From Bandits to Monte Carlo Tree Search: The optimistic principle applied to Optimization and Planning

Rémi Munos

SequeL project: Sequential Learning
http://researchers.lille.inria.fr/∼munos/

INRIA Lille - Nord Europe / Microsoft-Research NE

AAAI 2013, Bellevue, WA

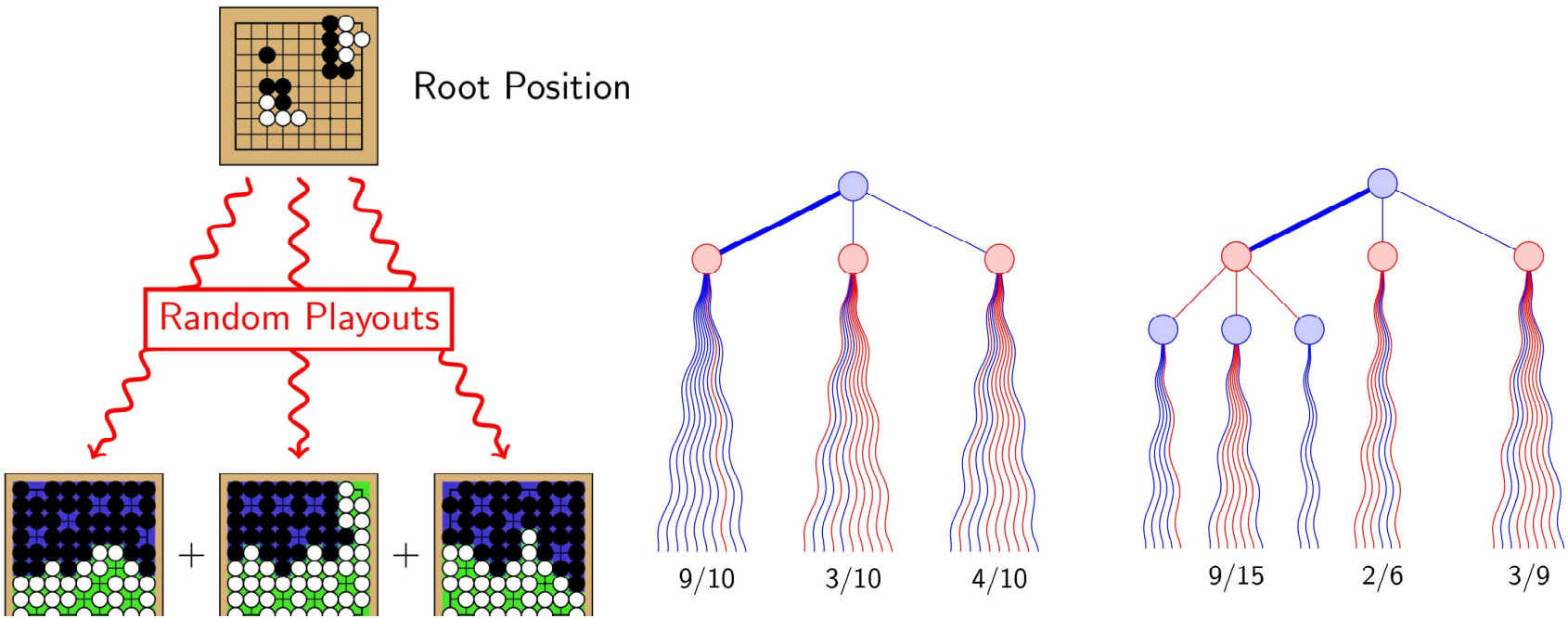
Thanks to ...

(in alphabetic order)

Jean-Yves Audibert, Sébastien Bubeck, Lucian Buşoniu, Alexandra Carpentier, Pierre-Arnaud Coquelin, Rémi Coulom, Sylvain Gelly, Jean-François Hren, Nathaniel Korda, Odalric-Ambrym Maillard, Gilles Stoltz, Csaba Szepesvári, Olivier Teytaud, Michal Valko, and Yizao Wang.

Initial motivation

Monte-Carlo Tree Search in computer-go



MCTS in Crazy-Stone (Rémi Coulom, 2005)

Idea: use bandits at each node of the tree search.

Monte-Carlo Tree Search

- Very efficient in several problems
- Very inefficient in many other problems (even toy problems)
- Not much theoretical guarantee...

We would like

- Understand how the "optimism in the face of uncertainty" principle works in hierarchical problems
- Define classes of problems for which variants of MCTS would be efficient

Outline of this tutorial

1. The stochastic multi-armed bandit
   - The K-armed bandit and UCB
   - The many-armed bandit
2. Monte Carlo Tree Search
   - Bandits in a hierarchy
   - Computer go and UCT
3. Optimistic optimization with known smoothness
   - Deterministic rewards
   - Stochastic rewards ($\mathcal{X}$-armed bandit)
4. Extension to unknown smoothness
5. Optimistic planning
   - Deterministic dynamics,
   - Open Loop planning
   - MDPs with a model

Main message of this tutorial

The **"optimism in the face of uncertainty" principle:**

- applies in a large class of decision making problems in stochastic and deterministic environments
- provides an efficient exploration of the search space by exploring the most promising areas first
- provides a natural transition from global to local search
- Performance depends on the "smoothness" of the function around the maximum w.r.t. some metric,
  - a measure of the quantity of near-optimal solutions,
  - and our knowledge or not of this metric.

The multi-armed bandit problem

Setting:

- Set of K arms (possible actions)
- At each time t, choose an arm $I_t \in \{1, \dots, K\}$ and receive a reward X_t coming from arm I_t.
- **Goal**: find an arm selection policy such as to maximize a function of the rewards.

Exploration-exploitation tradeoff:

- **Exploit**: act optimally according to our current beliefs
- **Explore**: learn more about the environment

Numerous variants

Different settings:

- **Stochastic environments**: the rewards are samples from probability distributions. We compare our strategy to the optimal oracle one.
- **Adversarial environments**: the rewards are chosen by an adversary. We compare our strategy to a class of possible strategies.
- **Action space** can be finite, countably infinite, continous (function optimization), combinatorial (paths), structured (policies), ...

Different targets:

- maximizing cumulative rewards, returning the best possible solution, estimating the values of all the arms, ...

Various applications

- Clinical trials [Thompson 1933]
- Ads placement on webpages
- Nash equilibria (traffic or communication networks, agent simulation, poker, ...)
- Packet routing, itinerary selection, ...
- Game-playing computers (Go, urban rivals, ...)
- Optimization / planning given a finite numerical budget

The stochastic multi-armed bandit problem

Setting:

- Set of K arms, defined by distributions ν_k (with support in $[0,1]$), whose law is unknown,
- At each time t, choose an arm $I_t \in \{1,\dots,K\}$ and receive reward $X_t \overset{i.i.d.}{\sim} \nu_{I_t}$.
- **Goal**: maximize the sum of rewards.

Definitions:

- Let $\mu_k = \mathbb{E}_{X\sim\nu_k}[X]$ be the mean value of arm k,
- Let $\mu^* = \max_k \mu_k$ the best mean value,

Define the cumulative **regret**:

$$R_n \overset{\text{def}}{=} \sum_{t=1}^{n} (\mu^* - X_t).$$

The cumulative regret

The expected cumulative regret is

$$\begin{aligned}
\mathbb{E}R_n &= \mathbb{E}\Big[\sum_{t=1}^{n} \mu^* - X_t\Big] \\
&= \mathbb{E}\Big[\mathbb{E}\Big[\sum_{t=1}^{n} \mu^* - X_t | I_t\Big]\Big] = \mathbb{E}\Big[\sum_{t=1}^{n} \mu^* - \mu_{I_t}\Big] \\
&= \mathbb{E}\Big[\sum_{k=1}^{K}(\mu^* - \mu_k)\sum_{t=1}^{n} \mathbf{1}\{I_t = k\}\Big] = \sum_{k=1}^{K} \Delta_k \mathbb{E}[T_k(n)],
\end{aligned}$$

where $\Delta_k \stackrel{\text{def}}{=} \mu^* - \mu_k$, and $T_k(n)$ is the number of times arm k has been pulled up to round n.

Goal: Find an arm selection policy such as to minimize $\mathbb{E}R_n$.

Proposed solutions

This is an old problem! [Robbins, 1952] Surprisingly, not fully solved yet!
Many proposed strategies:

- ϵ-**greedy exploration**: choose current best action with proba $1-\epsilon$, or random action with proba ϵ,
- **Bayesian exploration**: assign prior to the arm distributions and select arm according to the posterior distributions (Gittins index, Thompson sampling, ...)
- **Softmax exploration**: choose arm k with proba $\propto \exp(\beta \widehat{\mu}_k)$
- **Follow the perturbed leader**: choose best perturbed arm
- **Optimistic exploration**: select best arm in the most favorable environment compatible with observations

Optimism in the face of uncertainty

Follow the optimal policy in the most favorable environments among all enviroments that are reasonably plausible given past observations

- At time t, for each arm k, use past observations and some probabilistic argument to define high-probability confidence intervals containing the expected reward μ_k
- The most favorable environment for arm k is thus the upper confidence bound (UCB) on μ_k
- Simple implementation: play the arm having the largest UCB!

The UCB1 algorithm [Auer, Cesa-Bianchi, Fischer, 2002]

Upper Confidence Bound algorithm: at each time t, select the arm with highest UCB:

$$B_{k,t} \stackrel{\text{def}}{=} \widehat{\mu}_{k,t} + \sqrt{\frac{2\log(t)}{T_k(t)}},$$

where:

- $\widehat{\mu}_{k,t} = \frac{1}{T_k(t)} \sum_{s=1}^{t} X_s \mathbf{1}\{I_s = k\}$ is the empirical mean of the rewards received from arm k up to time t,
- $T_k(t) = \sum_{s=1}^{t} \mathbf{1}\{I_s = k\}$ is the number of times arm k has been pulled up to time t

Note that

- Sum of an *exploitation term* and an *exploration term*.
- $\sqrt{\frac{2\log(t)}{T_k(t)}}$ is a confidence interval term, so $B_{k,t}$ is a UCB.

Intuition of the UCB algorithm

Idea:

- The B-values $B_{k,t}$ are h.p. UCBs on μ_k. Indeed we have:

$$\mathbb{P}\Big(|\widehat{\mu}_{k,t} - \mu_k| \geq \sqrt{\frac{2\log(t)}{T_k(t)}}\Big) \quad \leq \quad \frac{2}{t^2},$$

using a union bound for all possible values of $T_k(t) \in \{1, \dots, t\}$ together with Chernoff-Hoeffding's inequality:

$$\mathbb{P}\Big(|\frac{1}{m}\sum_{i=1}^{m} Y_i - \mu| \geq \epsilon\Big) \quad \leq \quad 2e^{-2m\epsilon^2}$$

Why does it make sense?

Could we stay a long time playing a wrong arm?

Why does it make sense?

Could we stay a long time playing a wrong arm?

No, since

- The more we pull an arm k, the smaller the size of the confidence interval and the closer its UCB gets to its mean value μ_k
- But in h.p., it cannot be pulled once its UCB becomes smaller than μ^*

So each sub-optimal arm k can be only pulled a number of times $T_k(n)$ such that the size of its confidence interval $\sqrt{\frac{2 \log n}{T_k(n)}}$ is of order $\Delta_k = \mu^* - \mu_k$.

Regret bound for UCB

Proposition 1.

Each sub-optimal arm k is visited in average, at most:

$$\mathbb{E}T_k(n) \leq 8\frac{\log n}{\Delta_k^2} + 1 + \frac{\pi^2}{3}$$

times (where $\Delta_k \overset{\text{def}}{=} \mu^* - \mu_k > 0$).

Theorem 1.

Thus the expected regret is bounded by:

$$\mathbb{E}R_n = \sum_k \mathbb{E}[T_k(n)]\Delta_k \leq 8 \sum_{k:\Delta_k>0} \frac{\log n}{\Delta_k} + K(1 + \frac{\pi^2}{3}).$$

Intuition of the proof

Let k be a sub-optimal arm, and k^* be an optimal arm. At time t, if arm k is selected, this means that

$$\begin{aligned} B_{k,t} &\geq B_{k^*,t} \\ \widehat{\mu}_{k,t} + \sqrt{\frac{2\log(t)}{T_k(t)}} &\geq \widehat{\mu}_{k^*,t} + \sqrt{\frac{2\log(t)}{T_{k^*}(t)}} \\ \mu_k + 2\sqrt{\frac{2\log(t)}{T_k(t)}} &\geq \mu^*, \text{ with high proba} \\ T_k(t) &\leq \frac{8\log(t)}{\Delta_k^2} \end{aligned}$$

Thus, if $T_k(t) > \frac{8\log(t)}{\Delta_k^2}$, then there is only a small probability that arm k can be selected.

Proof of Proposition 1

Write $u = \frac{8\log(n)}{\Delta_k^2} + 1$. We have:

$$\begin{aligned} T_k(n) &\leq u + \sum_{t=u+1}^{n} \mathbf{1}\{I_t = k; T_k(t) > u\} \\ &\leq u + \sum_{t=u+1}^{n} \left[\mathbf{1}\{\hat{\mu}_{k,t} - \mu_k \geq \sqrt{\frac{2\log t}{T_k(t)}}\} + \mathbf{1}\{\hat{\mu}_{k^*,t} - \mu_k \leq -\sqrt{\frac{2\log t}{T_{k^*}(t)}}\}\right] \end{aligned}$$

Now, taking the expectation of both sides,

$$\begin{aligned} \mathbb{E}[T_k(n)] &\leq u + \sum_{t=u+1}^{n} 2t^{-2} \\ &\leq \frac{8\log(n)}{\Delta_k^2} + 1 + \frac{\pi^2}{3} \end{aligned}$$

Better confidence bounds imply smaller regret

- Chernoff-Hoeffding's inequality $1/t$-confidence bound:

$$\mathbb{E}X \leq \frac{1}{m}\sum_{i=1}^{m} X_i + \sqrt{\frac{\log t}{2m}}$$

- Bernstein's inequality:

$$\mathbb{E}X \leq \frac{1}{m}\sum_{i=1}^{m} X_i + \sqrt{\frac{2\sigma^2 \log t}{m}} + \frac{\log t}{3m}$$

- Empirical Bernstein's inequality:

$$\mathbb{E}X \leq \frac{1}{m}\sum_{i=1}^{m} X_i + \sqrt{\frac{2\widehat{\sigma}_t^{\,2} \log(3t)}{m}} + \frac{8\log(3t)}{3m}$$

[Audibert, Munos, Szepesvári, 2007], [Maurer, Pontil, 2009]

UCB-V

[Audibert, Munos, Szepesvári, 2007]

- **UCB using empirical variance estimate:**

$$B_{k,t} \stackrel{\text{def}}{=} \widehat{\mu}_{k,t} + \sqrt{2\frac{\widehat{\sigma_{k,t}}^2 \log(1.2t)}{T_k(t)}} + \frac{3\log(1.2t)}{T_k(t)}.$$

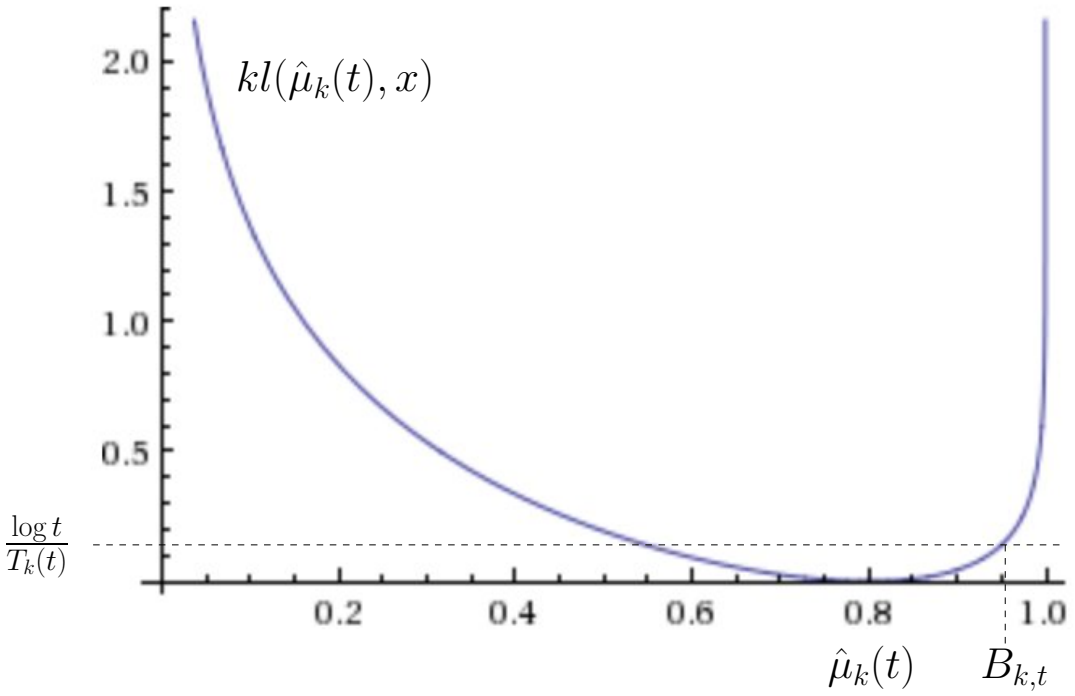
Then the expected regret is bounded as:

$$\mathbb{E}R_n \leq 10\Big(\sum_{k:\Delta_k>0} \frac{\sigma_k^2}{\Delta_k} + 2\Big)\log(n).$$

KL-UCB

[Garivier & Cappé, 2011] and [Maillard, Munos, Stoltz, 2011].
For Bernoulli distributions, define the kl-UCB

$$B_{k,t} \overset{\text{def}}{=} \sup \left\{ x \in [0,1],\ \mathrm{kl}(\hat{\mu}_k(t), x) \leq \frac{\log t}{T_k(t)} \right\}$$



(non-asymptotic version of Sanov's theorem)

KL-UCB

The regret of KL-UCB is then bounded as

$$\mathbb{E}R_n = \sum_{k:\Delta_k>0} \frac{\Delta_k}{\mathrm{kl}(\nu_k, \nu^*)} \log n + o(\log n).$$

This extends to several classes of distributions (one-dimensional exponential family, finitely supported, ...)
See also DMED [Honda, Takemura, 2010, 2011] and other related algorithms.

Idea: Use the full empirical distribution to get a refined UCB.

Lower bounds

For single-dimensional distributions [Lai, Robbins, 1985]:

$$\liminf_{n\to\infty} \frac{\mathbb{E}R_n}{\log n} \geq \sum_{k:\Delta_k>0} \frac{\Delta_k}{KL(\nu_k, \nu^*)}$$

For larger class of distributions $\mathcal{D}$ [Burnetas, Katehakis, 1996]:

$$\liminf_{n\to\infty} \frac{\mathbb{E}R_n}{\log n} \geq \sum_{k:\Delta_k>0} \frac{\Delta_k}{\mathcal{K}_{\inf}(\nu_k, \mu^*)},$$

where

$$\mathcal{K}_{\inf}(\nu, \mu) \stackrel{\text{def}}{=} \inf \left\{ KL(\nu, \nu') : \nu' \in \mathcal{D} \text{ and } \mathbb{E}_{X\sim\nu'}[X] > \mu \right\}.$$

Distribution-independent bound for UCB

Regret for UCB: $\mathbb{E}R_n = O(\log n \sum_{k \neq k^*} \frac{1}{\Delta_k})$

- The smaller the gaps, the harder the problem (for large n)
- For small n, the regret is trivially bounded by $n \max_k \Delta_k$

For small gaps it takes a long time to distinguish which arm is the best.

Proposition 2.

Distribution-independent bounds:

$$\sup_{\text{Distributions}} \mathbb{E}R_n \leq 2\sqrt{2Kn\left[\log n + 1 + \frac{\pi^2}{3}\right]}$$

Proof of the Proposition

We have

$$\begin{aligned}
\mathbb{E}R_n &= \sum_k \Delta_k \mathbb{E}T_k(n) \\
&= \sum_k \Delta_k \sqrt{\mathbb{E}T_k(n)}\sqrt{\mathbb{E}T_k(n)} \\
&\leq \sqrt{\sum_k \Delta_k^2 \mathbb{E}T_k(n)}\sqrt{\sum_k \mathbb{E}T_k(n)} \\
&\leq \sqrt{8Kn\Big[\log n + 1 + \frac{\pi^2}{3}\Big]}
\end{aligned}$$

since $\mathbb{E}T_k(n) \leq 8\frac{\log n}{\Delta_k^2} + 1 + \frac{\pi^2}{3}$ and $\sum_k T_k(n) = n$.

Distribution-independent bounds

We also have the lower bound [Cesa-Bianchi, Lugosi, 2006]:

$$\inf_{Algo} \sup_{Distributions} \mathbb{E}R_n = \Omega\left(\sqrt{Kn}\right)$$

Notice that a refined algorithm (MOSS [Audibert, Bubeck, 2009]) achieves the same order:

$$\sup_{Distributions} \mathbb{E}R_n = O\left(\sqrt{Kn}\right).$$

Many-armed bandits: an example

There is a countably infinite number of arms.

Unstructured set of actions: The rewards received so far do not tell us anything about the value of unobserved arms.

Example: Each day, select a restaurant:

- among the ones where you have already been
  - because it is good (Exploitation)
  - or not well known (Exploration)
- or choose a new one randomly (Discovery)

Other examples: Marketing (ex: sending catalogues), mining for valuable resources, ...

Many-armed bandits: Assumptions

We make a (probabilistic) assumption about the mean-value of any new arm.

- **Usual assumption**: the distribution of the mean-reward of a new arm is known [Banks, Sundaram, 1992], [Berry, Chen, Zame, Heath, Shepp, 1997].
- **Much weaker assumption:** Assume we know $\beta > 0$ such that

$$\mathbb{P}(\mu(\text{new arm}) > \mu^* - \epsilon) = \Theta(\epsilon^\beta),$$

β characterizes the probability of selecting near-optimal arms

Large $\beta \implies$ small chance of pulling good arm, thus one needs to pull many arms. And vice-versa.

