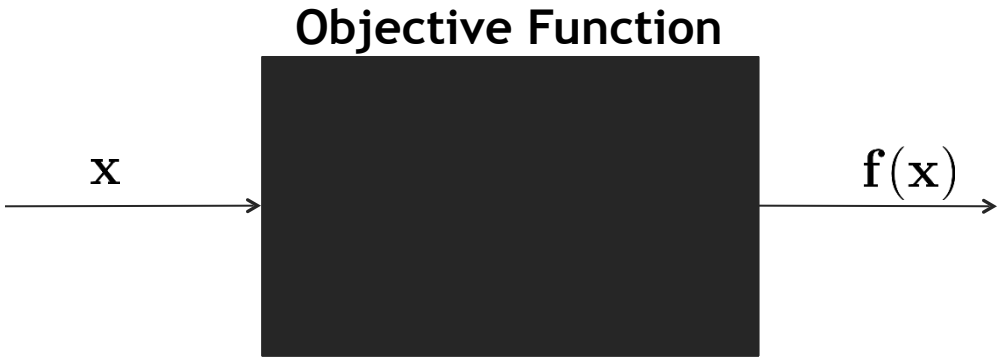
- Recurrent topic of interest for **centuries**

- Many **applications**:
  - Control/planning
  - Machine learning
  - Design/ manufacture

- Many **sub-fields**
  - Convex
  - Discrete
  - Multi-objective



Black-Box Optimization



- Zero-order (value)
- Closed-form
- High-order (gradient)
- Smoothness

- A **search problem** through **point-wise evaluations**.

Large-Scale Black-Box Optimization

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

- $f : \mathcal{X} = [-1, 1]^n \to \mathbb{R}$
- $n \gg 10^2$
- $\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) = f(\mathbf{x}^*) = f^*$

Related Work

1. **Decomposition Techniques**
   - Divide-and-conquer
   - E.g., axis-aligned decomposition into
     - additive functions
     - independent functions

2. **Embedding Techniques**
   - Motivated by empirical observation of **low effective dimensionality.**
   - E.g., Random Embedding (RE) techniques:
     - based on the random matrix theory.
     - probabilistic theoretical guarantees.

Contribution

- RE methods employ multiple runs to substantiate the probabilistic theoretical performance.

- **Motivation:**
  - Break away from the multiple-run framework and
  - Follow the optimism in the face of uncertainty principle (optimistic optimization).

- **Contribution:**
  1. Show that the mean variation in f value for a point $y \in Y$ projected randomly to f's decision space X is bounded for Lipschitz-continuous functions.
  2. **EmbbededHunter**: an algorithm for large-scale black-box optimization.

Notation

- $\mathcal{N}$ denotes the Gaussian distribution with zero mean and $1/n$ variance.
- $\{A_p\}_p \subseteq \mathbb{R}^{n \times d}$, with $d \ll n$, is a sequence of realization matrices of the random matrix $\mathbf{A}$ whose entries are sampled independently from $\mathcal{N}$.
- The Euclidean random projection of the ith coordinate $[\mathbf{y}]_i$ to $[\mathcal{X}]_i$ is defined as follows.

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

- $g_P(\mathbf{y})$ is a random (stochastic) function such that $g_P(\mathbf{y}) = f(\mathcal{P}_{\mathcal{X}}(\mathbf{A}\mathbf{y}))$ and $g_p(\mathbf{y}) = f(\mathcal{P}_{\mathcal{X}}(A_p\mathbf{y}))$ is a realization (deterministic) function, where $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^d$.

Random Embedding: Probabilistic Guarantee

• 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 $\mathcal{N}$, 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 $\eta \in (0, 1)$ (Wang et al., 2013).

Theorem I

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

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

  where $L > 0$ is the Lipschitz constant.

- The mean variation in the objective value for a point y in the low-dimensional space $\mathcal{Y} \subseteq \mathbb{R}^d$ projected randomly into the decision space $\mathcal{X}$ of Lipschitz-continuous problems is *bounded*.

