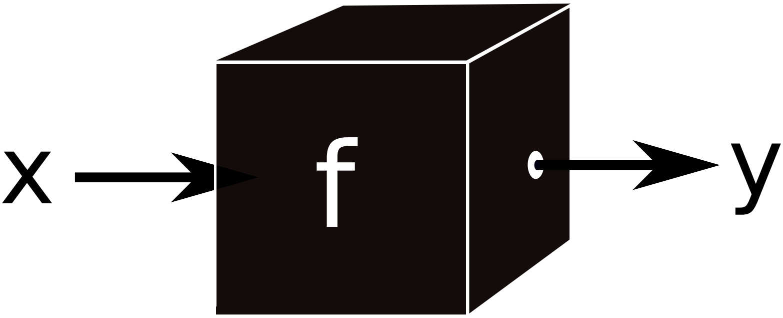
Simple, parameter-free and adaptive OPTIMIZATION with a MINIMAL local smoothness assumption

Peter Bartlett, Victor Gabillon, Michal Valko

ALT - March 23, 2019

Black box optimization



Also called zero-order optimization.

(before discussing the minimal assumptions, let us set the) Setting

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

Challenges: **First,** f has an **unknown smoothness**,

Later, f is stochastic with **unknown noise range** $\boldsymbol{b}$.

Protocol: At round t, select x_t, observe y_t such that

$$\mathbb{E}[y_t|x_t] = f(x_t) \qquad |y_t - x_t| \leq \boldsymbol{b}$$

After n rounds, **return** $x(n)$.

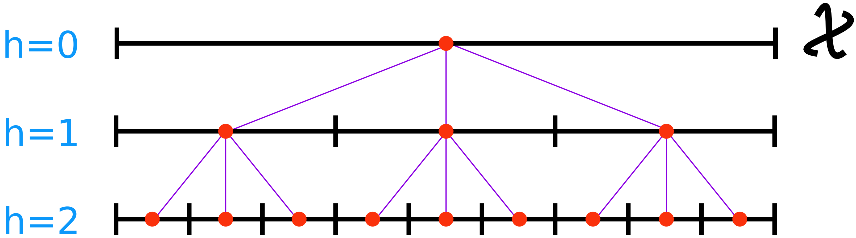
Loss: $r_n \triangleq \sup_{x\in\mathcal{X}} f(x) - f(x(n))$ **(simple regret)**

Minimal assumptions

- We want minimal assumptions.

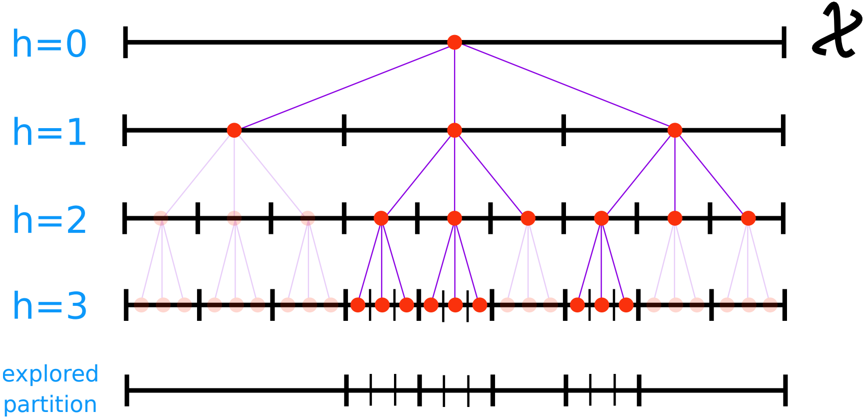
- The smoothness $\boldsymbol{d}$ of the function f is defined **with respect to** a fixed and given partitioning $\mathcal{P}$ of the search space $\mathcal{X}$.

- For any **depth** h, $\mathcal{X}$ is partitioned in K^h cells $(\mathcal{P}_{h,i})_{0\leq K^h-1}$.
- K-ary tree $\mathcal{T}$ where depth $h=0$ is the whole $\mathcal{X}$.



An example of partitioning in one dimension with $K=3$.

Optimizing becomes a **tree search** on the partition $\mathcal{P}$.



