Parallel Optimistic Optimization

jean-bastien.grill@inria.fr, michal.valko@inria.fr, and munos@google.com

Setting

POO is a global function maximizer:

Goal: Maximize $f : \mathcal{X} \to \mathbb{R}$ given a budget of n evaluations.

Challenges: f is ***stochastic*** and has ***unknown smoothness***

Protocol: At round t, select state x_t, observe r_t such that

$$\mathbb{E}[r_t | x_t] = f(x_t).$$

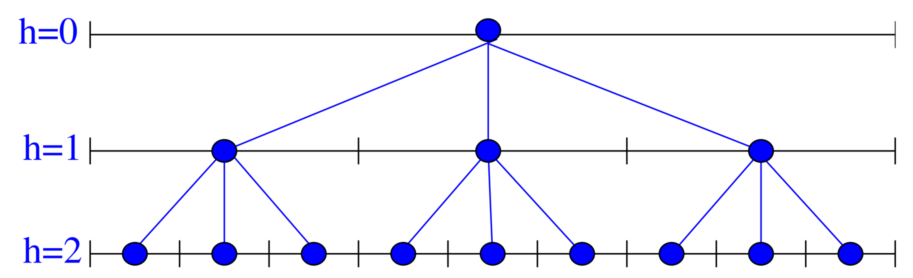
After n rounds, return a state $x(n)$.

Loss: $R_n = \sup_{x\in\mathcal{X}} f(x) - f(x(n))$

POO operates on a given **hierarchical partitioning** of $\mathcal{X}$:

For any h, $\mathcal{X}$ is partitioned in K^h cells $(\mathcal{X}_{h,i})_{0\le i\le K^h-1}$.

K-ary tree $\mathcal{T}_\infty$ where depth $h = 0$ is the whole $\mathcal{X}$.



Contributions

Extending class of functions that we can provably optimize.

Principled measure of the problem complexity.

	deterministic	stochastic
known smoothness	DOO	Zooming, HOO, HCT
unknown smoothness	DiRect, SOO	StoSOO, TaxonomyZoom, ATB, POO

Assumption

One *single* assumption:

Assumption 1. *There exists $\nu > 0$ and $\rho \in (0,1)$ such that*

$$\forall h \ge 0, \forall x \in \mathcal{P}_{h,i_h^\star},\quad f(x) \ge f(x^\star) - \nu\rho^h.$$

It's a one-side local Lipschitz-type of assumption constraining f only along the optimal path and **does not rely on any metric!**

Covers large class of functions: For example, any f with standard partitioning on $\mathbb{R}^p$ for which

$$f(x) \sim_{x\to x^\star} \beta\|x - x^\star\|^\alpha$$

Counter example: $f : x \mapsto 1/\ln x$ and a standard partitioning of $[0,1)$ does not verify Assumption 1.

Comparison to previous assumptions

Previous work assume there exists a semi-metric ℓ on $\mathcal{X}$ such that

A1 Local smoothness of f: For all $x \in \mathcal{X}$:

$$f(x^\star) - f(x) \le \ell(x, x^\star).$$

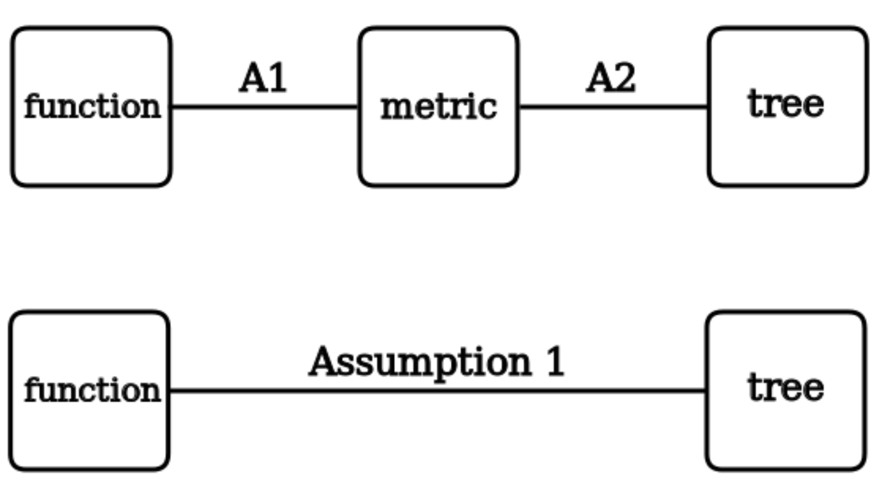
A2 Bounded diameters and well-shaped cells: There exist $\rho < 1$ and $\nu_1 \ge \nu_2 > 0$, such that for any depth $h \ge 0$ and index $i = 1,\dots,I_h$, the subset $\mathcal{P}_{h,i}$ is *contained by* and *contains* two open balls of radius $\nu_1\rho^h$ and $\nu_2\rho^h$ respectively.