- Mathematically, $\forall y \in \mathcal{Y} \subseteq \mathbb{R}^d$, we have

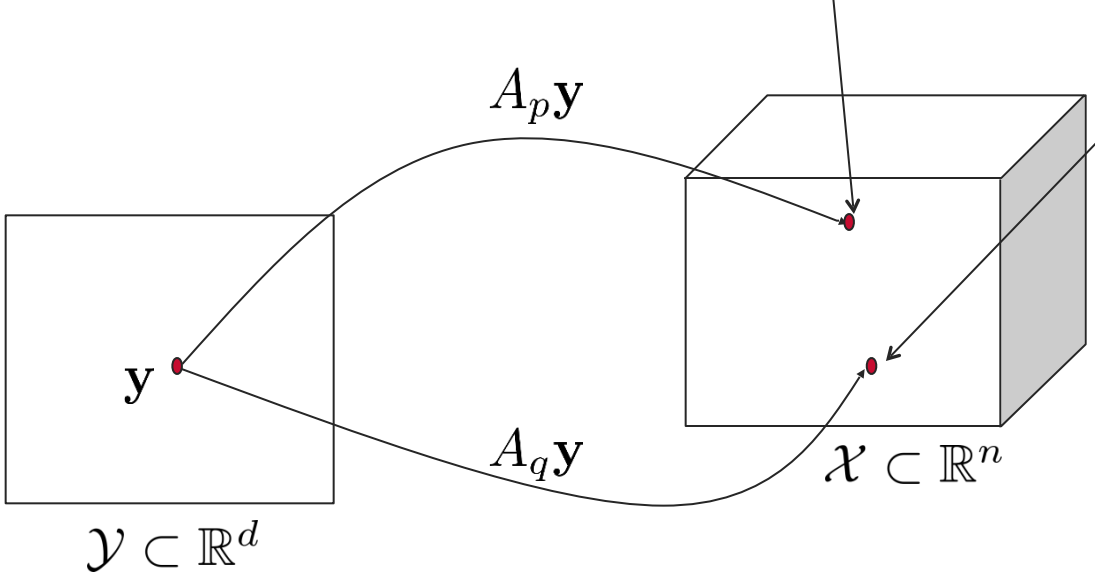
$$E[|g_p(\mathbf{y}) - g_q(\mathbf{y})|] \leq \sqrt{8} \cdot L \cdot ||\mathbf{y}|| \ .$$

Theorem I: Illustrated

$$E[||g_p(\mathbf{y}) - g_q(\mathbf{y})||] \leq \sqrt{8} \cdot L \cdot ||\mathbf{y}||$$

$$g_p(\mathbf{y}) = f(A_p\mathbf{y})$$

$$g_q(\mathbf{y}) = f(A_q\mathbf{y})$$



$$d \ll n$$

Theorem I: Proof

From the Lipschitz assumption, 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:

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

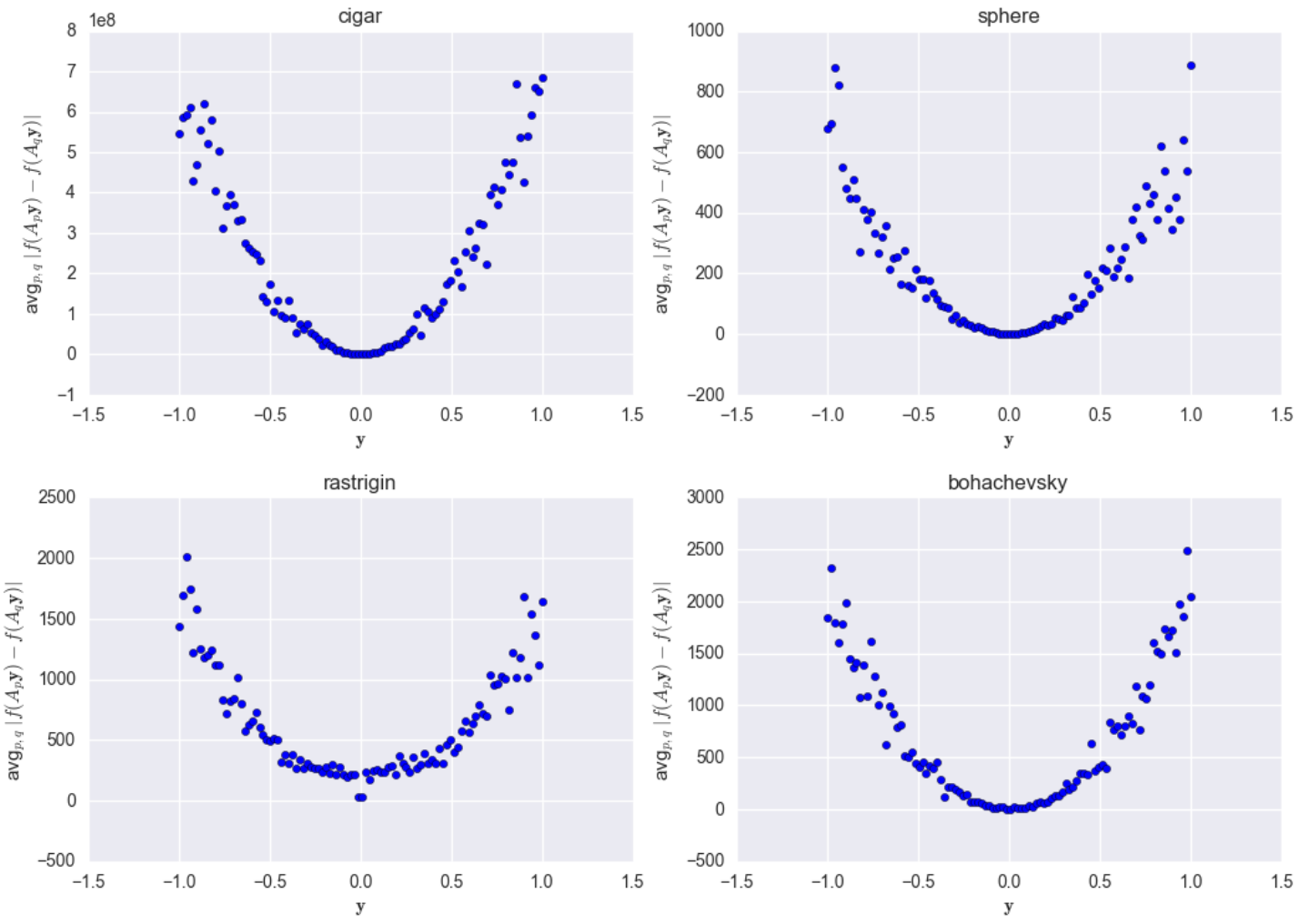
Thus, from Cauchy's inequality, we have

$$\begin{aligned}\mathbb{E}[|g_p(\mathbf{y}) - g_q(\mathbf{y})|] \leq& L \cdot ||\mathbf{y}|| \cdot \mathbb{E}[||A_p - A_q||] \\ \leq& L \cdot ||\mathbf{y}|| \cdot \sqrt{\frac{8}{n}} \cdot \sqrt{\max(n, d)} \qquad (1) \\ \leq& \sqrt{8} \cdot L \cdot ||\mathbf{y}|| \, ,\end{aligned}$$

where (1) is derived from (Hansen, 1988).

Theorem I: Numerical Validation

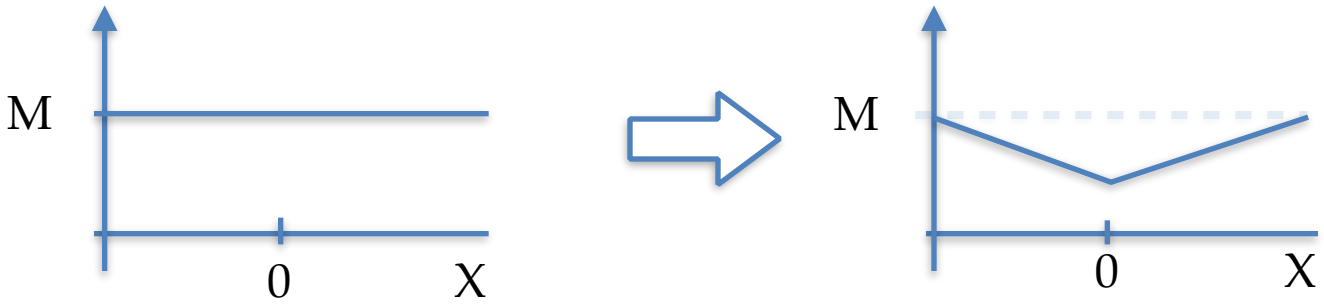
- Empirical mean of the absolute value difference of four functions evaluated at 20 random projections in $\mathbb{R}^n$ of a point $\mathbf{y}$ in $\mathbb{R}^d$ as a function of $\mathbf{y}$.