How to explore the tree smartly? (Track $x^\star$ as deep as poosible)

The assumption and the smoothness

Assumption (on the local smoothness around $x^\star$)

For some global optimum $x^\star$, there exists $\nu > 0$ and $\rho \in (0,1)$ such that $\forall h \in \mathbb{N}$, $\forall x \in \mathcal{P}_{h,i_h^\star}$,

$$f(x) \geq f(x^\star) - \nu\rho^h.$$

- The smoothness is local, around a $x^\star$.
- This guarantees that the algorithm will not under-estimate by more than $\nu\rho^h$ the value of optimal cell $\mathcal{P}_{h,i_h^\star}$ if it observes $f(x)$ with $x \in \mathcal{P}_{h,i_h^\star}$,.
- Now for the opposite question: How much non-optimal cells have values $\nu\rho^h$-close to optimal and therefore indiscernible from it? Let us **count** them!

The smoothness and the near-optimal dimension

Lets us bound $\mathcal{N}_h(3\nu\rho^h)$ as a function of the depth h.

- $d' > 0$ $\rightsquigarrow$ controls how $\mathcal{N}_h(3\nu\rho^h)$ explodes with h.
- $d' = 0$ $\rightsquigarrow$ bounded by a constant $\forall h$.

Definition

For any $\nu > 0$, $C > 1$, and $\rho \in (0,1)$, the **near-optimality dimension** $\boldsymbol{d}(\nu, C, \rho)$ of f with respect to the partitioning $\mathcal{P}$, is

$$\boldsymbol{d}(\nu, C, \rho) \triangleq \inf\left\{ d' \in \mathbb{R}^+ : \forall h \geq 0,\ \mathcal{N}_h(3\nu\rho^h) \leq C\rho^{-d'h} \right\},$$

where $\mathcal{N}_h(\varepsilon)$ is the number of cells $\mathcal{P}_{h,i}$ of depth h such that $\sup_{x\in\mathcal{P}_{h,i}} f(x) \geq f(x^\star) - \varepsilon$.

Previous work

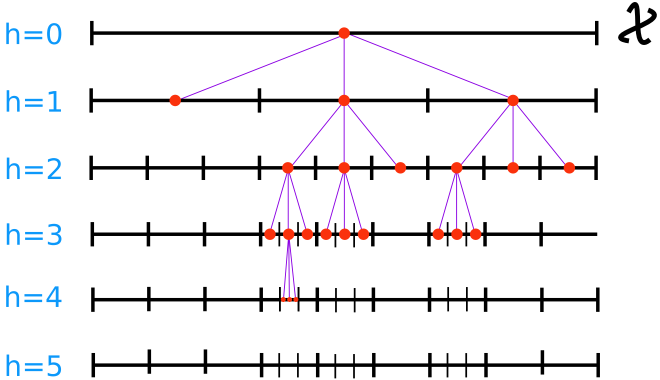
Previous and our new approaches under similar assumptions:

smoothness	**deterministic** $\boldsymbol{b} = 0$	**stochastic** $\boldsymbol{b} > 0$
(ν, ρ) known	`DOO`	`Zooming`, `HOO`
(ν, ρ) unknown	`DiRect`, `SOO`, `SequOOL`	`StoSOO`, `POO`, `StroquOOL`

- We tackle **unknown** smoothness (ν, ρ).
- Let us first consider $\boldsymbol{b} = 0$ and see how our new `SequOOL` improves upon `SOO`.

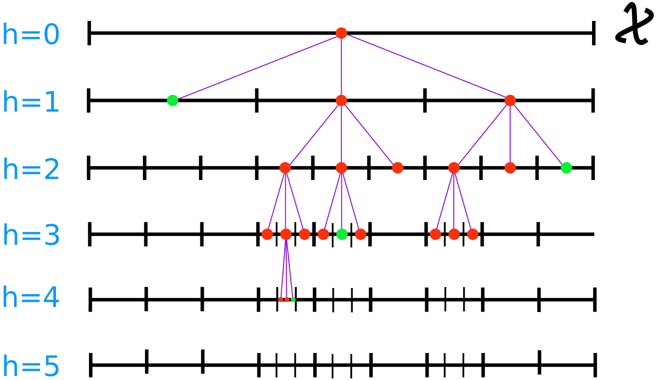
The SOO algorithm (Munos 2012)