We provide a **more natural** characterization

Prior algorithms don't use the metric. They only make use of (ν,ρ) and the partitioning.

What matters is **how much the partitioning fits** f. Any function can be trivially optimized given a perfectly adapted partitioning.

The metric is a **link** between the function and the partitioning.



By discarding the metric we merge the 2 assumptions. We convert a **topological** problem into a **combinatorial** one $\to$ Easier analysis!

Algorithm: POO

Parameters: K, $\mathcal{P} = \{\mathcal{P}_{h,i}\}$
  Optional parameters: $\rho_{\max}$, $\nu_{\max}$
Initialization:
  $D_{\max} \leftarrow \ln K/\ln(1/\rho_{\max})$
  $n \leftarrow 0$ {number of evaluation performed}
  $N \leftarrow 1$ {number of HOO instances}
  $\mathcal{S} \leftarrow \{(\nu_{\max}, \rho_{\max})\}$ {set of HOO instances}
while computational budget is available **do**
  while $N \ge \frac{1}{2}D_{\max}\ln(n/(\ln n))$ **do**
    for $i \leftarrow 1,\dots,N$ **do** {start new HOOs}
      $s \leftarrow \left(\nu_{\max}, \rho_{\max}{}^{2N/(2i+1)}\right)$
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{s\}$
      Perform $\frac{n}{N}$ function evaluation with HOO(s)
      Update the average reward $\widehat{\mu}[s]$ of HOO(s)
    end for
    $n \leftarrow 2n$
    $N \leftarrow 2N$
  end while{ensure there is enough HOOs}
  for $s \in \mathcal{S}$ **do**
    Perform a function evaluation with HOO(s)
    Update the average reward $\widehat{\mu}[s]$ of HOO(s)
  end for
  $n \leftarrow n + N$
end while
$s^\star \leftarrow \operatorname{argmax}_{s\in\mathcal{S}} \widehat{\mu}[s]$
Output: The deepest point evaluated by HOO($s^\star$)

How it works?

POO makes the use of HOO as a subroutine, an algorithm that *requires the knowledge* of the function smoothness.

POO automagically launches several HOO instances in parallel for different smoothness (ν,ρ)

At the end, POO selects the instance $s^\star$ which performed the best and returns the deepest point selected by this instance.

Why it works?

From the analysis: **few** HOO instances are needed – $O(\ln n)$.

From the experiments: most of the **evaluations are the same!**
$\to$ Saving time by sharing information over HOO instances.

Measure of complexity

Definition of the **near-optimality dimension** d

Definition 1. *For any partitioning $\mathcal{P}$, reals $\nu > 0$ and $\rho \in (0,1)$ verifying Assumption 1*

$$d(\nu,\rho) \stackrel{\text{def}}{=} \inf\left\{d' \in \mathbb{R}^+ : \exists C > 0,\ \forall h \ge 0,\ \mathcal{N}_h(2\nu\rho^h) \le C\rho^{-d'h}\right\}$$

where $\mathcal{N}_h(\varepsilon)$ is the number of near-optimal cells $\mathcal{P}_{h,i}$ of depth h i.e cells such that

$$\sup_{x\in\mathcal{P}_{h,i}} f(x) \ge f(x^\star) - \varepsilon$$

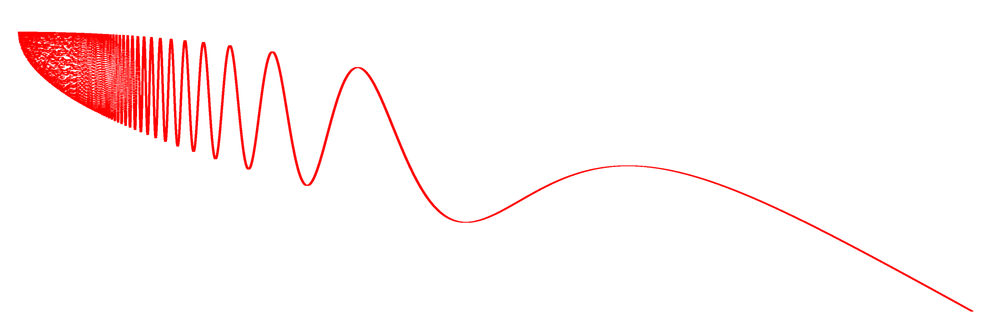
It measures **how much information** $\mathcal{P}$ gives us about f. The hierarchical partitioning $\mathcal{P}$ is the only prior information available.