UCB with Arm Increasing Rule [Wang, Audibert, Munos, 2008]

UCB-AIR:

- $K(0) = 0$. At time $t+1$, pull a new arm if

$$K(t) < \begin{cases} t^{\frac{\beta}{2}} & \text{if } \beta < 1 \text{ and } \mu^* < 1 \\ t^{\frac{\beta}{\beta+1}} & \text{if } \beta \geq 1 \text{ or } \mu^* = 1 \end{cases}$$

- Otherwise, apply UCB–V on the $K(t)$ current arms, i.e., play

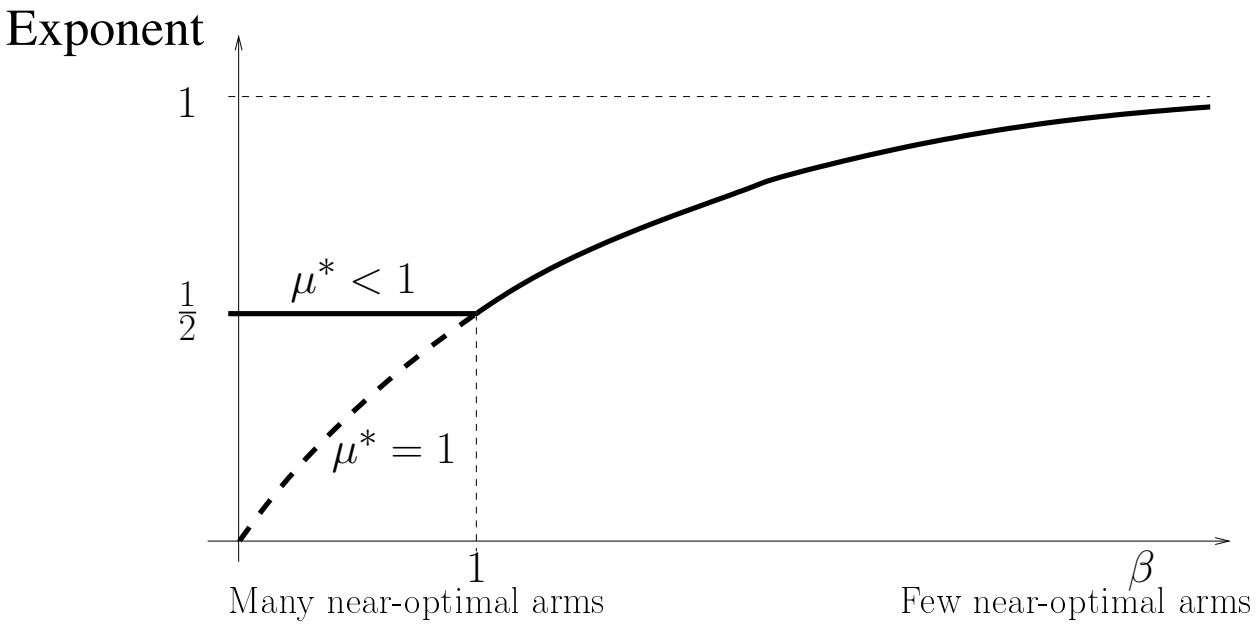
$$\underset{1 \leq k \leq K(t)}{\operatorname{argmax}} \underbrace{\hat{\mu}_{k,t}}_{\text{empirical rewards}} + \underbrace{\sqrt{\frac{2\widehat{V}_{k,t}\mathcal{E}_t}{T_k(t)}} + \frac{3\mathcal{E}_t}{T_k(t)}}_{\text{Confidence interval}},$$

with exploration sequence: $c \log(\log t) \leq \mathcal{E}_t \leq \log t$.

Regret analysis of UCB-AIR

Upper bound on the regret of UCB-AIR:

$$\mathbb{E}R_n = \begin{cases} \widetilde{O}(\sqrt{n}) & \text{if } \beta < 1 \text{ and } \mu^* < 1 \\ \widetilde{O}(n^{\frac{\beta}{1+\beta}}) & \text{if } \mu^* = 1 \text{ or } \beta \geq 1 \end{cases}$$



Lower bound: $\forall \beta > 0, \mu^* \leq 1$, for any algorithm $\mathbb{E}R_n = \Omega(n^{\frac{\beta}{1+\beta}})$.

Remarks and possible extensions

Remarks

- When $\beta > 1$ or $\mu^* = 1$ the upper and lower bounds match (up to logarithmic factor).
- Exploration-Exploitation-Discovery tradeoff:
  - **Exploitation**: Pull a good arm
  - **Exploration**: Pull an uncertain arm
  - **Discovery**: Pull a new arm
- The exploration sequence $\mathcal{E}_t$ can be of order $\log\log t$ (instead of $\log t$): discovery replaces exploration
- **Open question**: similar performance when β is unknown? (i.e. adaptive strategy that estimates β while minimizing regret).

Bandits with a structured set of actions

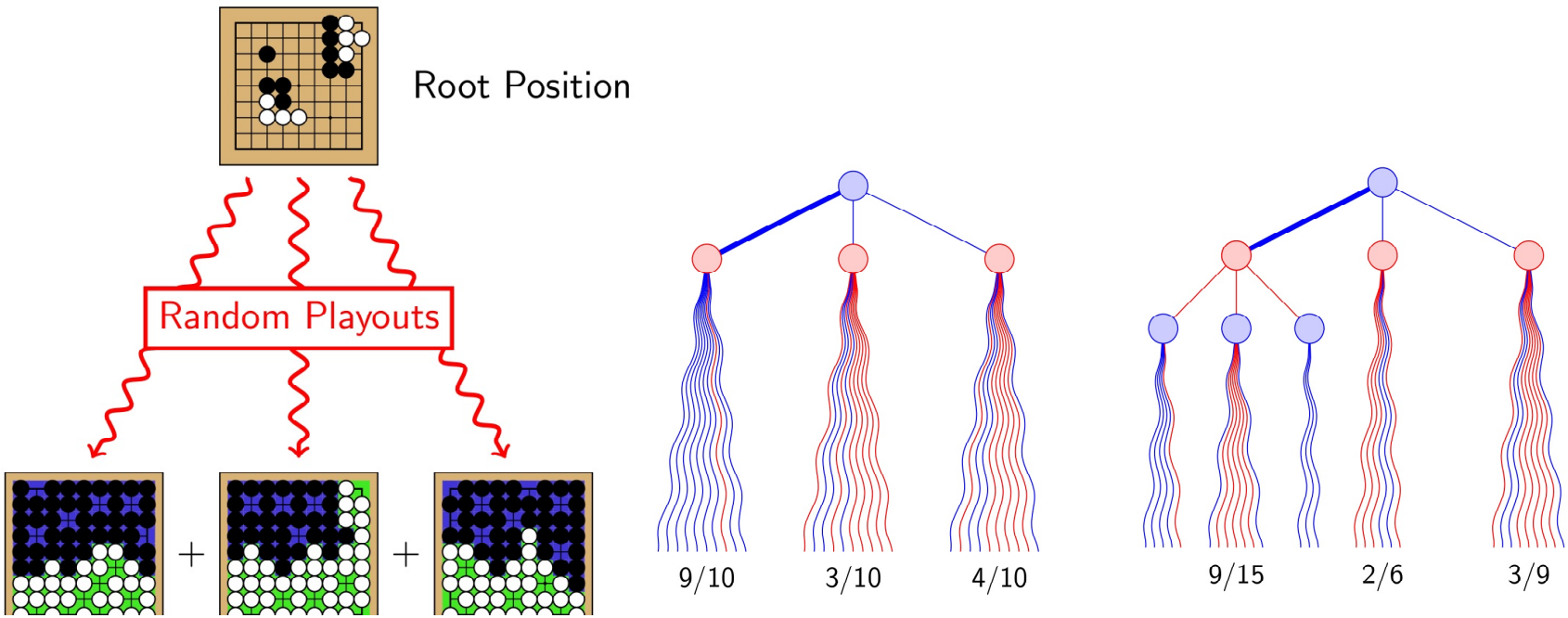
Optimism in the face of uncertainty extends to:

- Linear bandits [Auer, 2002], [Dani, Hayes, Kakade, 2008], [Abbasi-Yadkori, 2009], [Rusmevichientong, Tsitsiklis, 2010], [Filippi, Cappé, Garivier, Szepesvári, 2010]
- Convex bandits [Zinkevich, 2003], [Flaxman, Kalai, McMahan, 2005], [Hazan, Agarwal, Kale, 2006], [Bartlett, Hazan, Rakhlin, 2007], [Shalev-Shwartz, 2007], [Abernethy, Bartlett, Rakhlin, Tewari, 2008], [Narayanan, Rakhlin, 2010]
- Lipschitz bandits [Agrawal, 1995], [Kleinberg, 2004], [Auer, Ortner, Szepesvári, 2007], [Kleinberg, Slivkins, Upfall, 2008], [Bubeck, Munos, Stoltz, Szepesvári, 2011]
- Gaussian bandits [Dorard, Glowacka, Shawe-Taylor, 2009], [Grünewälder, Audibert, Opper, Shawe-Taylor, 2010], [Srinivas, Krause, Kakade, Seeger, 2010]
- Contextual bandits [Woodroofe, 1979], [Auer, 2002], [Wang, Kulkarni, Poor, 2005], [Pandey, Agarwal, Chakrabarti, Josifovski, 2007], [Langford, Zhang, 2007], [Hazan, Megiddo, 2007], [Rigollet, Zeevi, 2010], [Chu, Li, Reyzin, Schapire, 2011], [Slivkins, 2011]
- MDPs [Burnetas, Katehakis, 1997], [Jaksch, Ortner, Auer, 2010], [Bartlett, Tewari, 2009]
- Combinatorial bandits [Cesa-Bianchi, Lugosi, 2009], [Audibert, Bubeck, Lugosi, 2011]

Bandits in a hierarchy

- Bandit = tool to rapidly select the right action, given noisy estimate of their value
- Serve as building block for more complex problems
- Hierarchy of bandits: The reward obtained when pulling an arm is itself the return of another bandit in a hierarchy.
- Illustration: Monte-Carlo Tree Search in computer-go.

Monte-Carlo Tree Search in computer-go



MCTS in Crazy-Stone (Rémi Coulom, 2005)

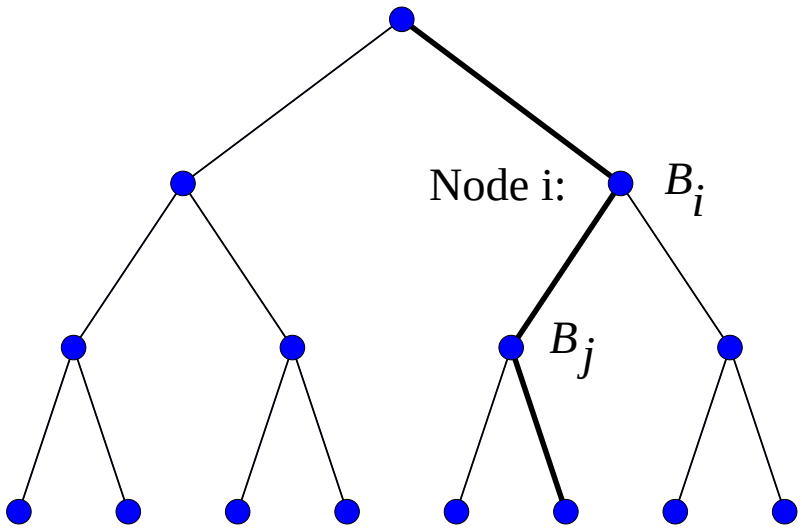
Idea: use bandits at each node of the tree search.

Hierarchical bandit algorithm

Upper Confidence Bound (UCB) algo at each node

$$B_j(t) \stackrel{\text{def}}{=} \hat{\mu}_j(t) + \sqrt{\frac{2\log(t)}{T_j(t)}}.$$

Intuition:
- Explore first the most promising branches
- Average converges to max



- Adaptive Multistage Sampling (AMS) algorithm [Chang, Fu, Hu, Marcus, 2005]
- UCB applied to Trees (UCT) [Kocsis and Szepesvári, 2006]

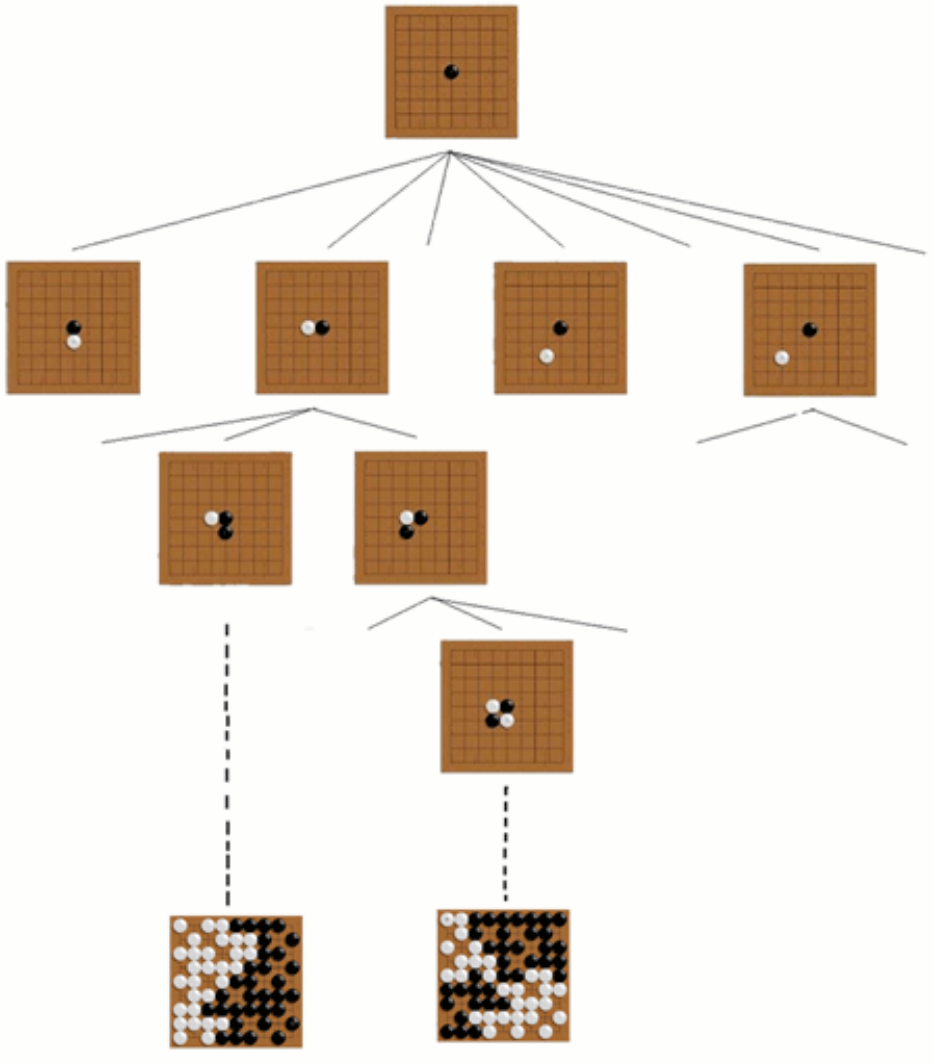
The MoGo program

[Gelly, Wang, Munos, Teytaud, 2006] + many others.

Features:

- Explore-Exploit with UCT
- Monte-Carlo evaluation
- Asymmetric tree expansion
- Anytime algo
- Use of features

Among world best programs!

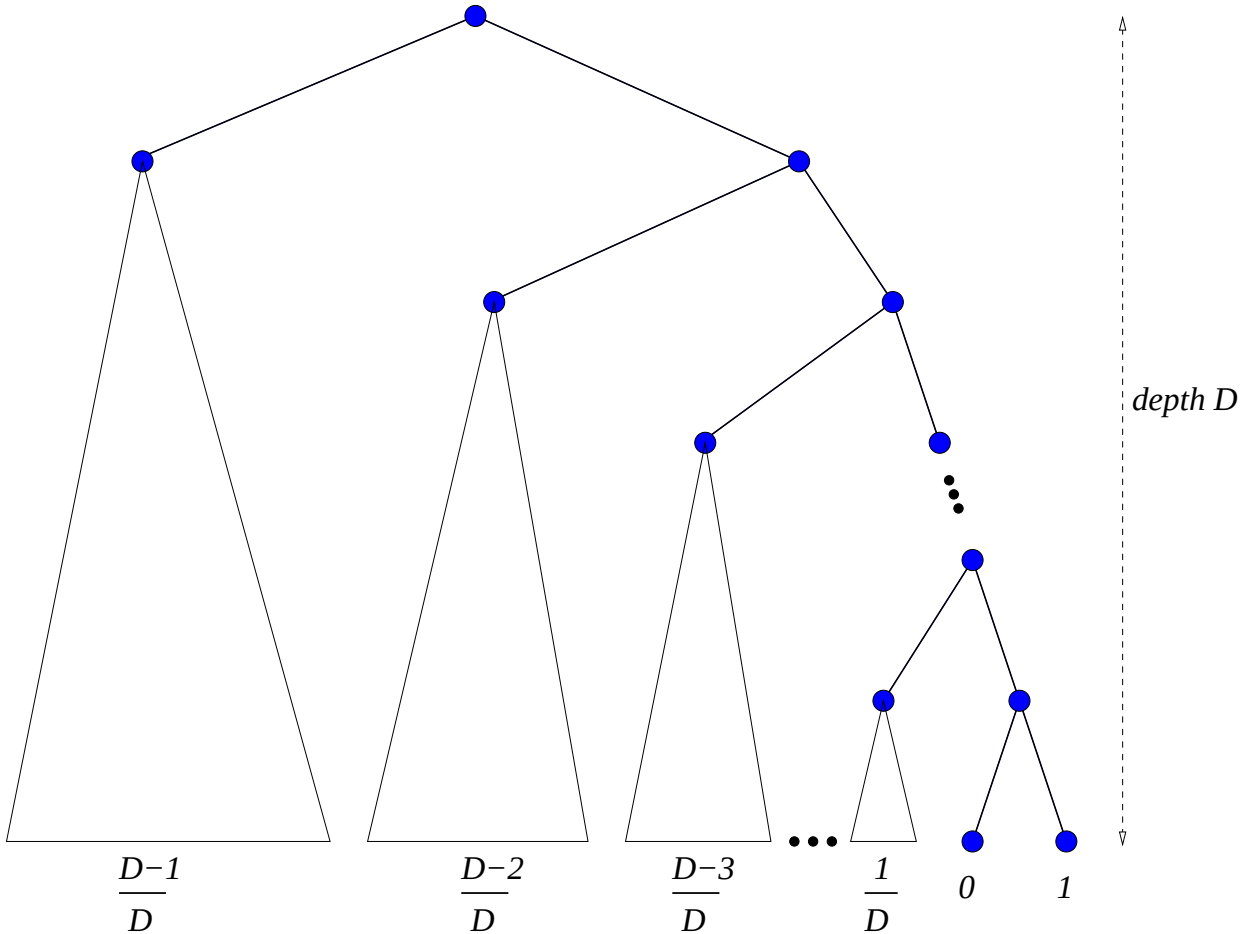


Asymptotic analysis of UCT

[Kocsis and Szepesvári, 2006]

- In a tree with finite depth, all leaves will be eventually explored an infinite number of times, thus by backward induction, UCT is consistent and the regret is $O(\log n)$.
- However, the constant can be so bad that there is not finite-time guarantee for any reasonable n.

Bad example for UCT



The left branches are explored exponentially more often than the right ones.

Finite-time analysis of UCT

The regret is disastrous: (see [Coquelin and Munos, 2007])

$$\mathbb{E}R_n = \Omega(\underbrace{\exp(\exp(\ldots \exp(}_{D \text{ times}} 1) \ldots))) + O(\log(n)),$$

whereas a uniform exploration of the tree would be "only" exponential in D.

Problem: at each node, the rewards are not i.i.d.
$\Longrightarrow$ **the B-values are not UCBs**.

UCT implicitely makes the assumption that the underlying function is very smooth.
Problems:

- Can we recover the optimistic principle?
- How should we define the smoothness of a function?

Optimization of a deterministic Lipschitz function

Problem: Find online the maximum of $f : X \to \boldsymbol{R}$, assumed to be Lipschitz:

$$|f(x) - f(y)| \leq \ell(x, y).$$

Protocol:

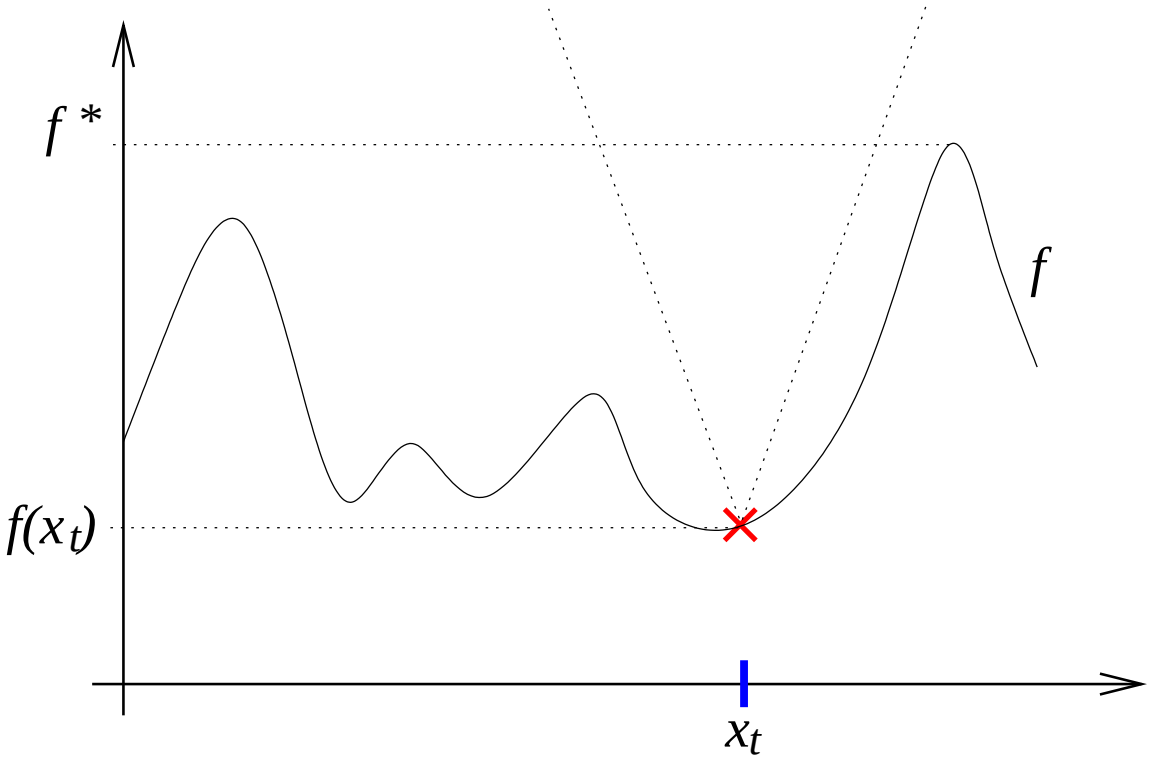
- For each time step $t = 1, 2, \ldots, n$ select a state $x_t \in X$
- Observe $f(x_t)$
- Return a state $x(n)$

Performance assessed in terms of the simple regret

$$r_n = f^* - f(x(n)),$$

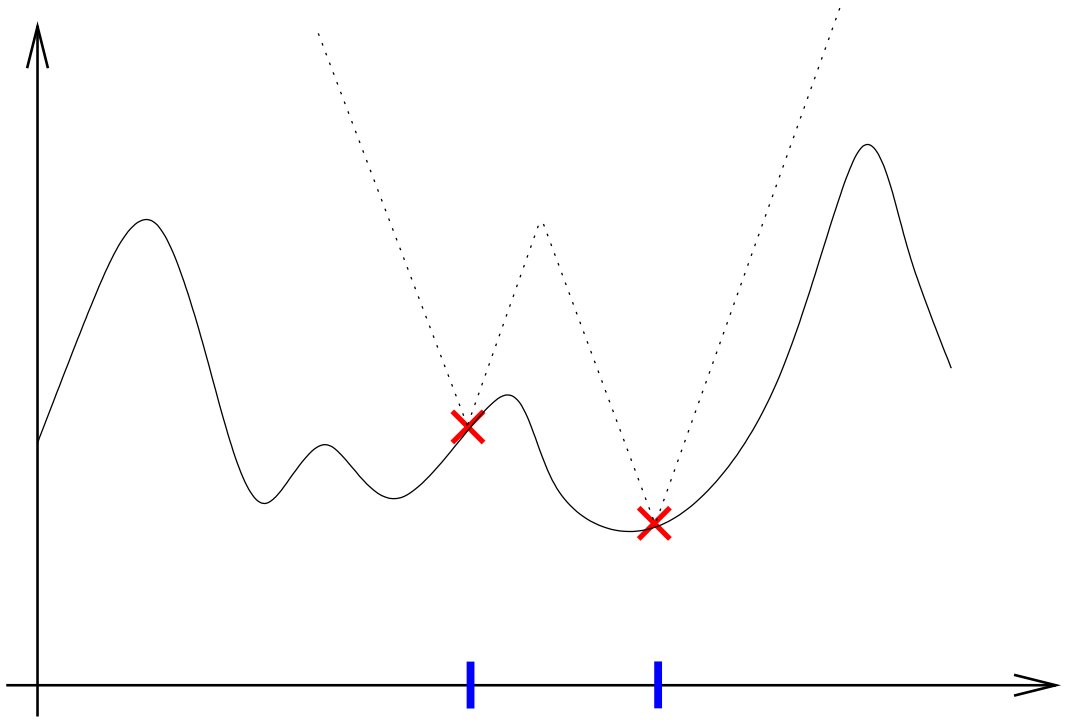
where $f^* = \sup_{x \in X} f(x)$.