Idea: Open *simultaneously* the cell with highest value at each depth h



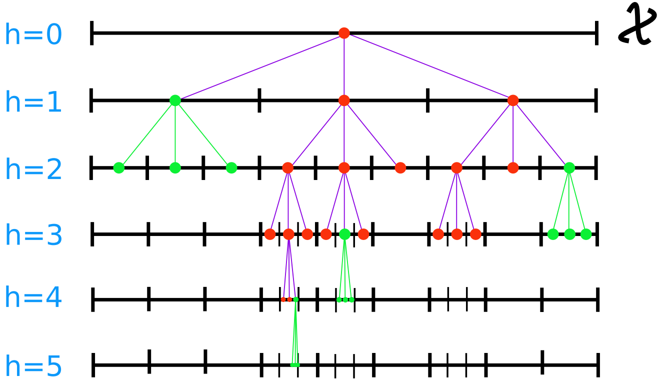
The SOO algorithm (Munos 2012)

Idea: Open *simultaneously* the cell with highest value at each depth h



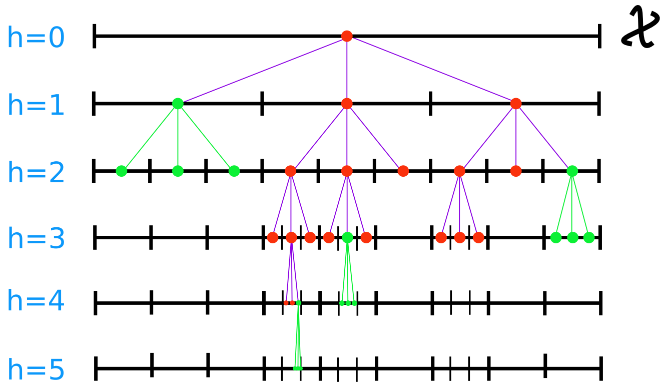
The SOO algorithm (Munos 2012)

Idea: Open *simultaneously* the cell with highest value at each depth h



The SOO algorithm (Munos 2012)

Idea: Open *simultaneously* the cell with highest value at each depth h



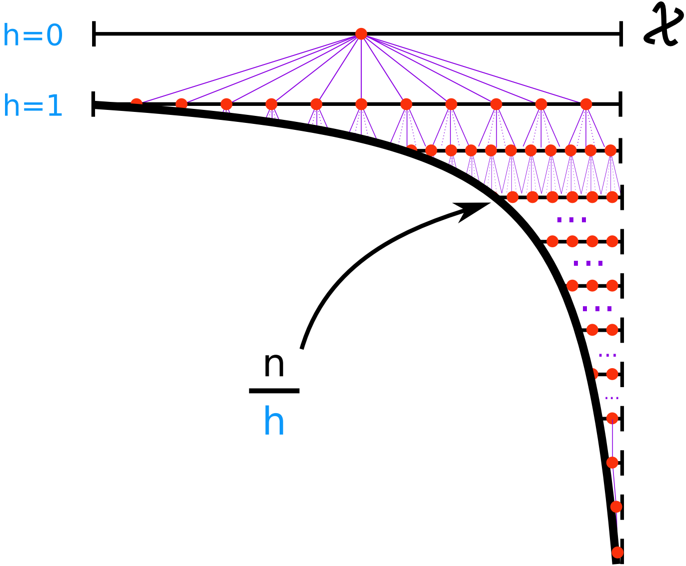
Why **simultaneous**? Why not **sequential**? Are all depths equal?