It is the size of the near-optimal set. This set is the cells that any algorithm would have to sample in order to discover the optimum.

Examples of $d = 0$ functions. Any function with same order upper and lower envelopes near its maximum for the standard partitioning.

A $d > 0$ function for the standard partitioning.

$$f(x) = 1 - \sqrt{x} + (-x^2 + \sqrt{x})\cdot(\sin(1/x^2) + 1)/2$$



Functions that behave differently in different dimensions have also $d > 0$. Nonetheless, for a specifically handcrafted partitioning, it is possible to have $d = 0$ even for those functions.

Analysis

Theorem 1. *Let R_n be the simple regret of POO at step n. For any (ν,ρ) verifying Assumption 1 such that $\nu \le \nu_{\max}$ and $\rho \le \rho_{\max}$ there exists κ such that for all n*

$$\mathbb{E}[R_n] \le \kappa \cdot \left(\left(\ln^2 n\right)/n\right)^{1/(d(\nu,\rho)+2)}$$

$$\kappa = \alpha \cdot D_{\max}(\nu_{\max}/\nu_\star)^{D_{\max}}$$

Where α is a constant independent of $(\rho_{\max},\nu_{\max})$ and $D_{\max}$ is defined as

$$D_{\max} \stackrel{\text{def}}{=} (\ln K)/\ln(1/\rho_{\max})$$

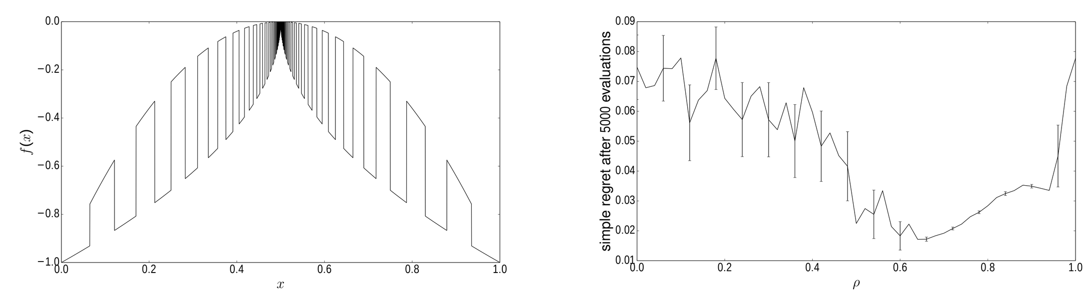
Matches performance of algorithms knowing the smoothness.

This is the performance of HOO run with $\nu_\star$ and $\rho_\star$

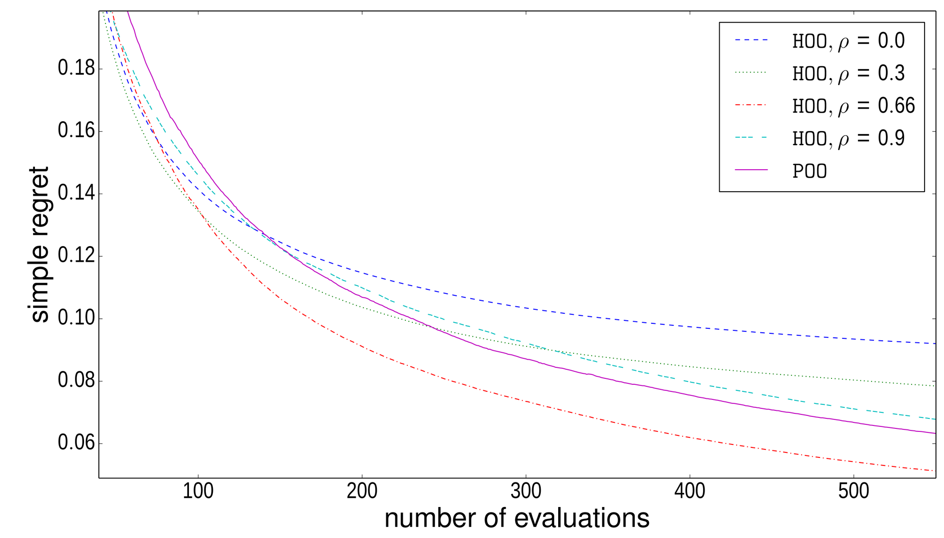
$$O\left(\left((\ln n)/n\right)^{1/(d(\nu_\star,\rho_\star)+2)}\right).$$

$D_{\max}$ is a **tight upper bound** on the near optimality dimension of any function verifying Assumption 1 for $\rho \le \rho_{\max}$.