Example in 1d



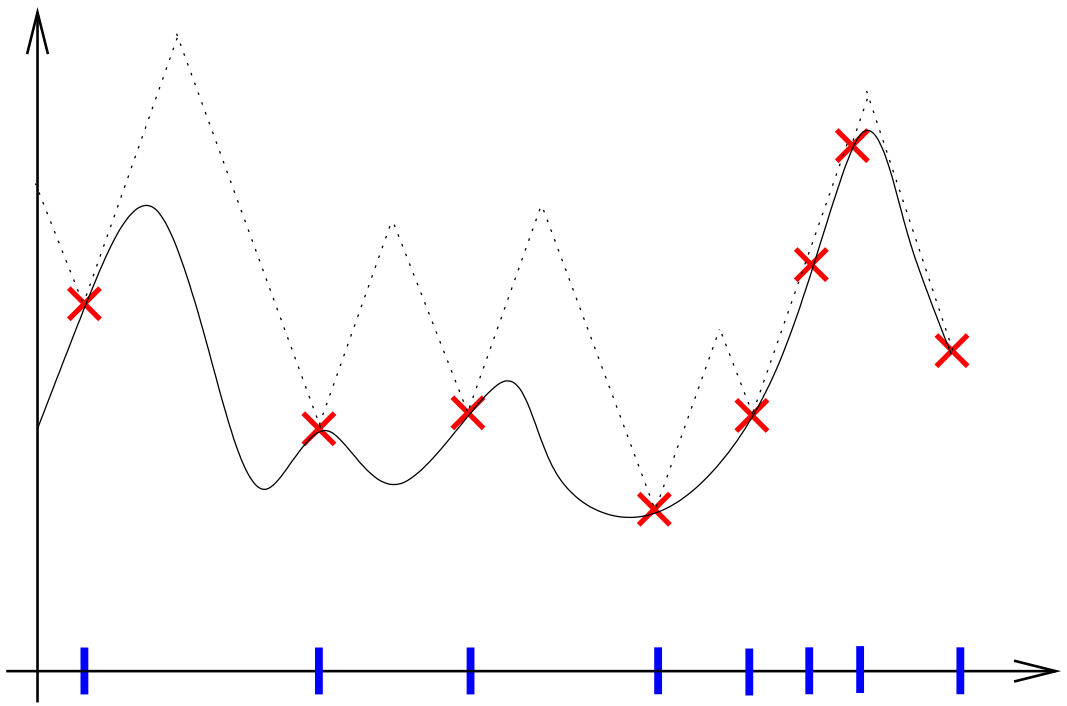
Lipschitz property $\rightarrow$ the evaluation of f at x_t provides a first upper-bound on f.

Example in 1d (continued)



New point $\rightarrow$ refined upper-bound on f.

Example in 1d (continued)



Question: where should one sample the next point?

Answer: select the point with highest upper bound!

"Optimism in the face of (partial observation) uncertainty"

Several issues

1. Lipschitz assumption is quite strong
2. Finding the optimum of the upper-bounding function may be hard!
3. How to handle noise?

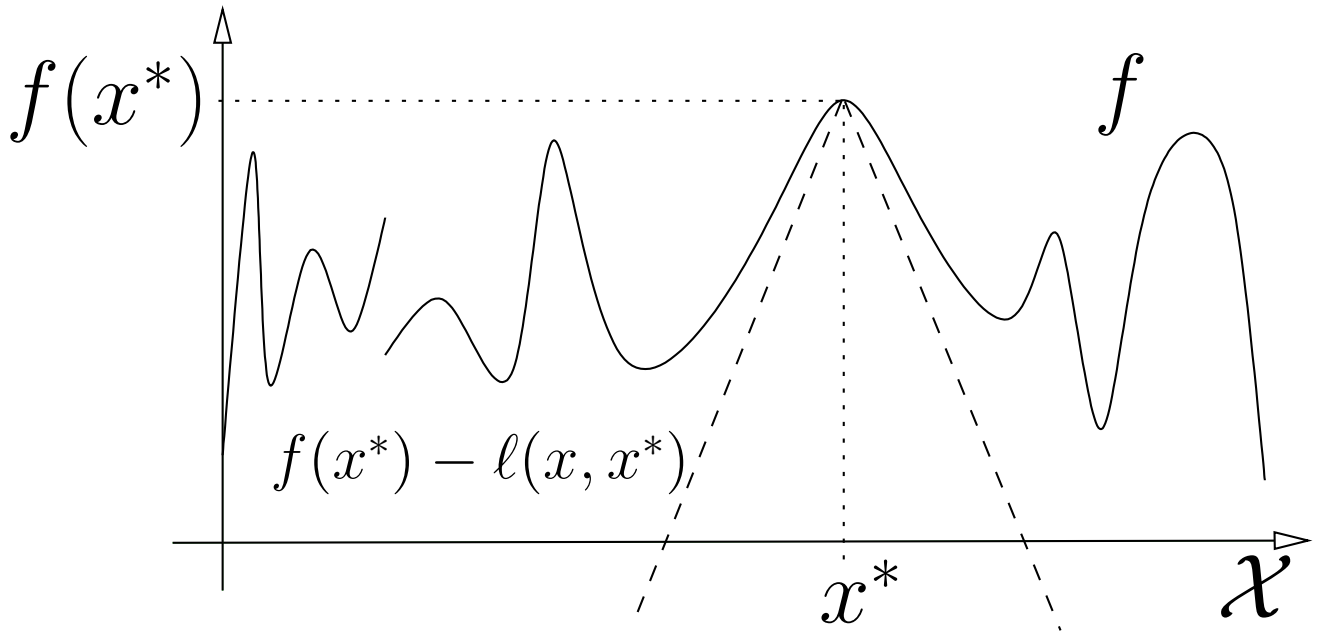
Local smoothness property

Assumption: f **is "locally smooth" around its max. w.r.t.** ℓ

1. $\mathcal{X}$ is equipped with a **semi-metric** ℓ: ℓ is symmetric, and $\ell(x, y) = 0 \Leftrightarrow x = y$.
2. For all $x \in \mathcal{X}$,

$$f(x^*) - f(x) \leq \ell(x, x^*).$$

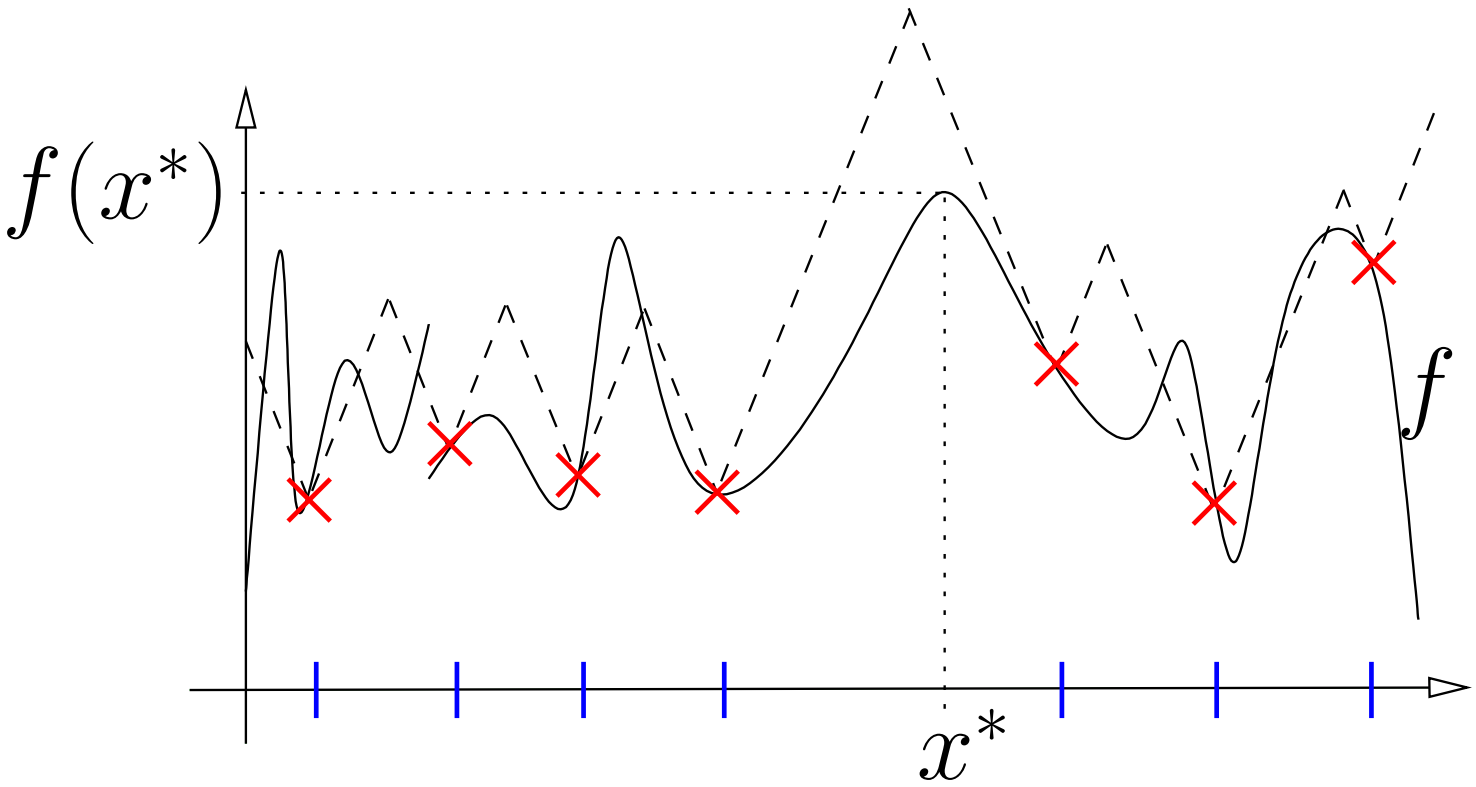
Local smoothness property



For all $x \in \mathcal{X}$,

$$f(x) \geq f(x^*) - \ell(x, x^*).$$

Local smoothness is enough!



Optimistic principle only requires:

- a true bound at the maximum
- the bounds gets refined when adding more points

Efficient implementation

Deterministic Optimistic Optimization (DOO) builds an adaptive partitioning of the domain where cells are refined according to their upper bounds.

- For $t = 1$ to n,
  - Let $\mathcal{T}_t$ be the current partition with cells X_i
  - Define an upper bound for each cell:

$$B_i = f(x_i) + \mathrm{diam}(X_i),$$

    where $x_i \in X_i$ and $\mathrm{diam}(X_i) \stackrel{\mathrm{def}}{=} \sup_{x,y \in X_i} \ell(x, y)$
  - Select the cell with highest bound

$$I_t = \operatorname*{argmax}_i B_i.$$

  - Expand I_t: refine the grid and evaluate f in children cells
- Return $x(n) \stackrel{\mathrm{def}}{=} \operatorname{argmax}_{\{x_t\}_{1 \le t \le n}} f(x_t)$

Properties of DOO

- At any time t, let X_{i^*} be the cell containing x^* and $x_{i^*} \in X_{i^*}$ the point where the function has been evaluated. Then

$$f(x^*) \leq f(x_{i^*}) + \ell(x^*, x_{i^*}) \leq f(x_{i^*}) + \text{diam}(X_{i^*}) = B_{i^*}$$

- Thus any suboptimal cell X_i such that

$$f(x_i) + \text{diam}(X_i) < f(x^*)$$

  will never be expanded.
- Thus finite-time performance guarantees can be obtained.

Near-optimality dimension

Define the **near-optimality dimension** of f as the smallest $d \geq 0$ such that $\exists C$, $\forall \epsilon$, the set of ε-optimal states

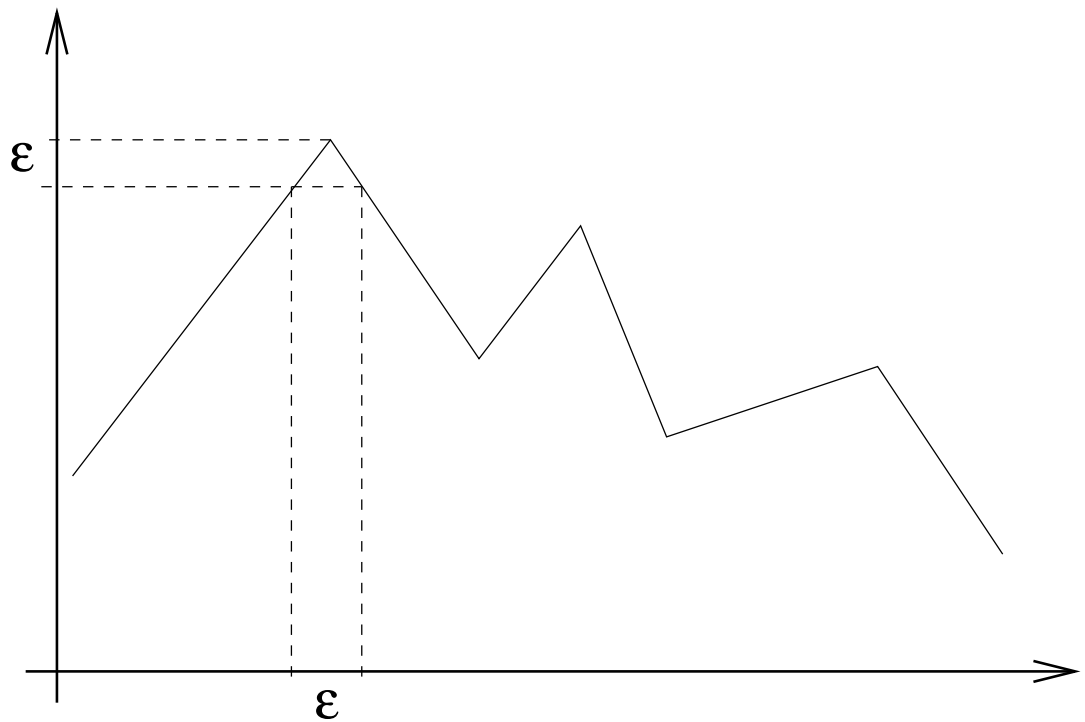
$$X_\varepsilon \stackrel{\text{def}}{=} \{x \in X, f(x) \geq f^* - \varepsilon\}$$

can be covered by $C\varepsilon^{-d}$ ℓ-balls of radius ε.

Example 1:

Assume the function is piecewise linear at its maximum:

$$f(x^*) - f(x) = \Theta(||x^* - x||).$$

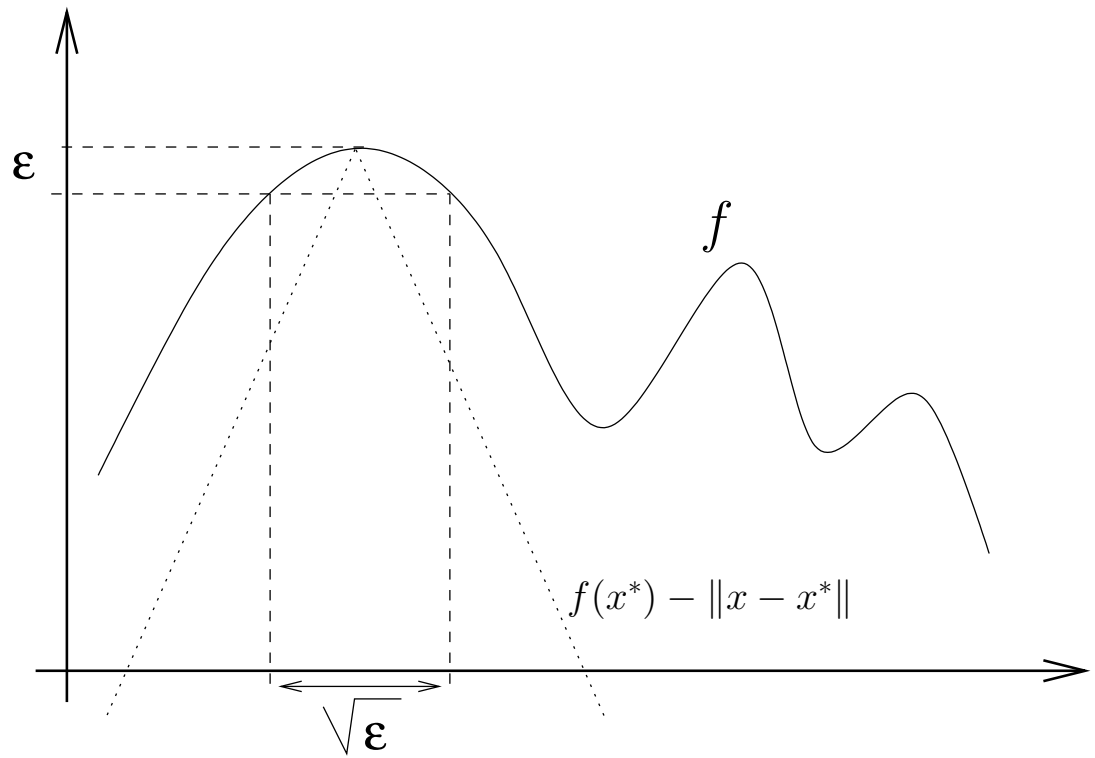


Using $\ell(x, y) = \|x - y\|$, it takes $O(\epsilon^0)$ balls of radius ϵ to cover X_ε. Thus $d = 0$.

Example 2:

Assume the function is locally quadratic around its maximum:

$$f(x^*) - f(x) = \Theta(||x^* - x||^2).$$

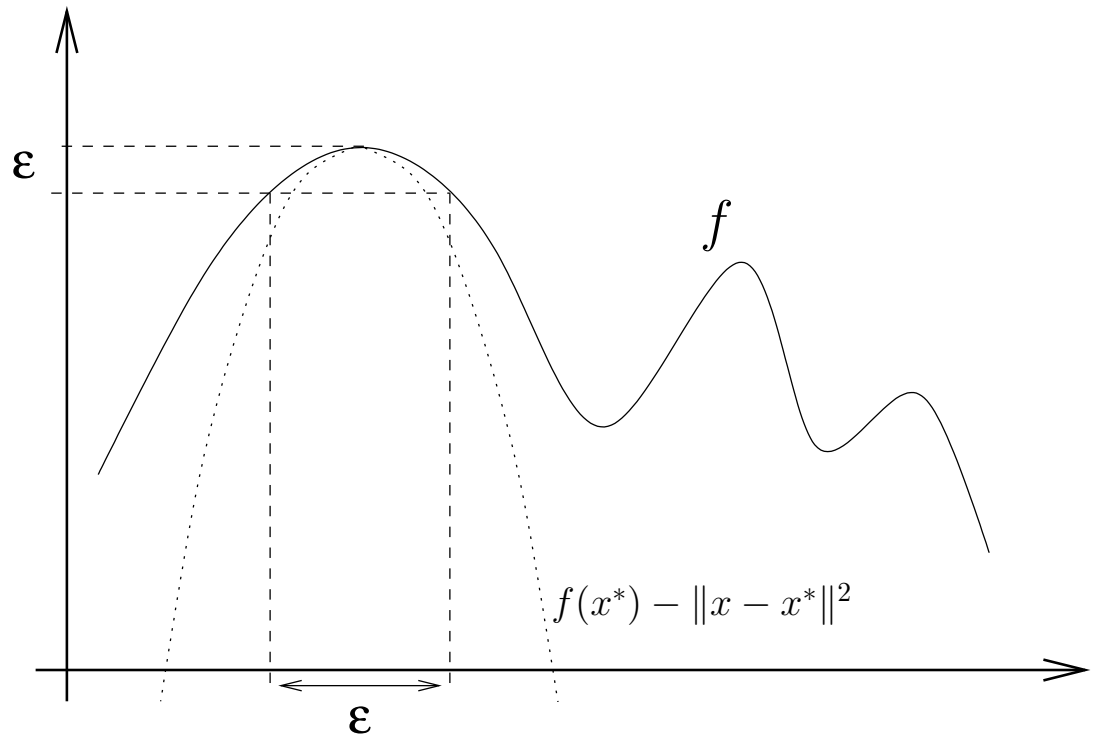


For $\ell(x, y) = ||x - y||$, it takes $O(\epsilon^{-D/2})$ balls of radius ϵ to cover X_ε (of size $O(\epsilon^{D/2})$). Thus $d = D/2$.