The sequential approach

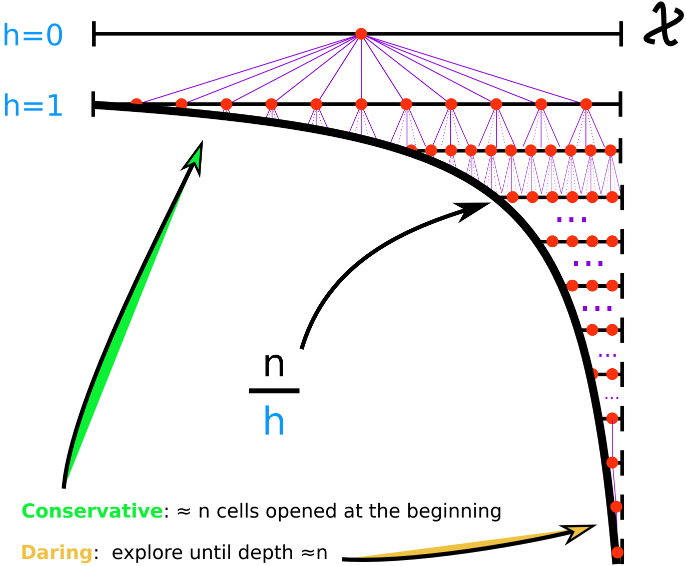
- **New approach:** First opens cells at depth h and then at depth $h+1$ and so on, without coming back to lower depths.
- **Why? Notice that:** the location of the exploration at depth $h+1$ is based on the exploration at depth h.
- **So:** Explore depth h *as best as you can* before starting exploring depth $h+1$.

Don't be **simultaneous**, be **sequential**: Let us introduce `SequOOL`.

Zipf exploration: Open best $\frac{n}{h}$ cells at depth h



Zipf exploration: Open best $\frac{n}{h}$ cells at depth h



Conservative: ≈ n cells opened at the beginning

Daring: explore until depth ≈n

Let us count the number of **openings** performed by `SequOOL` by summing over depths h.

$$n + \frac{n}{2} + \frac{n}{3} + \ldots + \frac{n}{h} + \ldots + 1 \approx n \log n$$

So instead of $\frac{n}{h}$ lets open $\frac{n}{h \log n}$ at each depth h.

The SequOOL algorithm

Parameters: n, $\mathcal{P} = \{\mathcal{P}_{h,i}\}$

Initialization: Open $\mathcal{P}_{0,1}$. $h_{\max} \leftarrow \lfloor n/\overline{\log}(n) \rfloor$.

For $h = 1$ to $h_{\max}$

- Open $\lfloor h_{\max}/h \rfloor$ cells $\mathcal{P}_{h,i}$ of depth h with largest values $f_{h,j}$.

Output $x(n) \leftarrow \underset{x_{h,i}:\mathcal{P}_{h,i} \in \mathcal{T}}{\arg\max} f_{h,i}$.

Simple and parameter free (doubling trick to forget n).

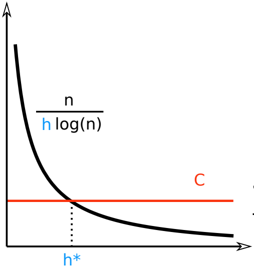
Simple regret r_n analysis

	$d > 0$	$d = 0$
SequOOL	$\left(\frac{\log(n)}{n}\right)^{\frac{1}{d}}$	$e^{-\frac{n}{\log(n)}}$
SOO(ε)	$\left(\frac{1}{n}\right)^{\frac{1-\varepsilon}{d}}$, for any $\varepsilon > 0$	$e^{-\sqrt{n}}$
DOO	$\left(\frac{1}{n}\right)^{\frac{1}{d}}$	e^{-n}

- The improvement is in gray (and exponential).
- We argue $d = 0$ is common and $d > 0$ needs an engineer.
- DOO knows the smoothness (ν, ρ) **(Munos 2012)**.

Main idea in the proof

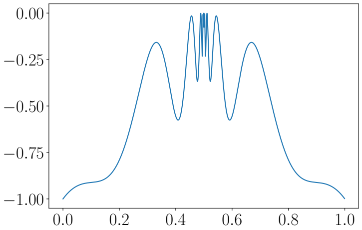
- When $\boldsymbol{d} = 0$ there are (at most) C near optimal cells at each depth h.
- Recursively, to track $x^\star$ at depth h: Be sure to open more than the best C cells at each depth h.
- **Oracle solution:** If C was known, just open the best C nodes until depth n/C.
- With Zipf: Ok as long as $n/(hlog(n)) \geq C \Rightarrow h \approx n/(Clog(n))$.
- Only lose a log factor.
- Intuition: find an integrable function with the heaviest tail.