How can we make use of Theorem I:

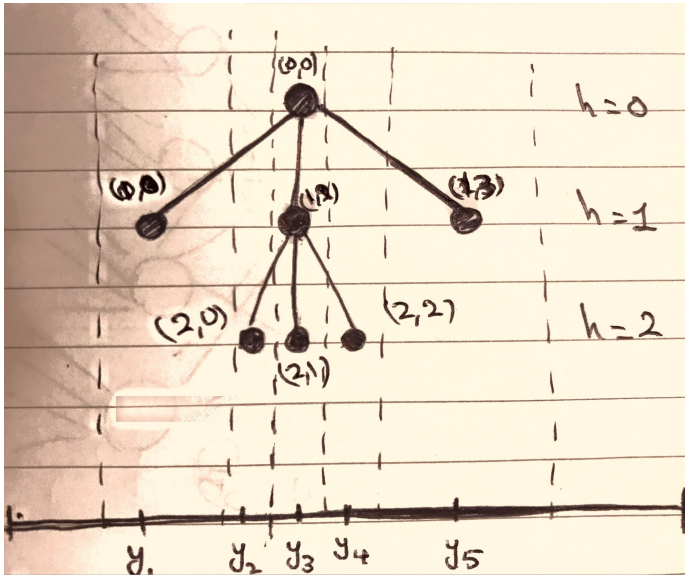
1. Bound the number of random embeddings (M) of a sampled point from the low-dimensional space in proportion to its norm.

2. Favour points with larger norm values.



An Algorithmic Instance: **EmbeddedHunter**

- **EmbeddedHunter** is a $\mathcal{Y}$-partitioning tree-search algorithm.
- The partitioning is represented by a K-ary tree $\mathcal{T}$, where nodes of the same depth h correspond to a partition of K^h subspaces / cells.
- For each node (h, i), f is evaluated at the center point $\mathbf{y}_{h,i}$ of its cell $\mathcal{Y}_{h,i}$ once or more times with different projections based on $||\mathbf{y}_{h,i}||$.



An Algorithmic Instance: **EmbeddedHunter**

- $\mathbf{y}_{0,0} = \mathbf{y}_{1,2} = \mathbf{y}_{2,1} = \mathbf{y}_3$
- For center nodes, decide to sample a new projection based on $||\mathbf{y}_{h,i}||$, say up to $M||\mathbf{y}_{h,i}|| + 1$ projections.
- (2,2) has $\mathbf{y}^*$ for some A^* - $f^* = f(A^*\mathbf{y}^*)$.
- Lipschitz, bounded cell, and the Johnson-Lindenstrauss Lemma (Achlioptas 2003),

$$f(A^*\mathbf{y}_4) - f^* \leq L||A^*\mathbf{y}_4 - A^*\mathbf{y}^*|| \leq \delta(h = 2)$$

- Theorem I:

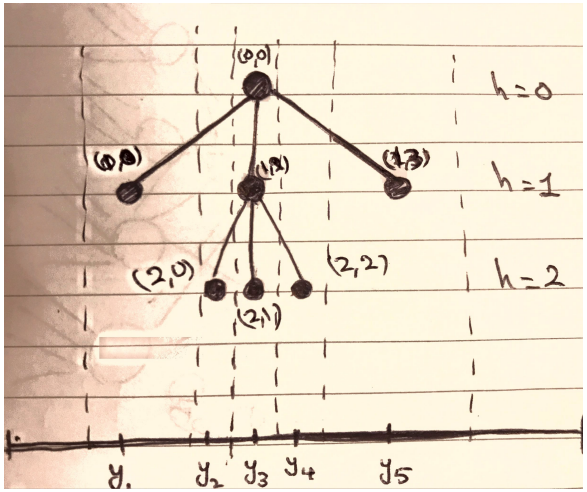
$$f(A_p\mathbf{y}_4) - f(A^*\mathbf{y}_4) \leq O(||\mathbf{y}_4||) \leq \tau(h = 2, i = 2)$$