Example 2:

Assume the function is locally quadratic around its maximum:

$$f(x^*) - f(x) = \Theta(||x^* - x||^2)$$

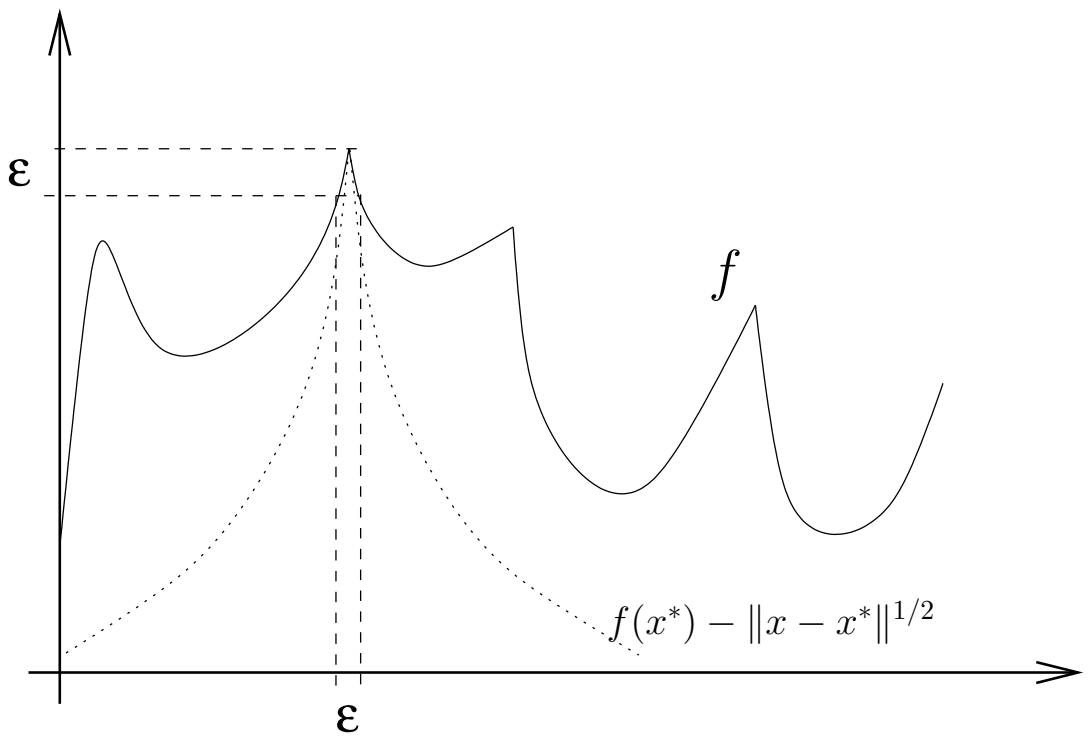


For $\ell(x, y) = ||x - y||^2$, it takes $O(\epsilon^0)$ ℓ-balls of radius ϵ to cover X_ε. Thus $d = 0$.

Example 3:

Assume the function has a square-root behavior around its maximum:

$$f(x^*) - f(x) = \Theta(||x^* - x||^{1/2})$$



For $\ell(x, y) = \|x - y\|^{1/2}$ we have $d = 0$.

Example 4:

Assume $\mathcal{X} = [0,1]^D$ and f is locally equivalent to a polynomial of degree $\alpha > 0$ around its maximum (i.e. f is α-smooth):

$$f(x^*) - f(x) = \Theta(||x^* - x||^\alpha)$$

Consider the semi-metric $\ell(x,y) = \|x - y\|^\beta$, for some $\beta > 0$.

- If $\alpha = \beta$, then $d = 0$.
- If $\alpha > \beta$, then $d = D(\frac{1}{\beta} - \frac{1}{\alpha}) > 0$.
- If $\alpha < \beta$, then the function is not locally smooth wrt ℓ.

Analysis of DOO (deterministic case)

Assume that the ℓ-diameters of the nodes of depth h decrease exponentially fast with h (i.e., $\mathrm{diam}(h) = c\gamma^h$, for some $c > 0$ and $\gamma < 1$).
This is true for example when $\mathcal{X} = [0,1]^D$ and $\ell(x,y) = \|x-y\|^\beta$ for some $\beta > 0$.

Theorem 2.

The loss of DOO is

$$r_n = \begin{cases} \left(\frac{C}{1-\gamma^d}\right)^{1/d} n^{-1/d} & \text{for } d > 0, \\ c\gamma^{n/C-1} & \text{for } d = 0. \end{cases}$$

(Remember that $r_n \overset{\text{def}}{=} f(x^*) - f(x(n))$).

Sketch of proof

- Only cells X_i of depth h such that $f(x_i) + \text{diam}(h) \geq f(x^*)$ may be expanded by DOO
- From the definition of d, the number of such cells is less that $C\text{diam}(h)^{-d}$
- The number of node expansions $n \leq C \sum_{h=0}^{h_{\max}} \text{diam}(h)^{-d}$
- For $d > 0$, $n = O\left(\text{diam}(h_{\max})^{-d}\right)$ and the value of the returned point $x(n)$ is at least as good as $f(x_{\max})$ for the deepest expanded node (of depth $h_{\max}$):

  $$f(x(n)) \geq f(x_{\max}) \geq f(x^*) - \text{diam}(h_{\max}) \geq f(x^*) - O(n^{-1/d}).$$

- For $d = 0$, $n = Ch_{\max}$ and $f(x(n)) \geq f(x^*) - O(\gamma^{n/C})$.

About the local smoothness assumption

Assume f satisfies $f(x^*) - f(x) = \Theta(||x^* - x||^\alpha)$.

Use DOO with the semi-metric $\ell(x, y) = \|x - y\|^\beta$:

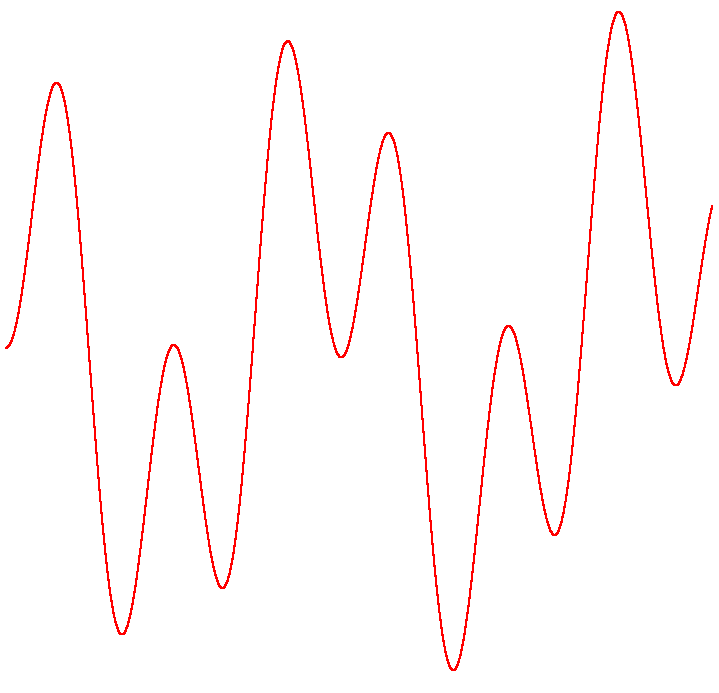
- If $\alpha = \beta$, then $d = 0$: the true "local smoothness" of the function is known, and exponential rate is achieved.
- If $\alpha > \beta$, then $d = D(\frac{1}{\beta} - \frac{1}{\alpha}) > 0$: we under-estimate the smoothness, which causes more exploration than needed.
- If $\alpha < \beta$: We over-estimate the true smoothness and DOO may fail to find the global optimum.

The performance of DOO heavilly relies on our knowledge of the true local smoothness.

Experiments [1]

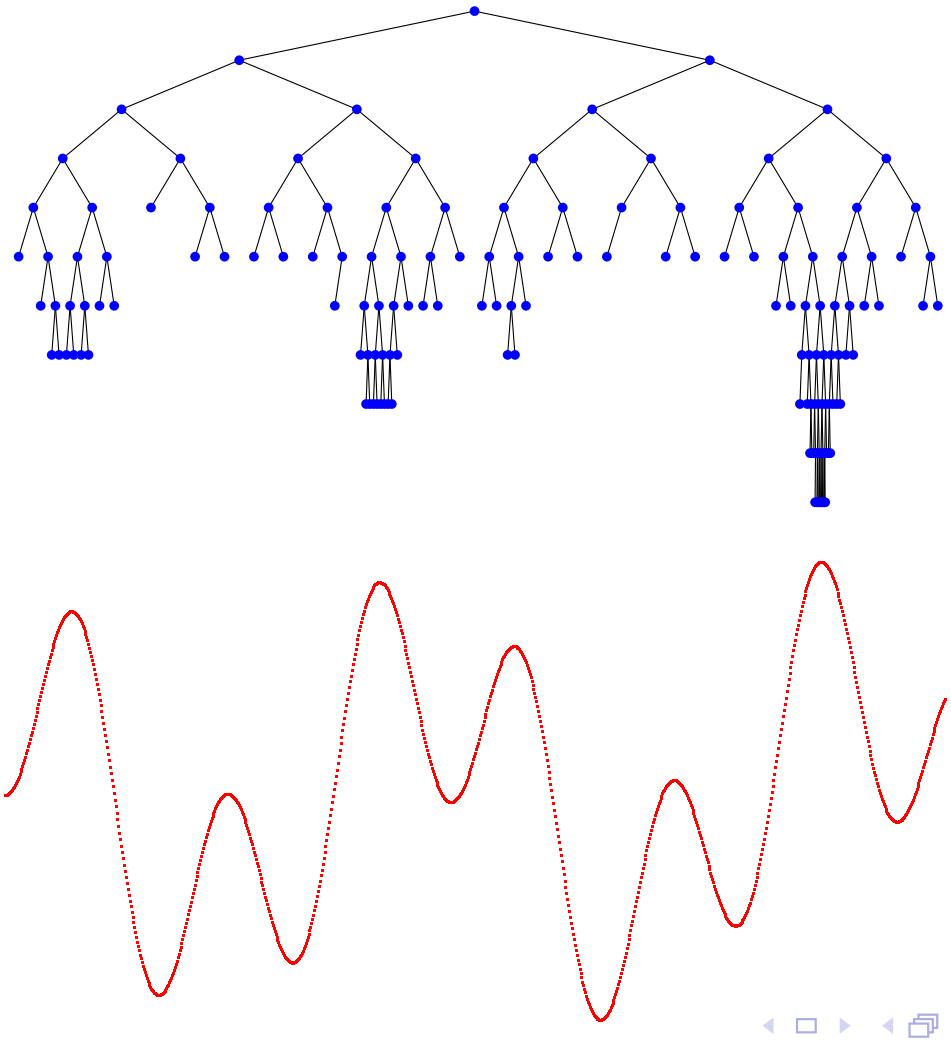
$f(x) = \frac{1}{2}(\sin(13x)\sin(27x) + 1)$ satisfies the local smoothness assumption $f(x) \geq f(x^*) - \ell(x, x^*)$ with

- $\ell_1(x, y) = 14|x - y|$ (i.e., f is globally Lipschitz),
- $\ell_2(x, y) = 222|x - y|^2$ (i.e., f is locally quadratic).



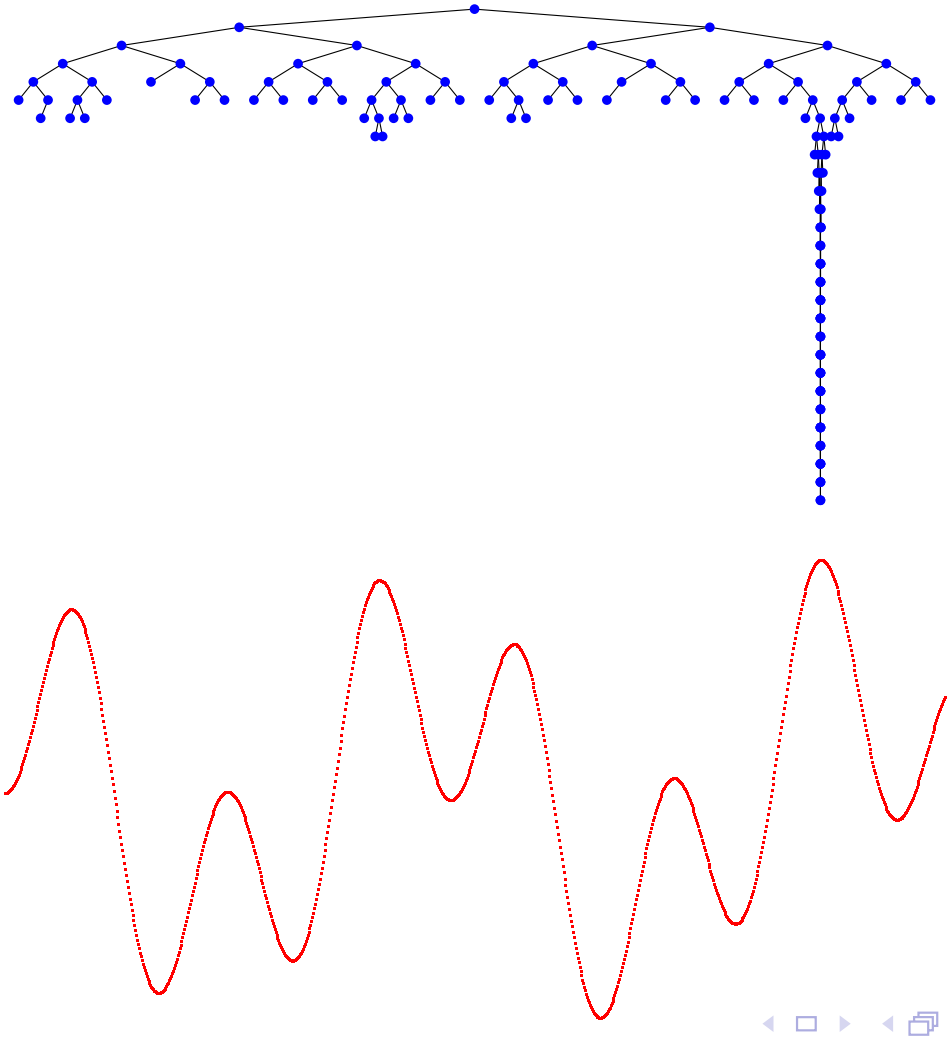
Experiments [2]

Using $\ell_1(x, y) = 14|x - y|$ (i.e., f is globally Lipschitz). $n = 150$.



Experiments [3]

Using $\ell_2(x, y) = 222|x - y|^2$ (i.e., f is locally quadratic). $n = 150$.



Experiments [4]

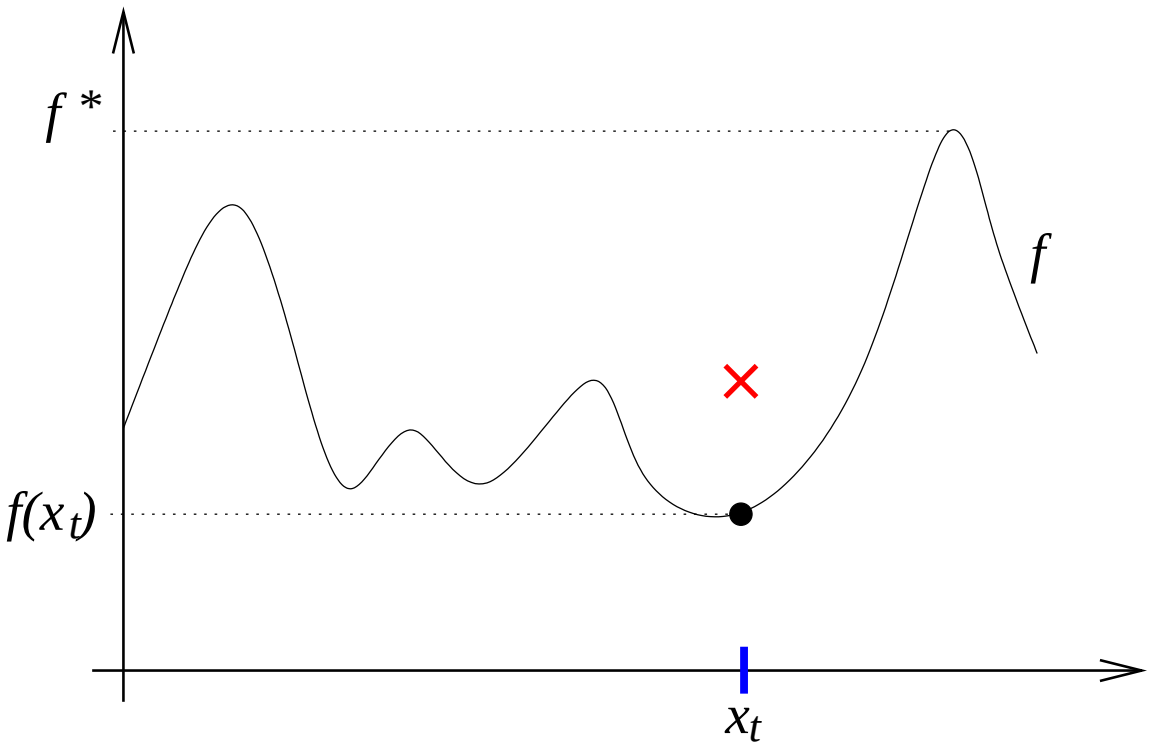
n	uniform grid	DOO with ℓ_1 ($d = 1/2$)	DOO with ℓ_2 ($d = 0$)
50	1.25×10^{-2}	2.53×10^{-5}	1.20×10^{-2}
100	8.31×10^{-3}	2.53×10^{-5}	1.67×10^{-7}
150	9.72×10^{-3}	4.93×10^{-6}	4.44×10^{-16}

Loss r_n for different values of n for a uniform grid and DOO with the two semi-metric ℓ_1 and ℓ_2.

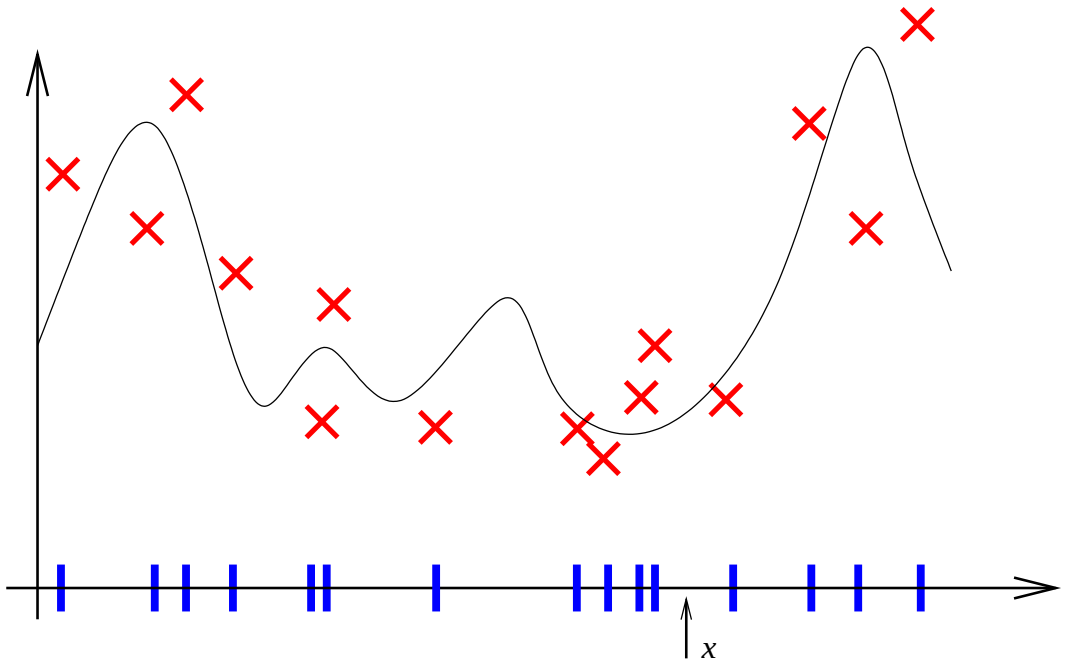
How to handle noise?

The evaluation of f at x_t is perturbed by noise:

$$r_t = f(x_t) + \epsilon_t, \text{ with } \mathbb{E}[\epsilon_t | x_t] = 0.$$

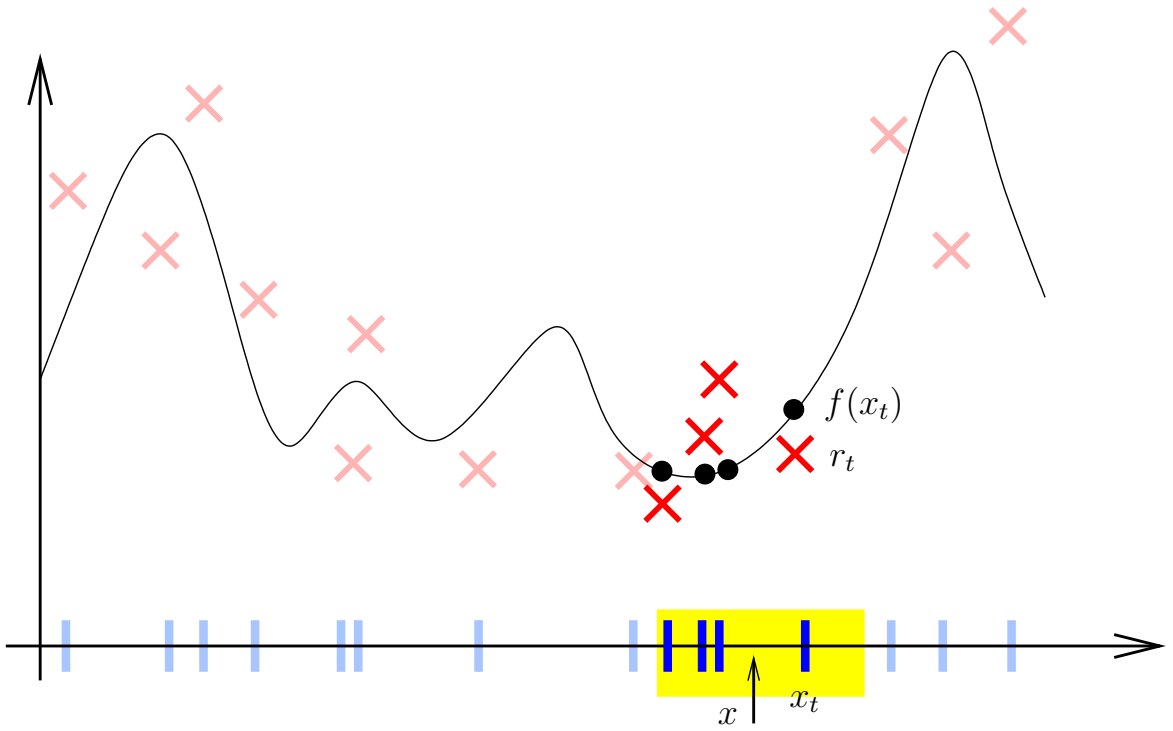


Where should one sample next?