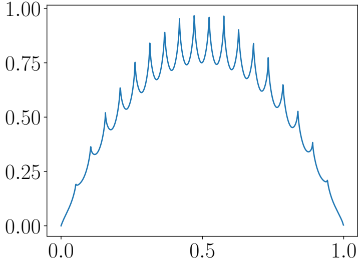
(Find a function that stays as long as possible above C and integrates to n without knowing C)

Some experiments

We play with one dimensional benchmarks.



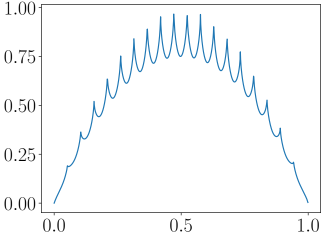
$d > 0$



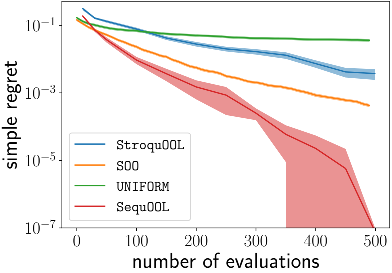
$d = 0$

SequOOL: Truly exponential rates

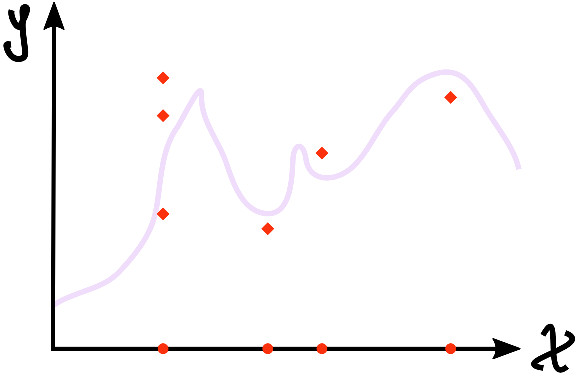
We play with 1 dimension benchmarks.



$d = 0$



Noisy case

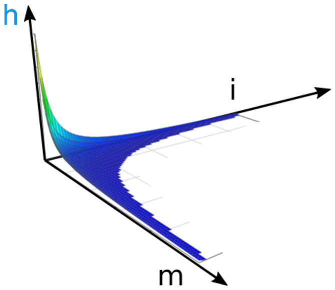


- Needs to pull more each x to limit uncertainty.
- **Tradeoff:** the more you pull each x the less deep you can explore.

Noisy case: StroquOOL

- **Idea:** Launch parallel SequOOL (m) with $m = 1, 2, 4, 8, \ldots, n$ where in SequOOL (m) each cell is pulled m times and the deepest explored $h_{\max} \approx n/m$.

- **Rephrased as:** At depth h order the cells by decreasing value and open the i-th best cell with $m = \frac{n}{hi}$ estimations.



Simple regret $\mathbb{E}r_n$ analysis

	$b > 0$	$b = 0$	
		$d > 0$	$d = 0$
StroquOOL	$\left(b\frac{\log^2(n)}{n}\right)^{\frac{1}{d+2}}$	$\left(\frac{\log^2(n)}{n}\right)^{\frac{1}{d}}$	$e^{-\frac{n}{\log^2(n)}}$
POO ($\widetilde{b}$)	$\left(\widetilde{b}\frac{\log(n)}{n}\right)^{\frac{1}{d+2}}$	$\left(\frac{\log(n)}{n}\right)^{\frac{1}{d+2}}$	$\left(\frac{\log(n)}{n}\right)^{\frac{1}{d+2}}$
SequOOL		$\left(\frac{\log^2(n)}{n}\right)^{\frac{1}{d}}$	$e^{-\frac{n}{\log(n)}}$

- No use of UCB in StroquOOL! No need to know the range of noise b! POO($\widetilde{b}$) needs to use $\widetilde{b}$.
- If $\widetilde{b} >> b > 0$, StroquOOL improves upon POO ($\widetilde{b}$).
- For $b = 0$, the improvement is in gray .
- We adapt to noise and recover almost the results of SequOOL when $b = 0$.

Thank you!