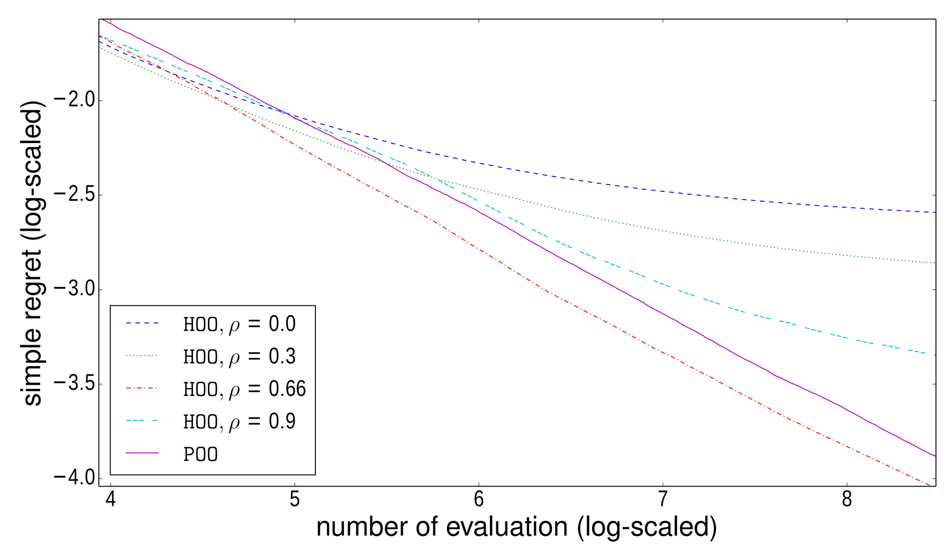
Experiments



Left: Function we ran experiments on. It has $d > 0$.
Right: Performance of a HOO instance as a function of ρ.



Regret after 500 evaluations of HOO with different ρ and POO.



Regret after 5000 evaluation in a log-log scale.

HOO with low value of ρ **gets stuck**. It does not explore enough.

HOO with high value of ρ **wastes time** to explore too much.

POO performs almost as well as optimally fitted HOO!

Among 100 instances **only two** needed a fresh evaluations.

Code at: `https://sequel.lille.inria.fr/Software/POO`

References

HOO: Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. *$\mathcal{X}$-armed Bandits*, NIPS 2009

SOO: Rémi Munos. *Optimistic Optimization of Deterministic Functions without the Knowledge of its Smoothness*, NIPS 2011.

TaxonomyZoom: Aleksandrs Slivkins. *Multi-armed Bandits on Implicit Metric Spaces*, NIPS 2011.

StoSOO: Michal Valko, Alexandra Carpentier, and Rémi Munos. *Stochastic Simultaneous Optimistic Optimization*, ICML 2013.

HCT: MG Azar, Alessandro Lazaric, and Emma Brunskill: *Online Stochastic Optimization under Correlated Bandit Feedback*. ICML 2014.

ATB: Adam D. Bull. *Adaptive-treed bandits*. Bernoulli, 2015.

Background: Optimistic Optimization for Trees

HOO **is close to** UCT but HOO has *finite-time* performance guarantees whereas UCT analysis is *asymptotic* only

HOO **follows an optimistic strategy:** HOO defines upper bounds for every path and selects the maximum one.

HOO **makes use of proper upper bounds** — defined as the minimum of $U_{h,i}(t)$ over the path.

The third term ρ^h in $U_{h,i}(t)$ is **function dependent**.

$$U_{h,i}(t) = \widehat{\mu}_{h,i}(t) + \sqrt{\frac{2\ln(t)}{N_{h,i}(t)}} + \nu\rho^h,$$

$\to$ t is the number of evaluations
$\to$ $\widehat{\mu}_{h,i}(t)$ is the empirical mean of f in $\mathcal{P}_{h,i}$
$\to$ $N_{h,i}(t)$ is the number of evaluations of f in $\mathcal{P}_{h,i}$.