How to define a high probability upper bound at any state x?

UCB in a given cell



For a fixed domain $X_i \ni x$ containing T_i points $\{x_t\} \in X_i$, we have that $\sum_{t=1}^{T_i} r_t - f(x_t)$ is a Martingale. Thus by Azuma's inequality, with $1/n$-confidence,

$$\frac{1}{T_i}\sum_{t=1}^{T_i} r_t + \sqrt{\frac{\log n}{2T_i}} \geq \frac{1}{T_i}\sum_{t=1}^{T_i} f(x_t).$$

UCB in a given cell

For an optimal cell $X_i \ni x^*$ containing T_i points $\{x_t\} \in X_i$,

- we have

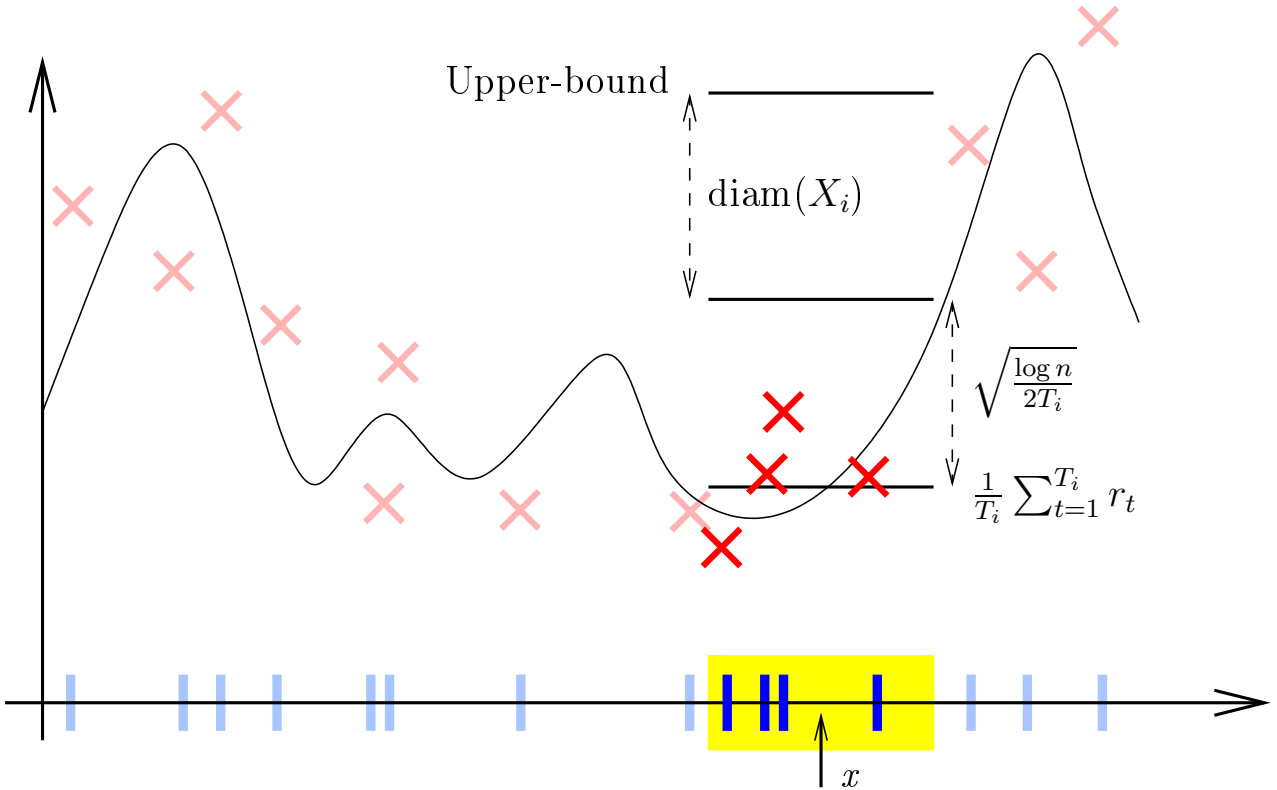
$$\frac{1}{T_i}\sum_{t=1}^{T_i} f(x_t) \geq f(x^*) - \frac{1}{T_i}\sum_{t=1}^{T_i} \ell(x_t, x^*) \geq f(x^*) - \mathrm{diam}(X_i).$$

  (where we used $f(x_t) \geq f(x^*) - \ell(x_t, x^*)$)

- Thus

$$\frac{1}{T_i}\sum_{t=1}^{T_i} r_t + \sqrt{\frac{\log n}{2T_i}} + \mathrm{diam}(X_i) \geq f(x^*).$$

Upper confidence bound



In any cell X_i define the UCB: $\frac{1}{T_i}\sum_{t=1}^{T_i} r_t + \sqrt{\frac{\log n}{2T_i}} + \text{diam}(X_i)$.

Tradeoff between size of the confidence interval and diameter.

Considering several domains we can derive a tighter UCB

Optimistic principle for $\mathcal{X}$-armed bandits

- Consider a series of partitions $\mathcal{T}_h$ of the domain in cells $\{X_{h,i}\}_i$
- Define a UCB for all cells of each partition

$$B_{h,i} = \widehat{\mu}_{h,i}(t) + \sqrt{\frac{\log t}{2\,T_{h,i}(t)}} + \mathrm{diam}(X_{h,i})$$

- Define tighter UCB function:

$$B(x) = \min_{X_{h,i} \ni x} B_{h,i},$$

- Select the point with highest UCB:

$$x_{t+1} \in \operatorname*{argmax}_x B(x).$$

Hierarchical Optimistic Optimization (HOO)

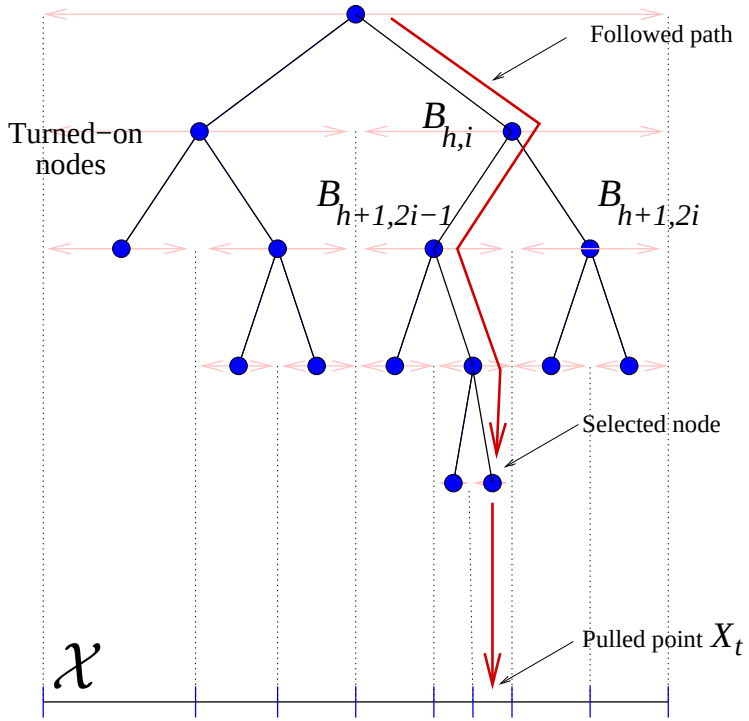
[Bubeck, Munos, Stoltz, Szepesvári, 2008]: Use a tree representation to build a hierarchical partitioning of $\mathcal{X}$

HOO Algorithm:

At round t, tree $\mathcal{T}_t$

- Select a leaf J_t of $\mathcal{T}_t$ by following a path from the root that maximizes the B-values,
- Expand J_t:
- Select $x_t \in X_{J_t}$ (arbitrarily)
- Observe reward $r_t = f(x_t) + \epsilon_t$

and update the B-values:



$$B_i(t) \stackrel{\text{def}}{=} \min\left[\widehat{\mu}_{i,t} + \sqrt{\frac{2\log(t)}{T_i(t)}} + \text{diam}(X_i), \max_{j \in \mathcal{C}(i)} B_j(t)\right]$$

Properties of HOO

- HOO selects a leaf J_t such that

$$J_t \in \operatorname*{argmax}_{j \in \mathcal{L}_t} \min_{i \in \mathcal{P}(j)} \left[\widehat{\mu}_{i,t} + \sqrt{\frac{2\log(t)}{T_i(t)}} + \operatorname{diam}(X_i) \right]$$

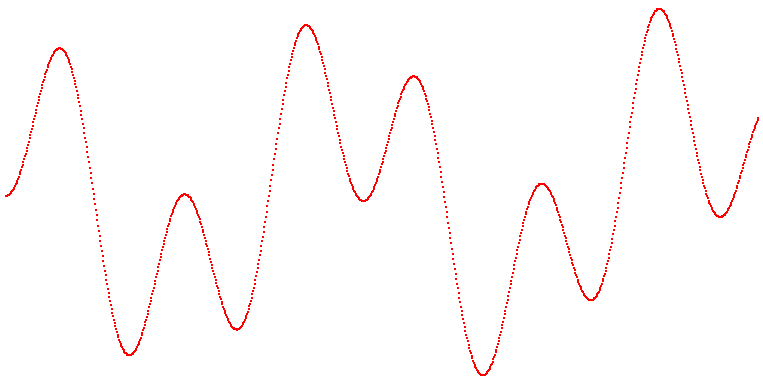
- For any $X_i \ni x^*$, $B_i(t)$ is a h.p. UCB on $f(x^*)$.
- Thus any suboptimal node X_i such that

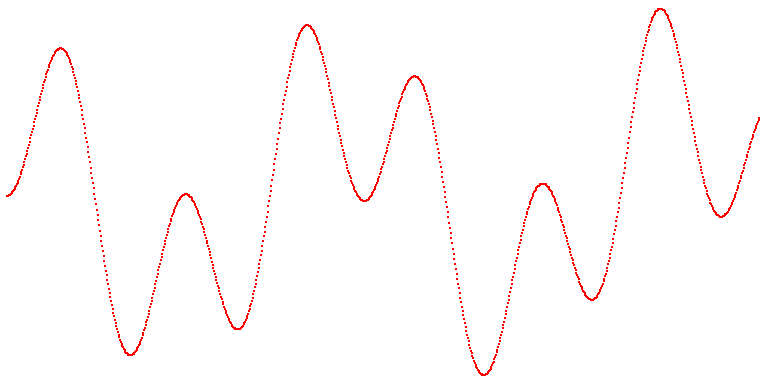
$$\sup_{x \in X_i} f(x) + \sqrt{\frac{2\log(t)}{T_i(t)}} + \operatorname{diam}(X_i) < f(x^*)$$

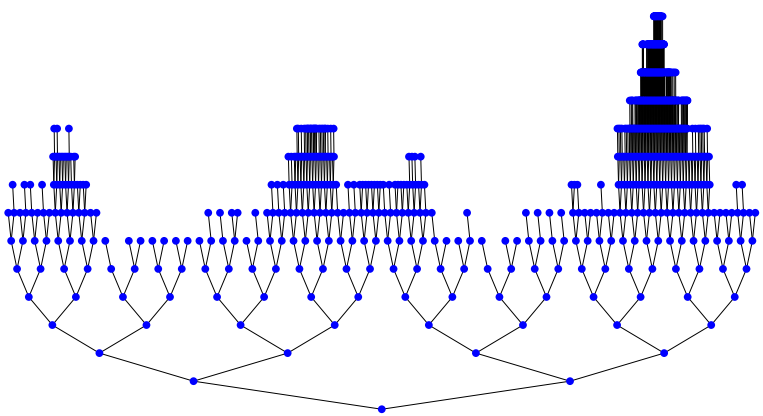
will not be selected.

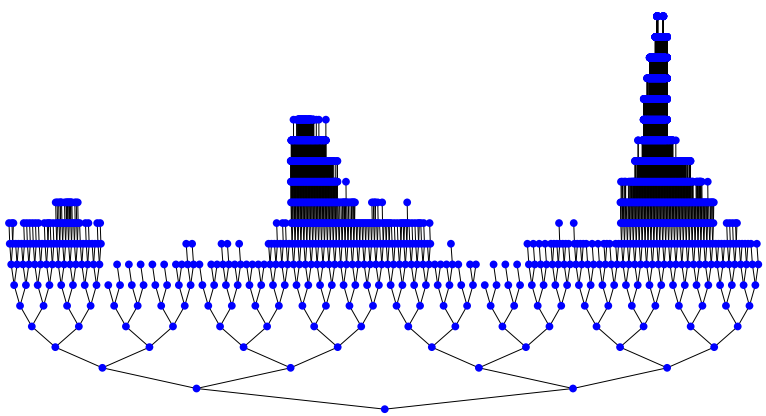
Example in 1d

$r_t \sim \mathcal{B}(f(x_t))$ a Bernoulli distribution with parameter $f(x_t)$









Resulting tree at time $n = 1000$ and at $n = 10000$.

Analysis of HOO

Assuming a slightly stronger assumption on f (weak Lipschitz): $\forall x, y \in \mathcal{X}$,

$$f(y) - f(x) \leq \max\{f^* - f(y), \ell(x, y)\}$$

Theorem 3.

Assume that the diameters decrease exponentially fast with their depth h. The loss of HOO is

$$r_n = O\left(\left[\frac{n}{\log n}\right]^{-\frac{1}{d+2}}\right)$$

(recall that for deterministic rewards $r_n = O(n^{-1/d})$ for $d > 0$)
(see also the Zooming algorithm [Kleinberg, Slivkins, Upfal, 2008]).

Example

Let $\mathcal{X} = [0,1]^D$. Assume that f is locally α-smooth around x^*, i.e.

$$f(x^*) - f(x) = \Theta(||x^* - x||^\alpha).$$

Choose $\ell(x,y) = ||x-y||^\beta$.

- **If the smoothness of the function is known** ($\alpha = \beta$): the loss of HOO is $O(\sqrt{\log n / n})$. The rate is **independent of the dimension**.
- **The smoothness is underestimated** ($\alpha > \beta$): $d = D(1/\beta - 1/\alpha)$ and the loss is $\widetilde{O}(n^{-1/(d+2)})$
- **The smoothness is overestimated** ($\alpha < \beta$): the weak-Lipschitz assumption is violated, thus there is no finite-time guarantee (e.g., UCT = HOO with $\beta = \infty$)

Cumulative regret versus loss

The **cumulative regret** is defined as: $R_n = \sum_{t=1}^{n} f(x^*) - f(x_t)$,

whereas the **loss** (or **simple regret**) is $r_n = f(x^*) - f(x(n))$.

Properties:

- Any exploration strategy with cumulative regret R_n can be turned into a strategy with expected loss $\mathbb{E}r_n = \frac{1}{n}\mathbb{E}R_n$ using $x(n) = x_N$ where $N \sim \mathcal{U}(\{1, 2, \ldots, n\})$.
- In multi-armed bandits there exists much better strategies for the simple regret [Audibert, Bubeck, Munos, 2010], [Bubeck, Munos, Stoltz, 2009].
- In bandits with many arms, the situation is not so clear...

Assume that the smoothness is unknown

Previous algorithms heavily rely on the knowledge or the local smoothness of the function (i.e. knowledge of the best metric).

Question: When the smoothness is unknown, is it possible to implement the optimistic principle for function optimization?

Some approaches relies on estimating the local or global smoothness of the function [Bubeck, Stoltz, Yu, 2011], [Slivkins, 2011], [Bull, 2013].

DIRECT algorithm

Assume f is Lipschitz but the Lipschitz constant L is unknown.

The DIRECT algorithm [Jones, Perttunen, Stuckman, 1993] expands simultaneously all nodes that may potentially contain the maximum for some value of L.

Illustration of DIRECT

The sin function and its upper bound for $L = 2$.

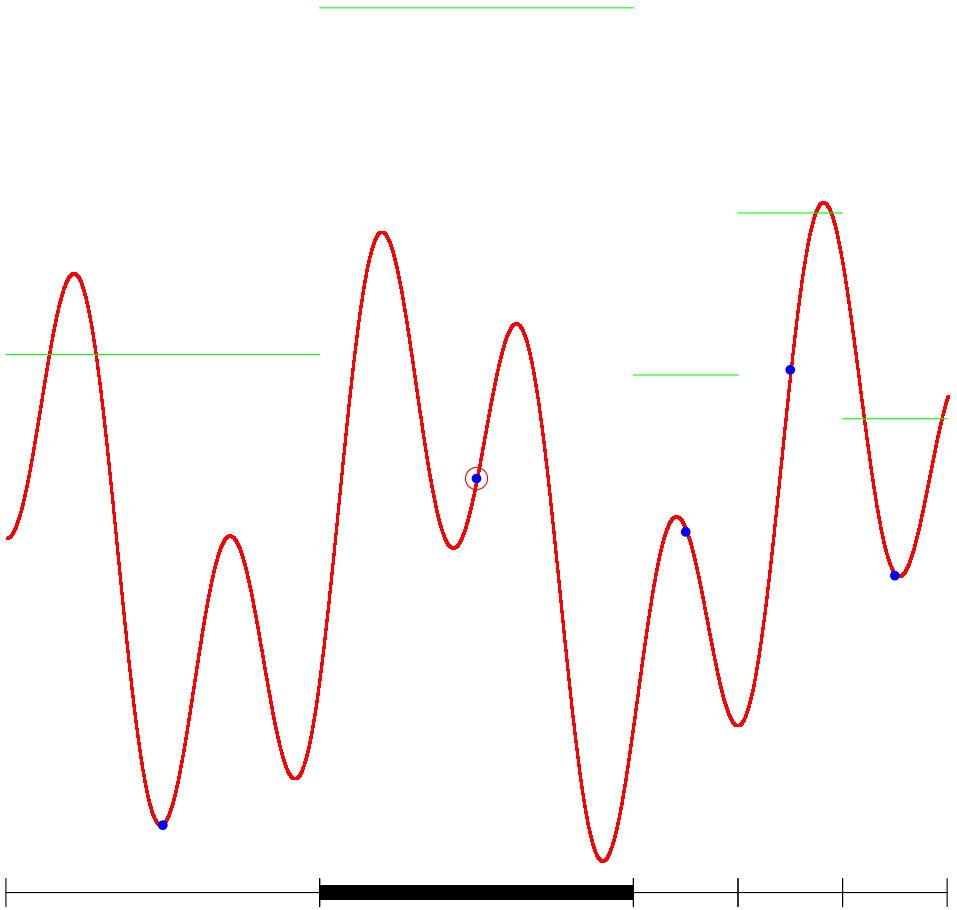
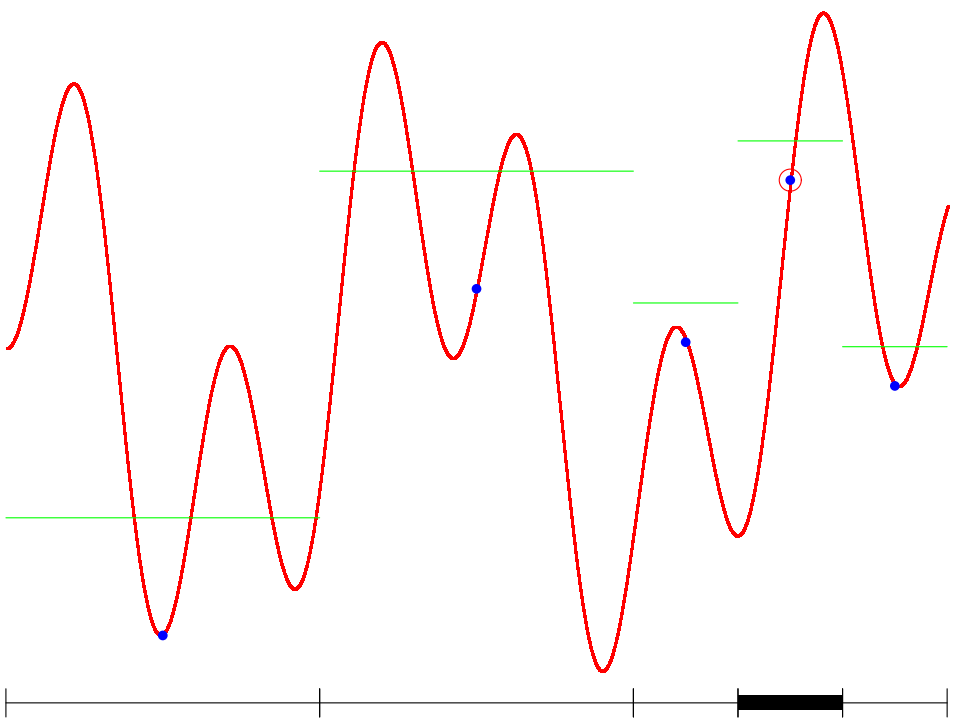
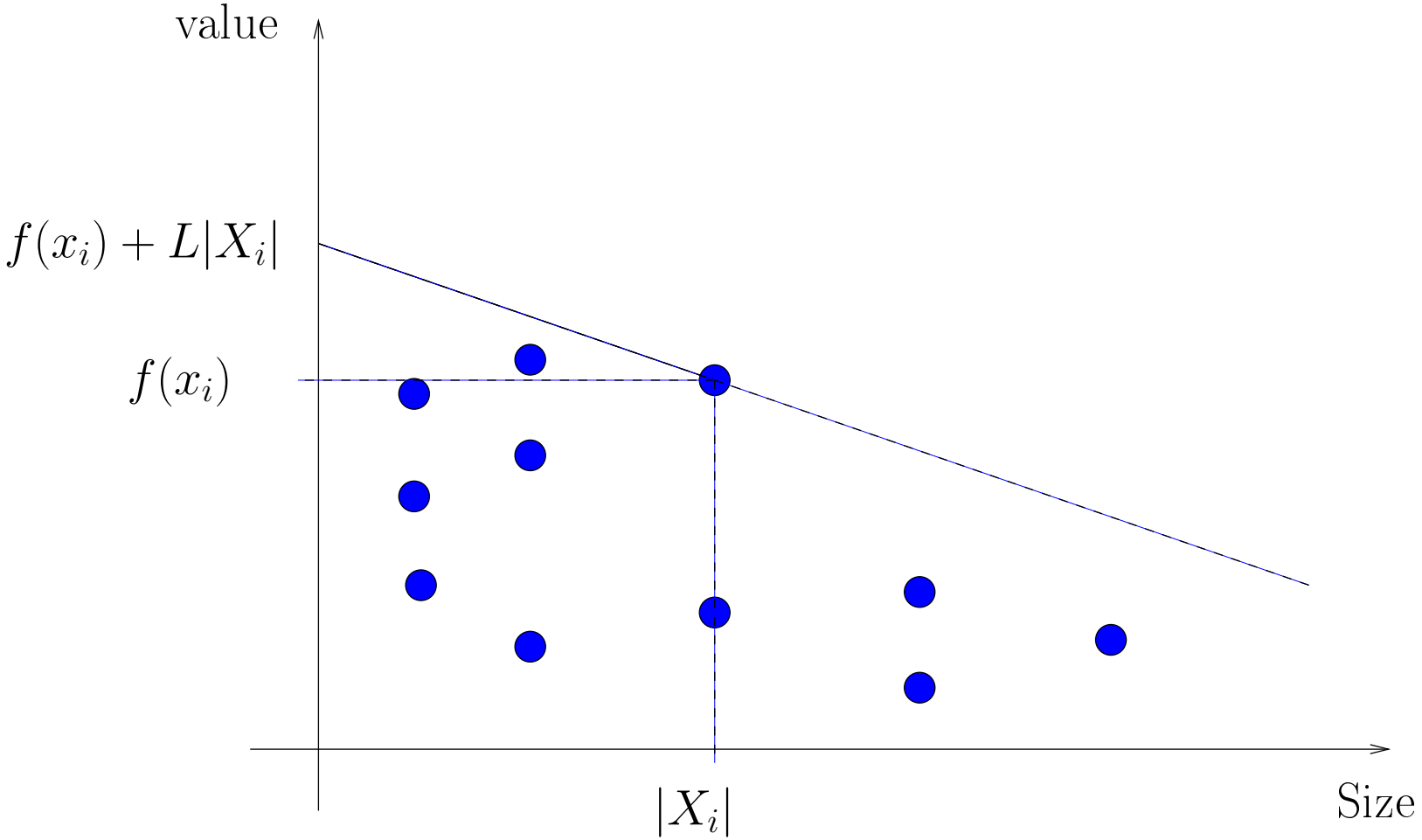