- We can score each cell (h, i) with:

$$f_{h,i} + \delta(h) + \tau(h, i)$$

- Note that if $f_{2,1} \leq f_{2,2}$,

$$f_{2,1} \leq f^* + \delta(2) + \tau(2, 2)$$



An Algorithmic Instance: **EmbeddedHunter**

- But we do not know neither τ nor δ.
- Act **optimistically**.
- Group and order cells by their **depths** and **center points' norms**
- Expand the (a) cell whose best(smallest) obtained function value is smaller than those of cells of **smaller depths** *or* of the **same depth** but **greater center points' norms**.

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 $g_P(\mathbf{y}_{0,0})$.

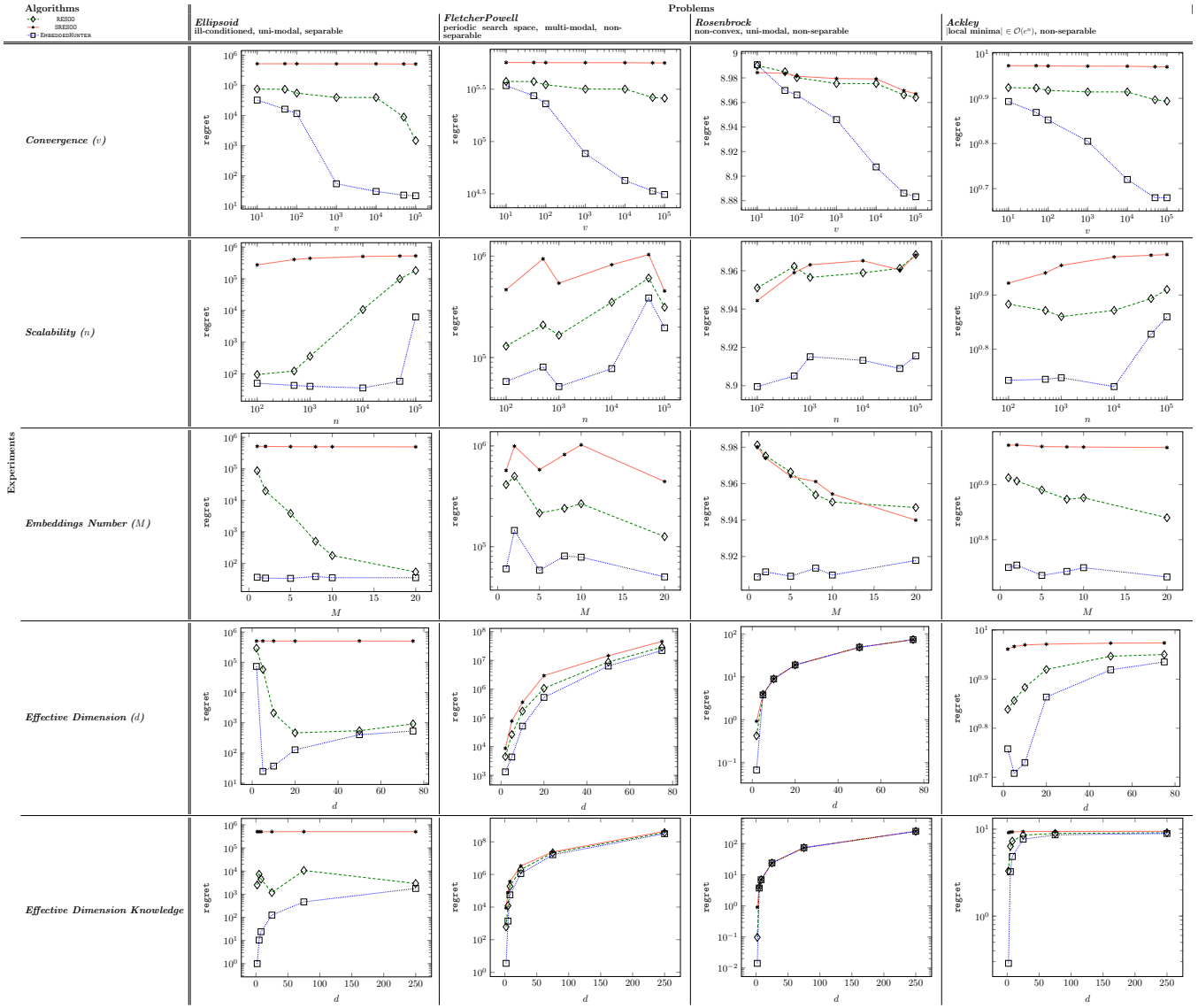
while evaluation budget is not exhausted **do**
  $\nu_{\min} \leftarrow \infty$
  for $l = 0$ **to** $\min\{\text{depth}(\mathcal{T}_t), h_{\max}\}$ **do**
    for $j = 1$ **to** $|\Gamma_{l,t}|$ **do**
      Select $(l, o) = \arg\min_{(h,i)\in\mathcal{L}^j_{t,l}} f^*_{h,i}$
      if $f^*_{l,o} < \nu_{\min}$ **then**
        $\nu_{\min} \leftarrow f^*_{l,o}$
        Expand (l, o) into its child nodes
        Evaluate (l, o)'s child nodes by g_P
        Add (l, o)'s child nodes to $\mathcal{T}_t$
      end if
    end for
    $\mathcal{T}_{t+1} \leftarrow \mathcal{T}_t$
    $t \leftarrow t + 1$
  end for
end while
return $f^*_v = \min_{(h,i)\in\mathcal{T}_t} f^*_{h,i}$