Illustration of DIRECT

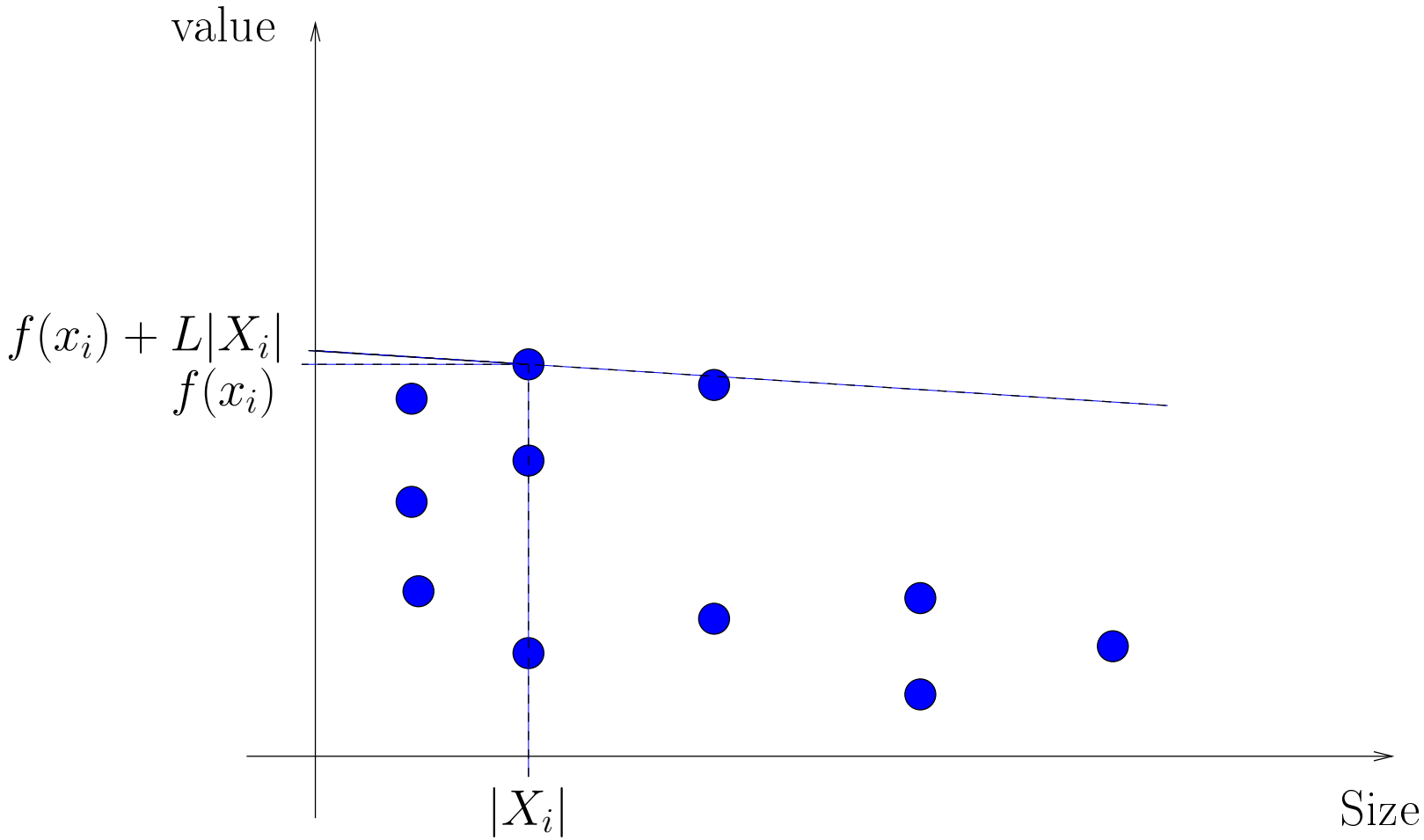
The sin function and its upper bound for $L = 1/2$.



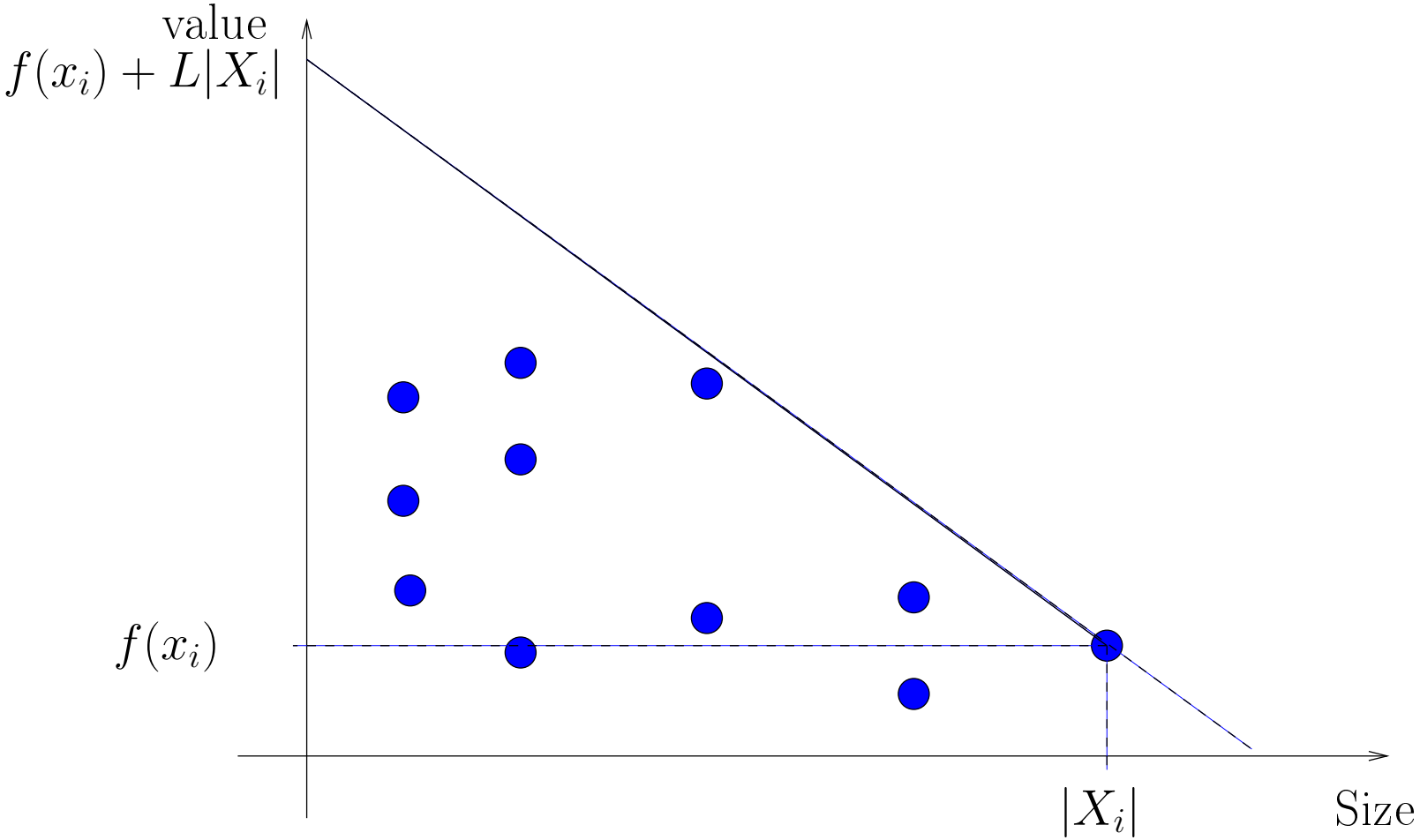
Other representation of DIRECT



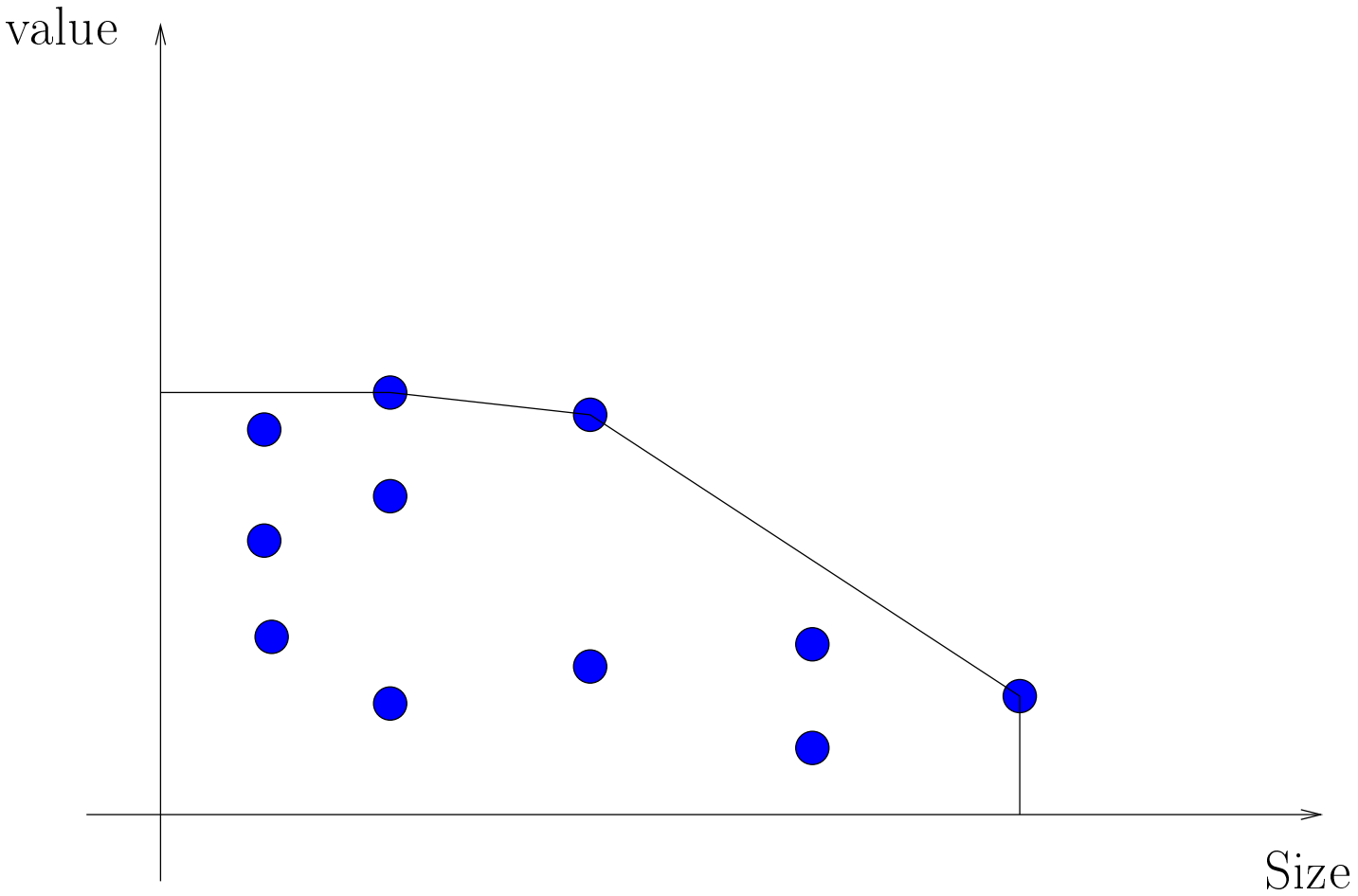
Other representation of DIRECT



Other representation of DIRECT



Other representation of DIRECT

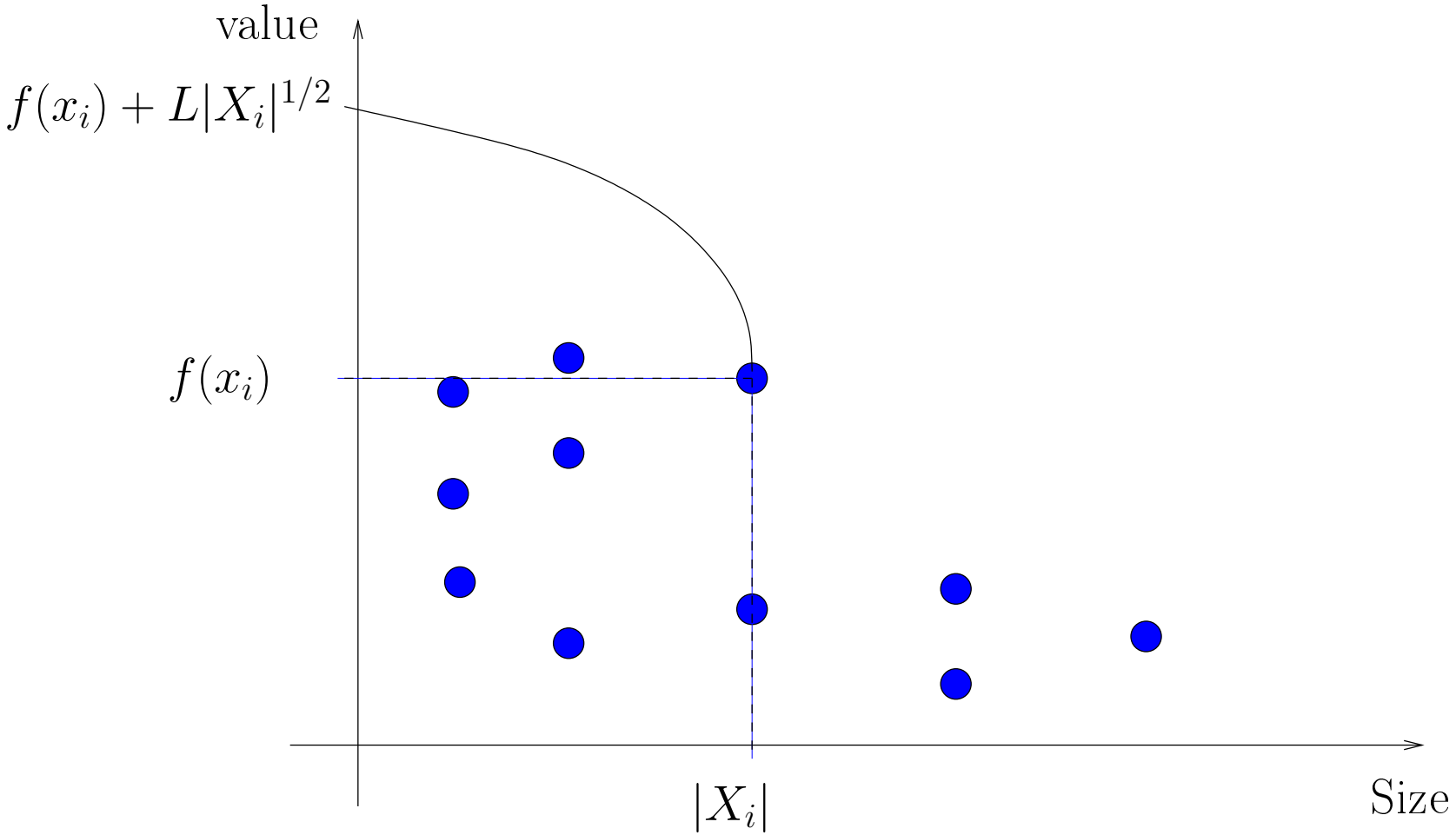


Limitations of DIRECT

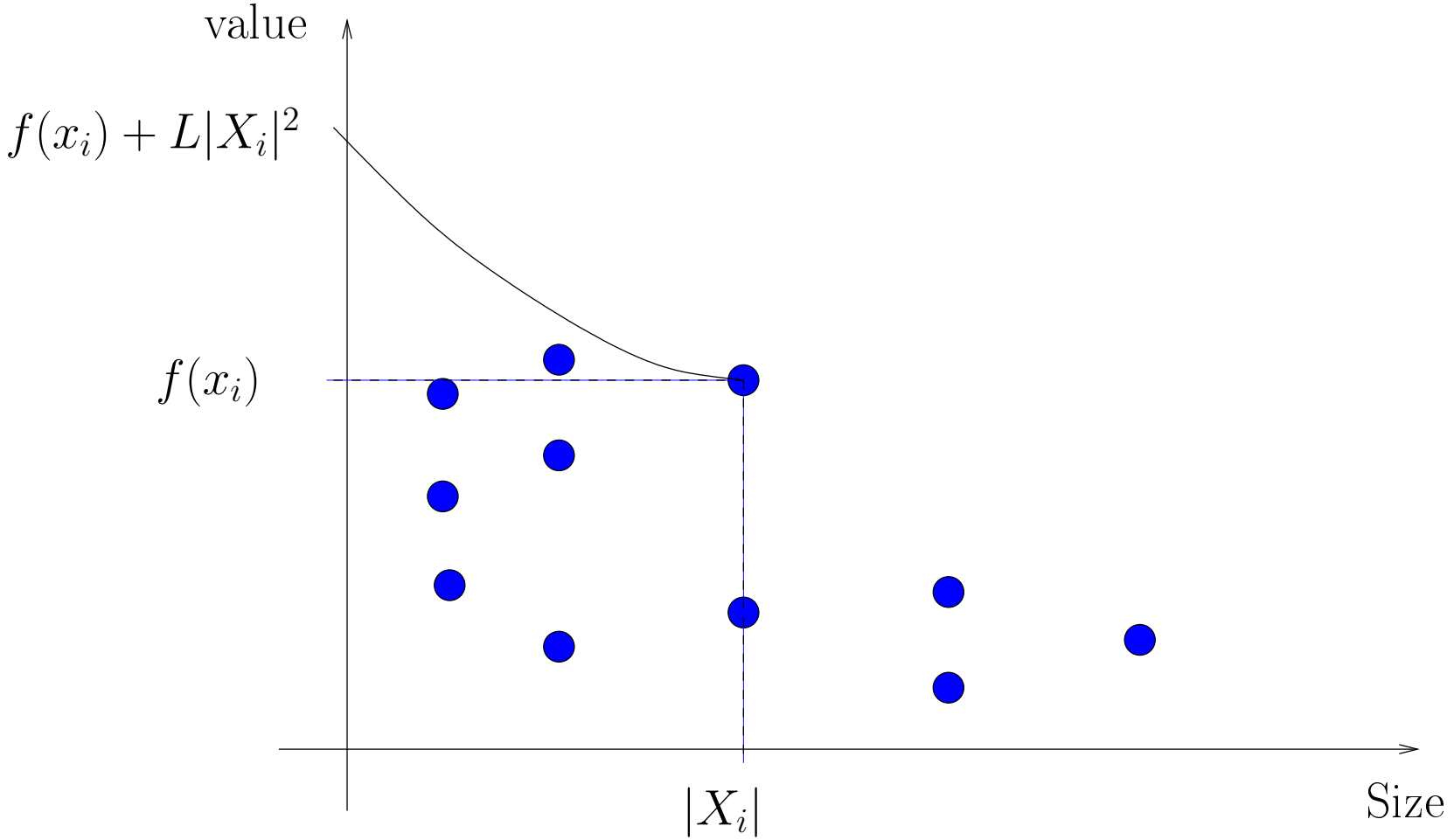
Assuming the function is globally Lipschitz is too restrictive. We would like to handle the general case where:

- where the function is only **locally smooth** w.r.t. ℓ
- for **any semi-metric** ℓ

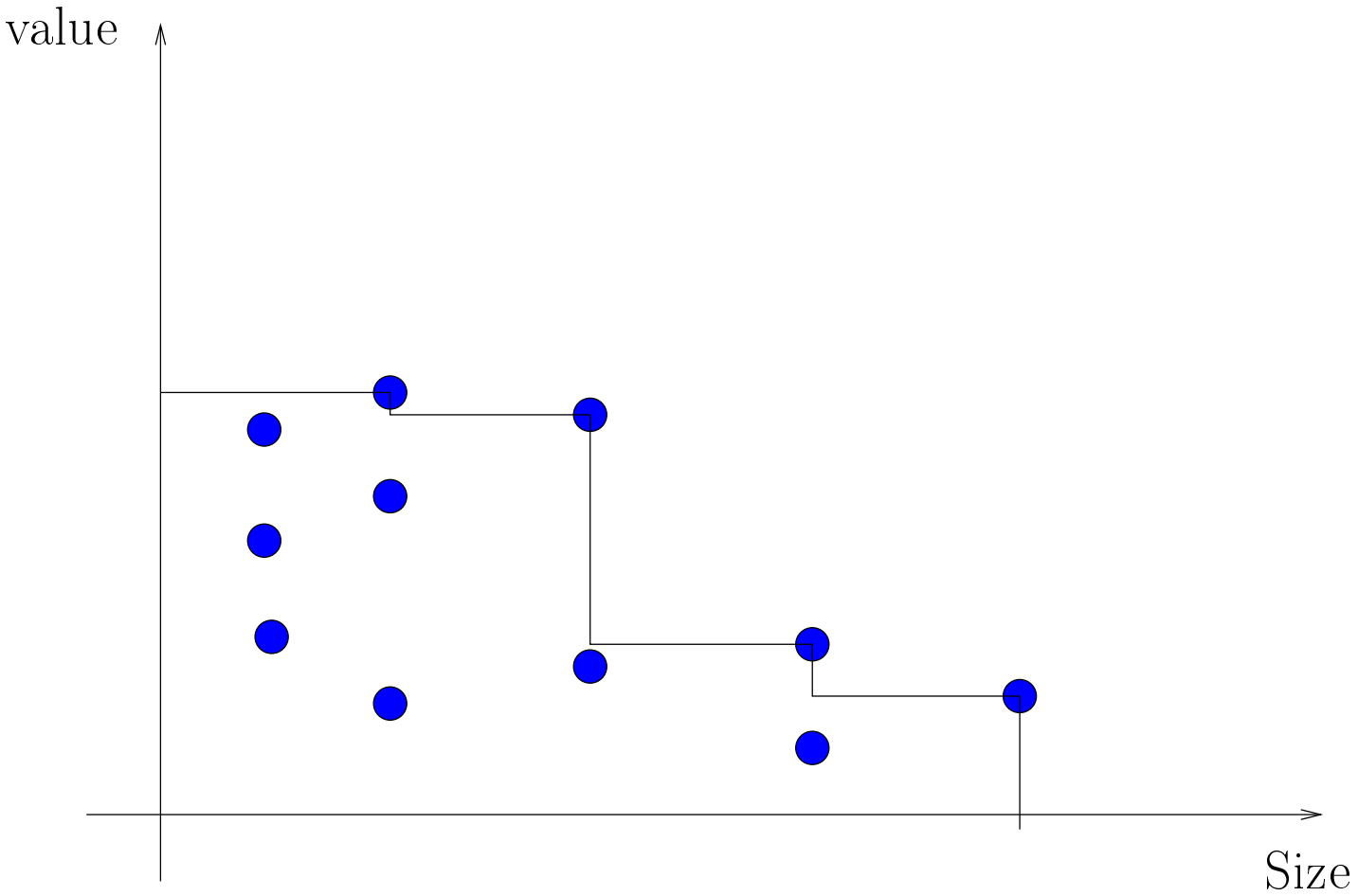
Extension to $\ell(x, y) = L\|x - y\|^{1/2}$



Extension to $\ell(x, y) = L\|x - y\|^2$



Extension to any ℓ!



Simultaneous Optimistic Optimization (SOO)

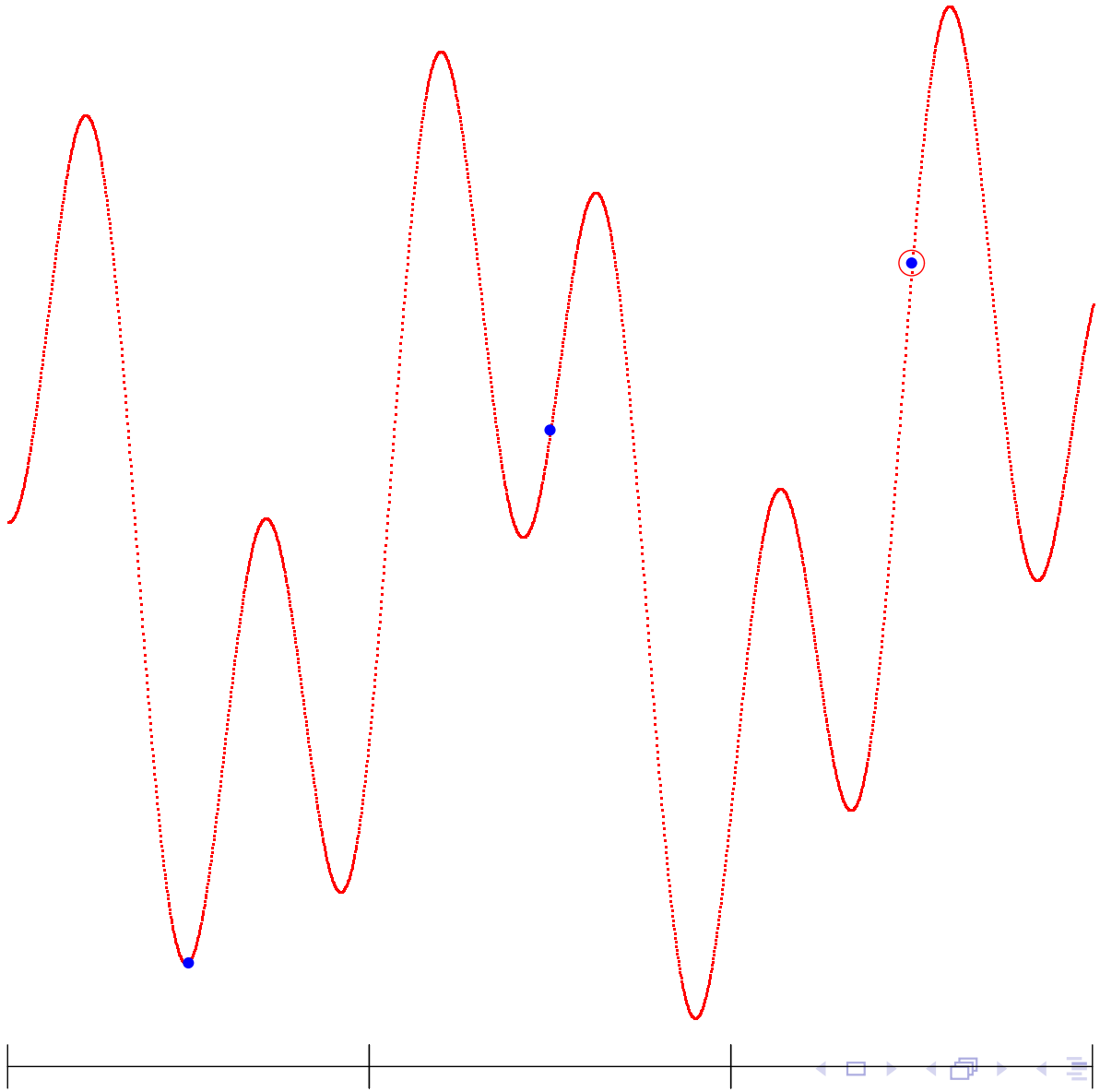
[Munos, 2011]

- Expand several leaves simultaneously
- SOO expands at most one leaf per depth
- SOO expands a leaf only if its value is larger that the value of all leaves of same or lower depths.
- At round t, SOO does not expand leaves with depth larger than $h_{\max}(t)$

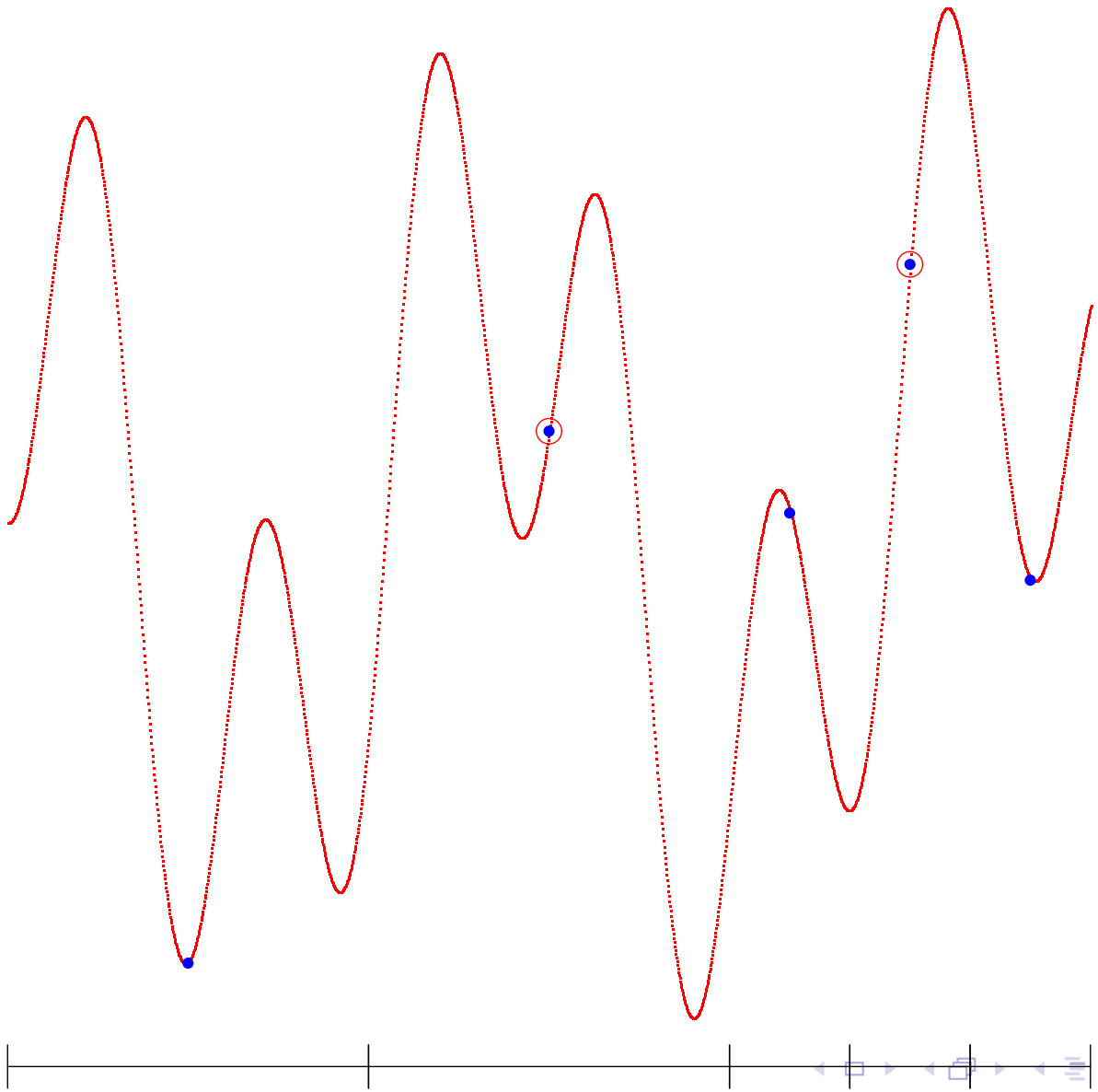
SOO algorithm

Input: the maximum depth function $t \mapsto h_{\max}(t)$
Initialization: $\mathcal{T}_1 = \{(0,0)\}$ (root node). Set $t = 1$.
while True **do**
  Set $v_{\max} = -\infty$.
  for $h = 0$ to $\min(\text{depth}(\mathcal{T}_t), h_{\max}(t))$ **do**
    Select the leaf $(h,j) \in \mathcal{L}_t$ of depth h with max $f(x_{h,j})$ value
    if $f(x_{h,i}) > v_{\max}$ **then**
      Expand the node (h,i), Set $v_{\max} = f(x_{h,i})$, Set $t = t+1$
      if $t = n$ **then** return $x(n) = \arg\max_{(h,i)\in\mathcal{T}_n} x_{h,i}$
    end if
  end for
end while.