Numerical Validation: Setup

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, $h_{max} = \sqrt{v}$.

Numerical Validation: Results



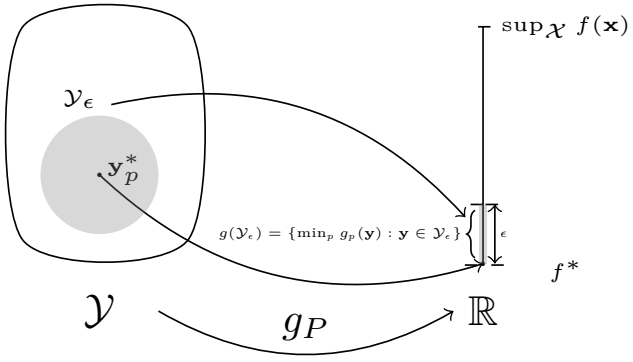
Theoretical Analysis

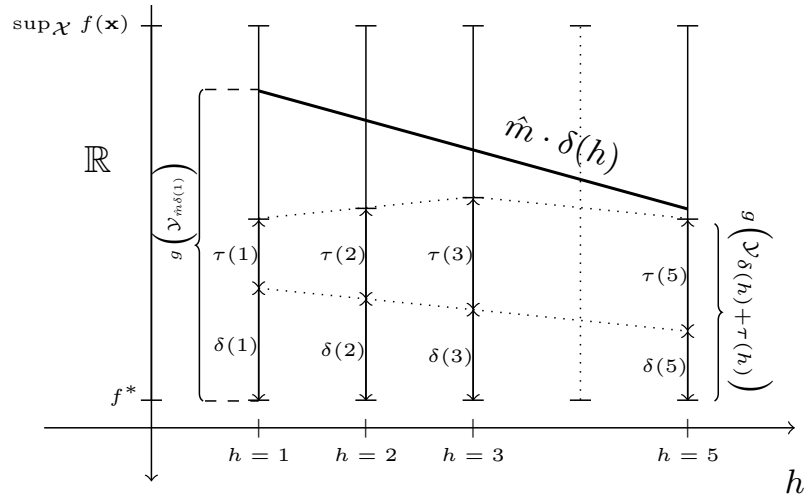
■ Define $h(t)$ as the smallest $h \geq 0$ such that:

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

where t is the number of iterations. Then `EmbeddedHunter`'s regret is bounded as

$$r(t) \leq \min_{h \leq \min(h(t), h_{max}+1)} \tau(h) + \delta(h) \ .$$





Code & Experiments

- https://ash-aldujaili.github.io/eh-lsopt/