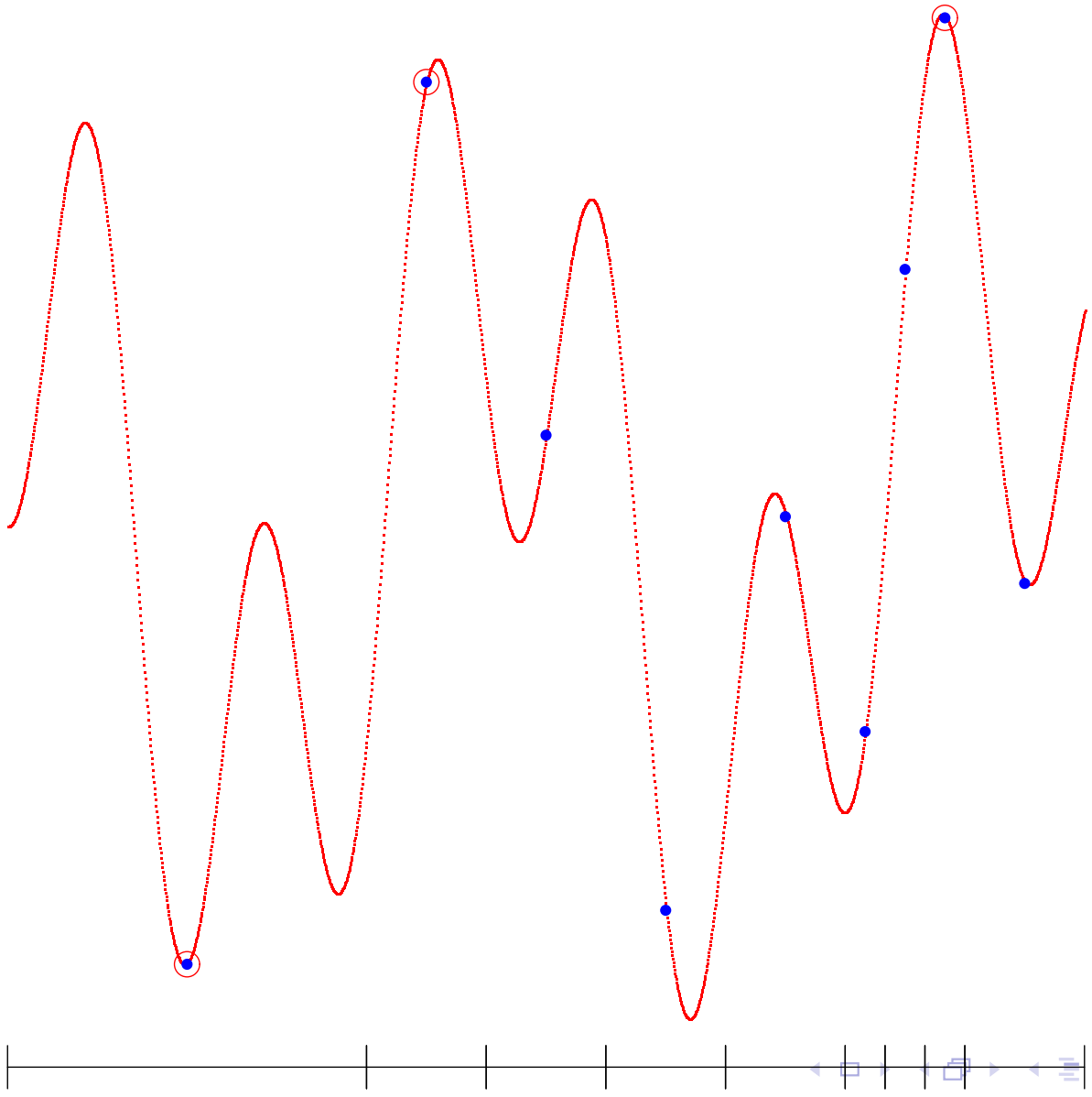
SOO



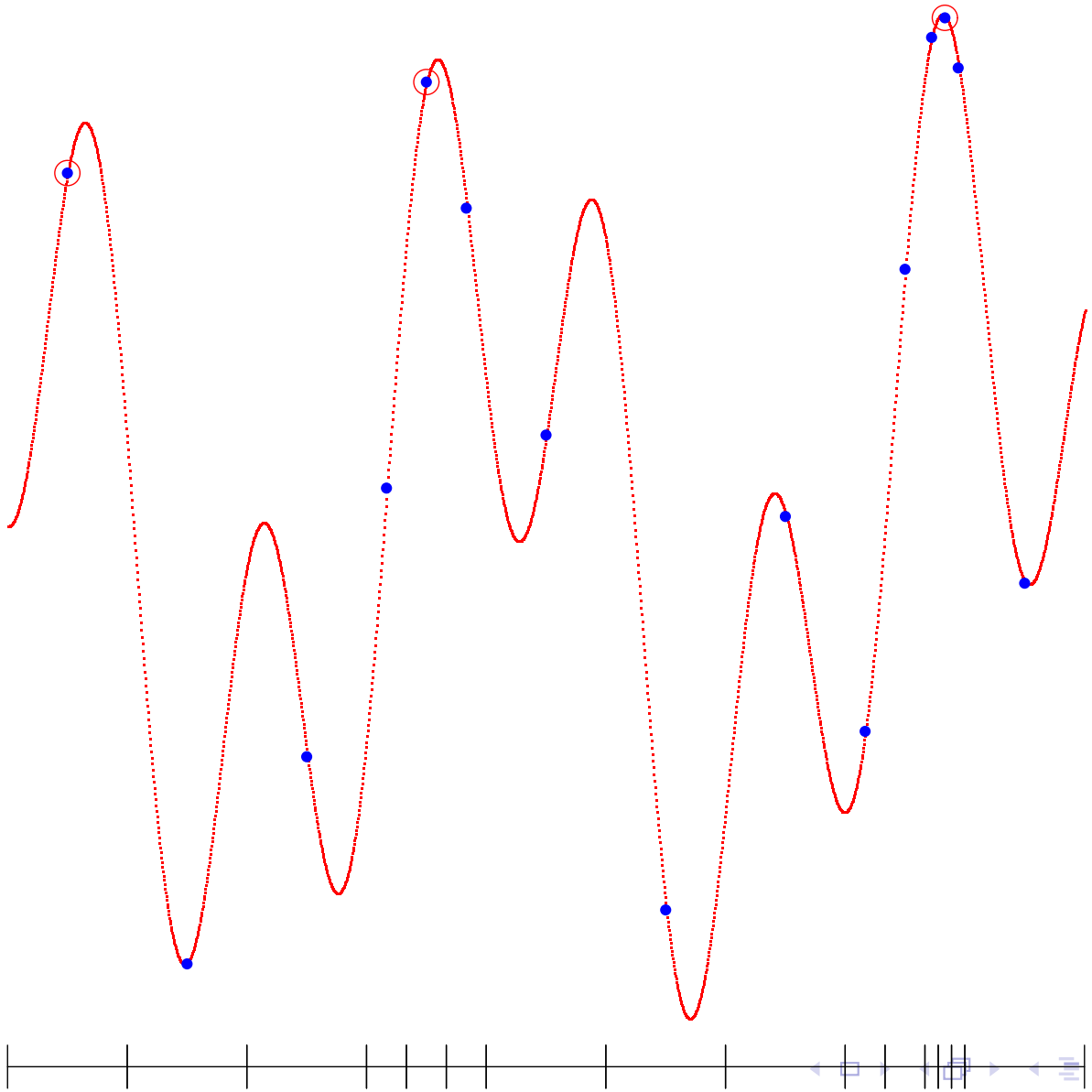
SOO



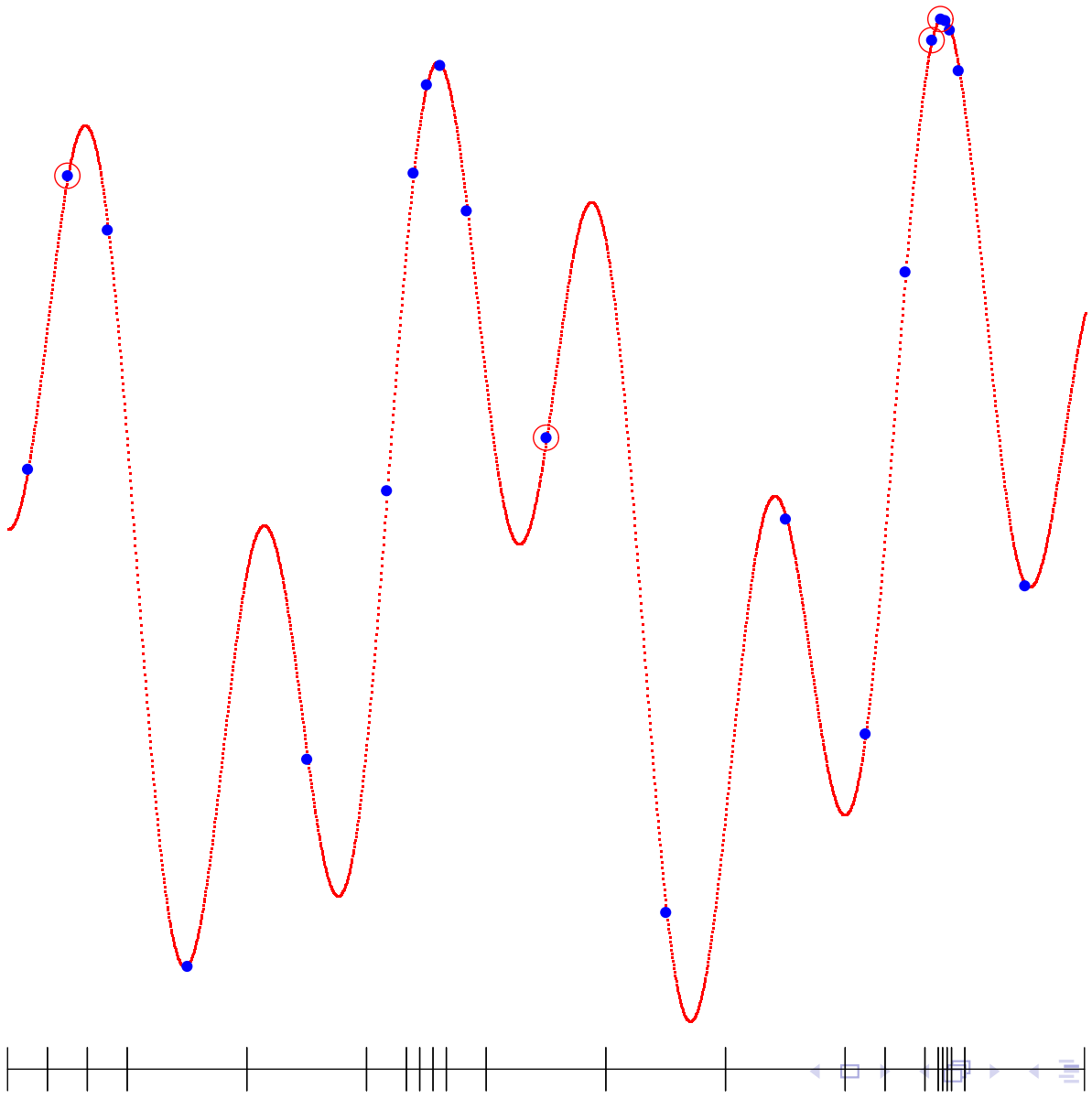
SOO



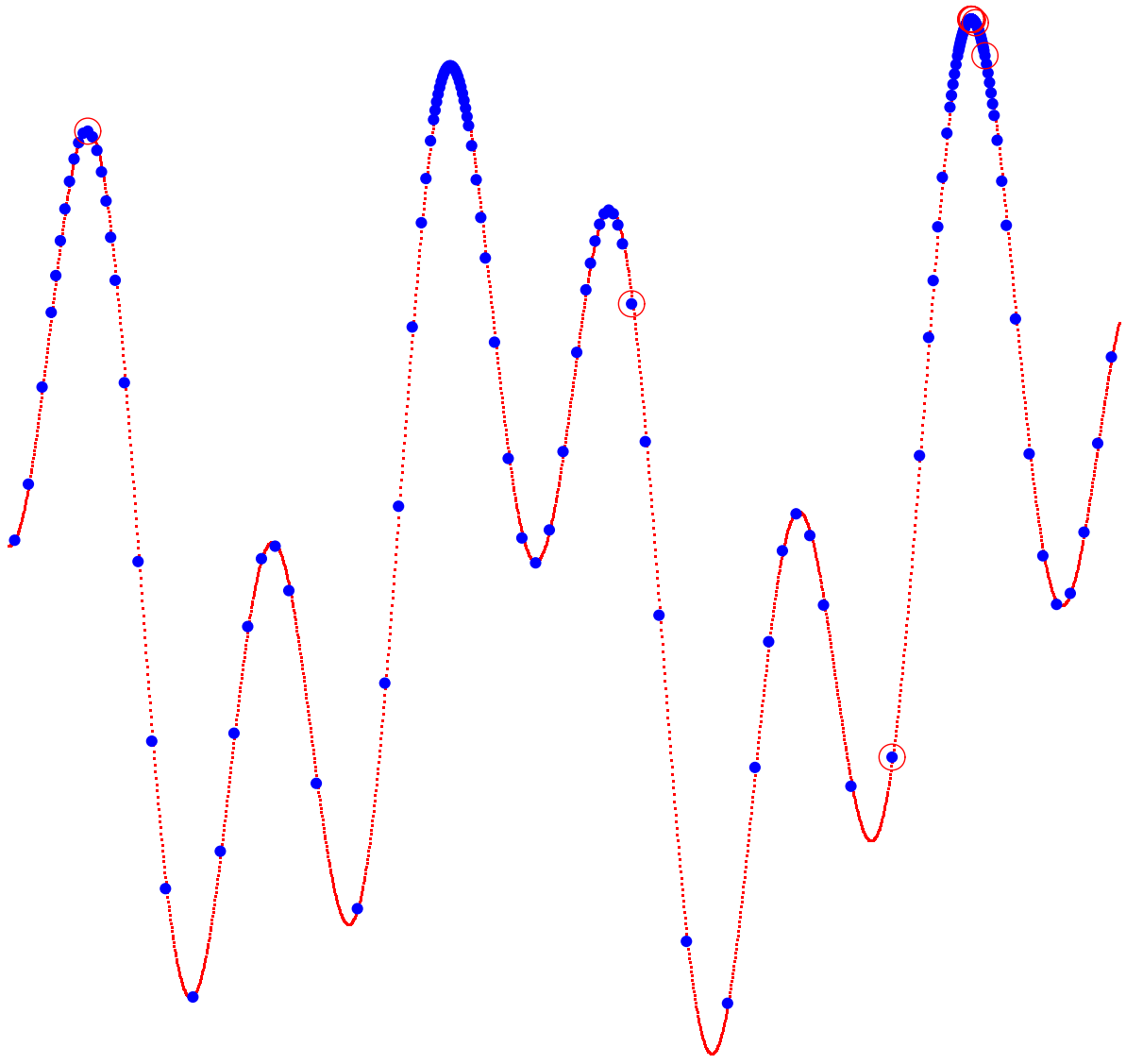
SOO



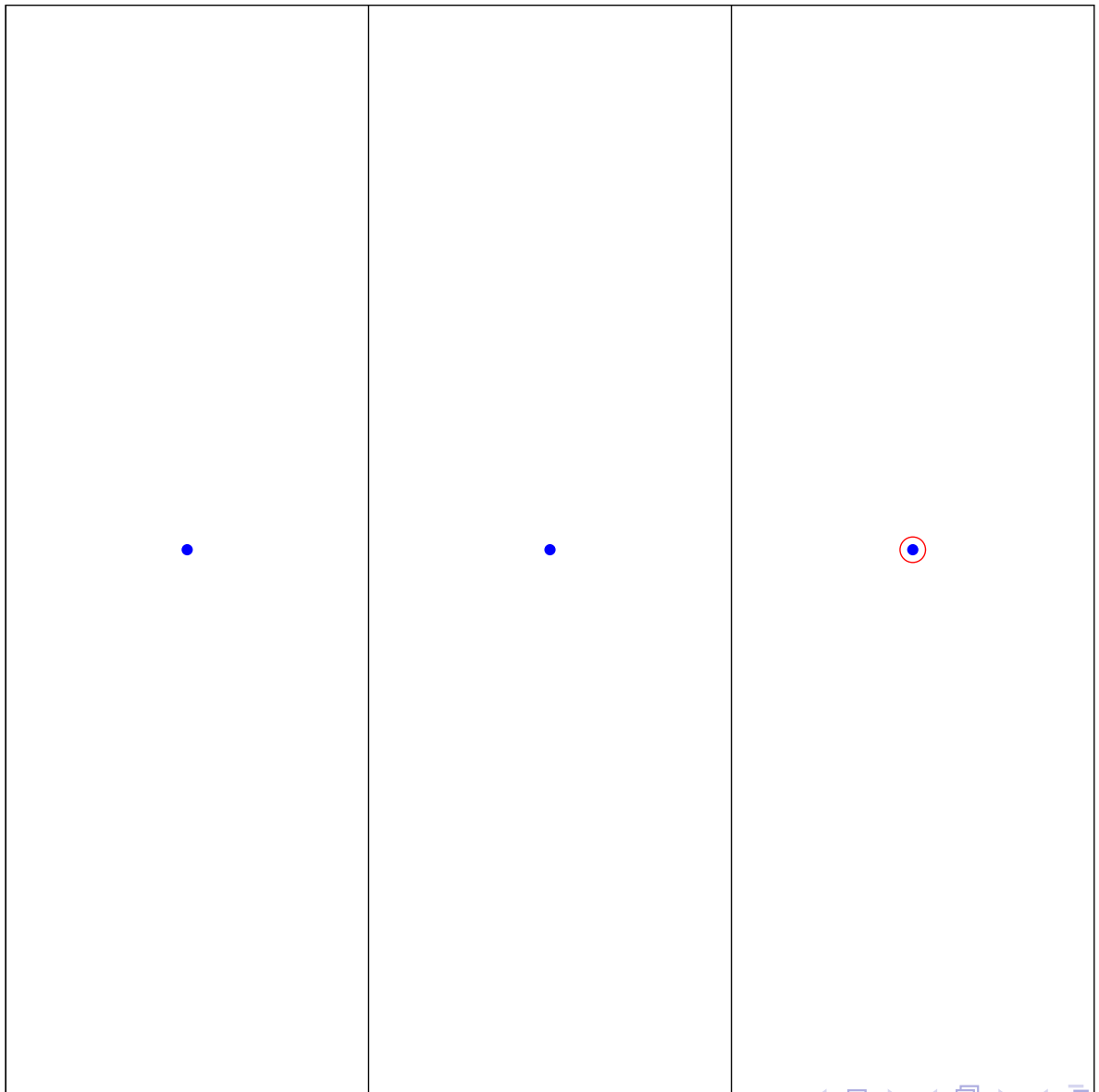
SOO



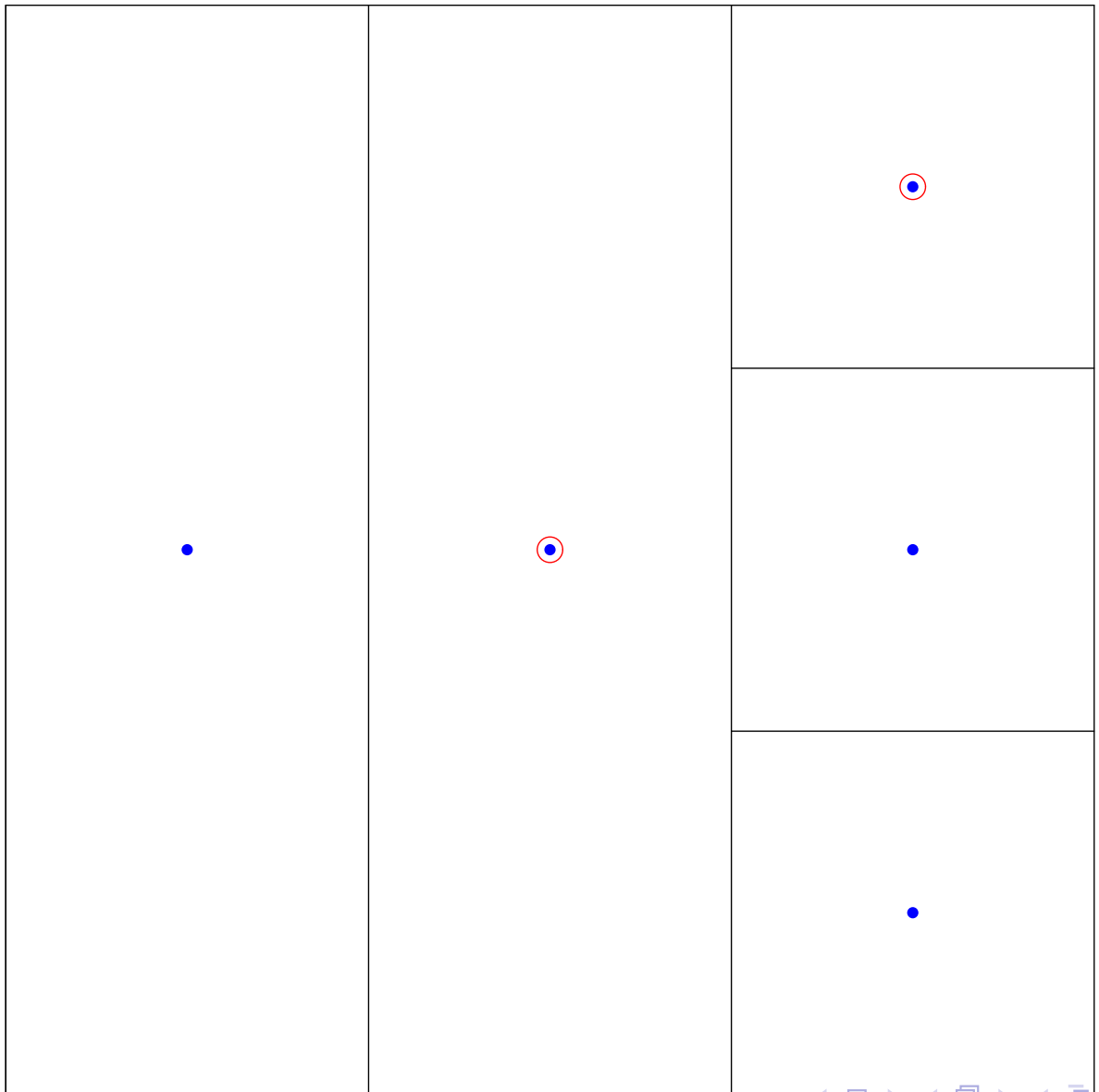
SOO



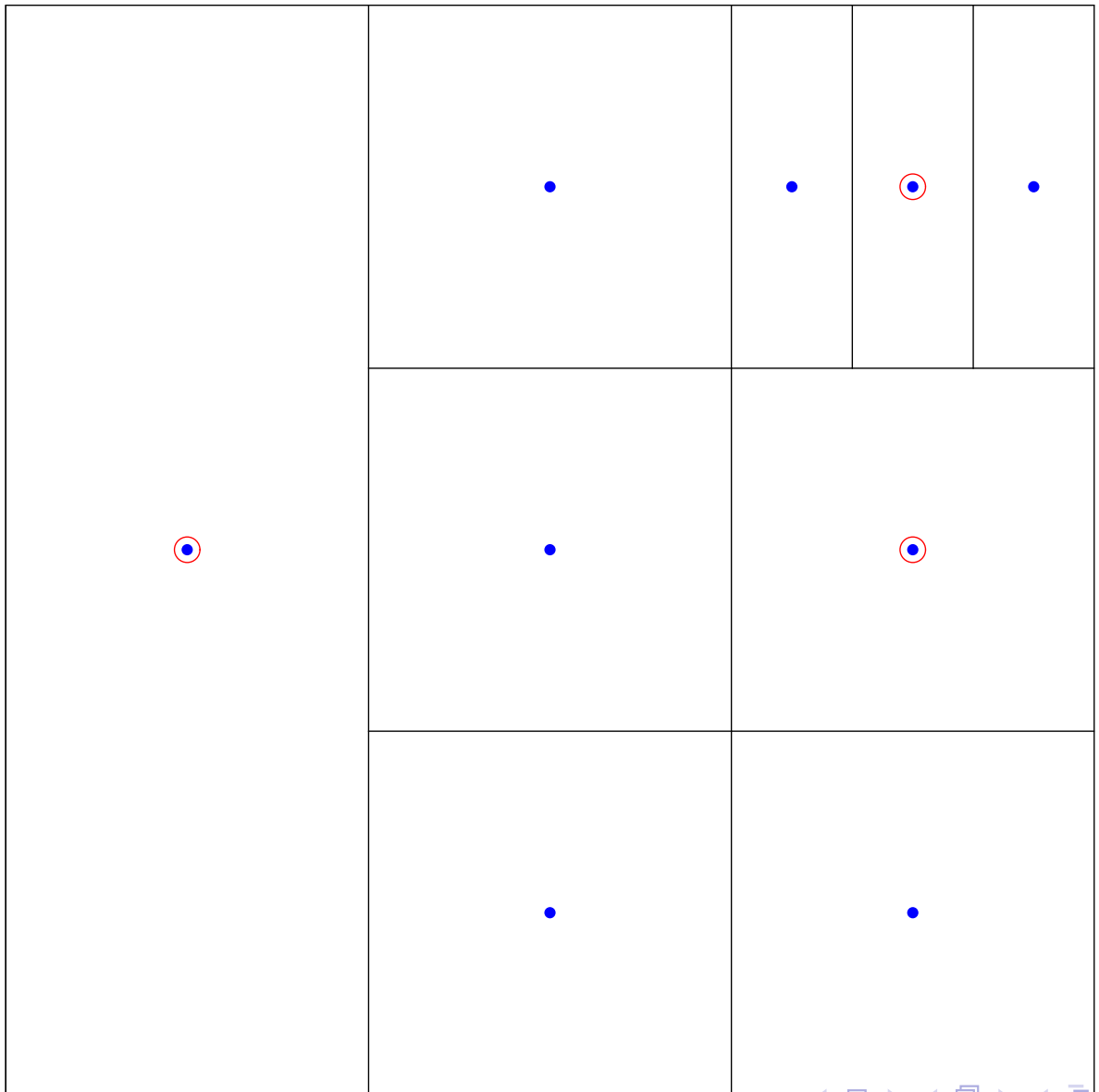
SOO



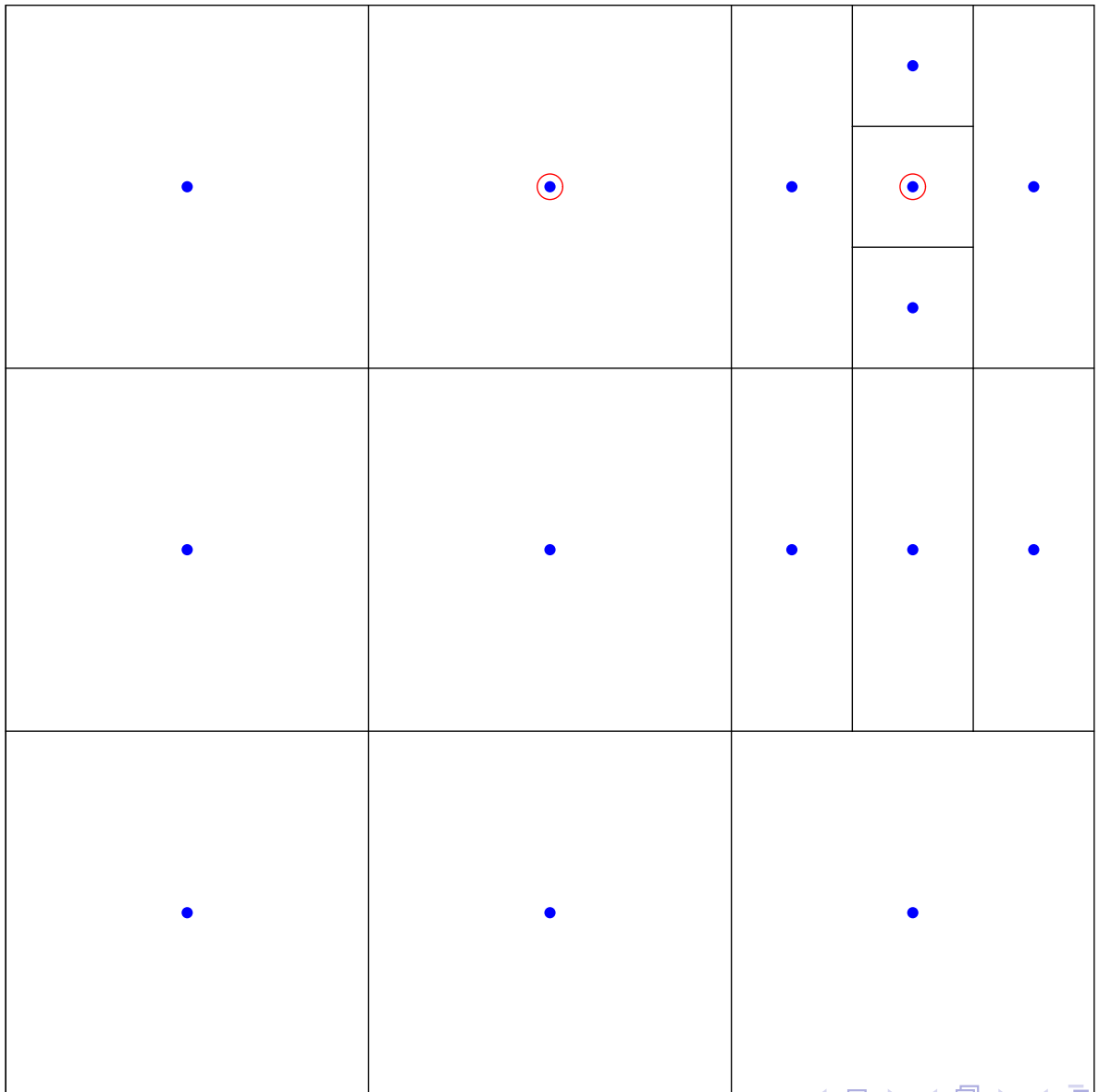
SOO



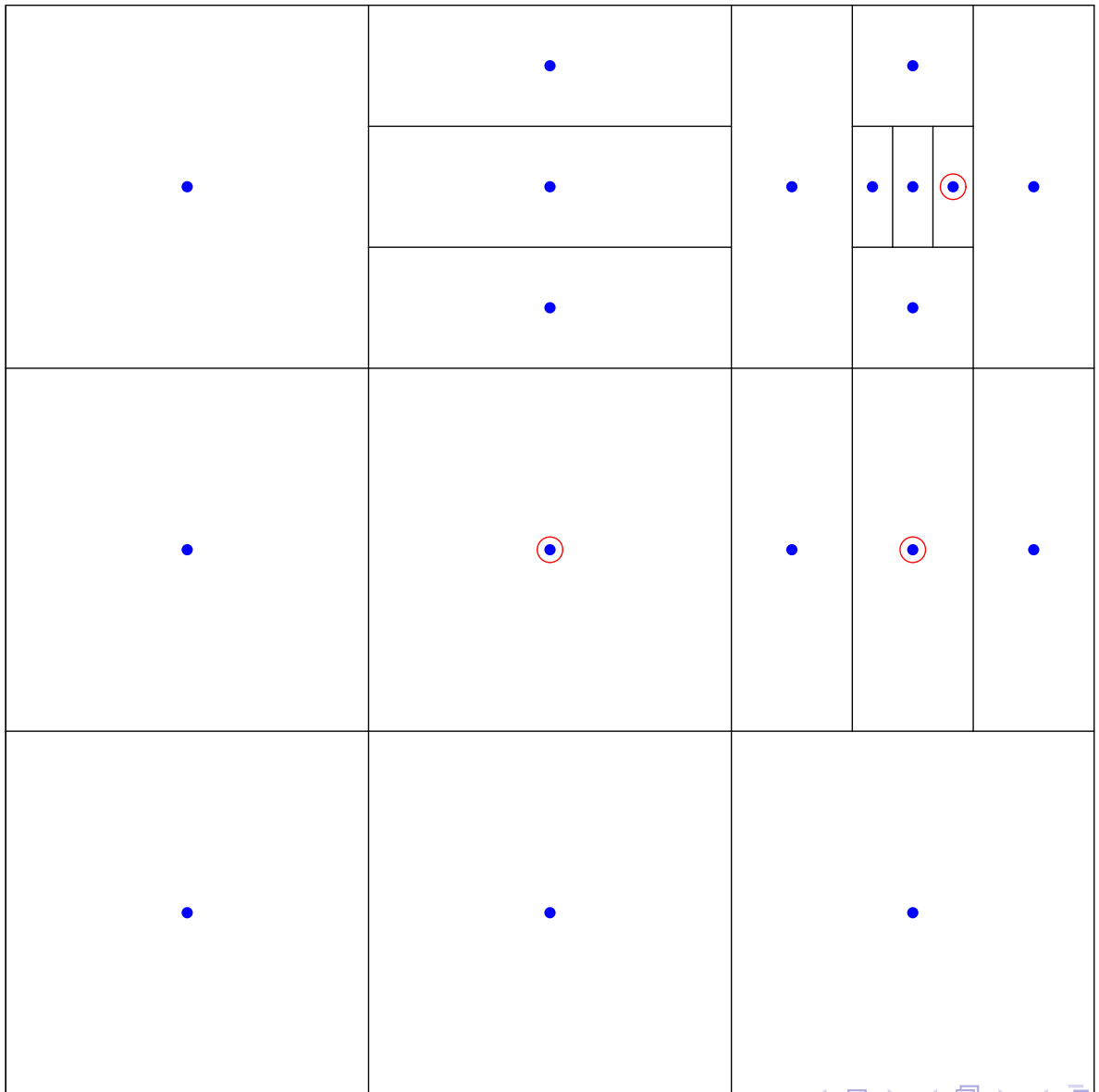
SOO



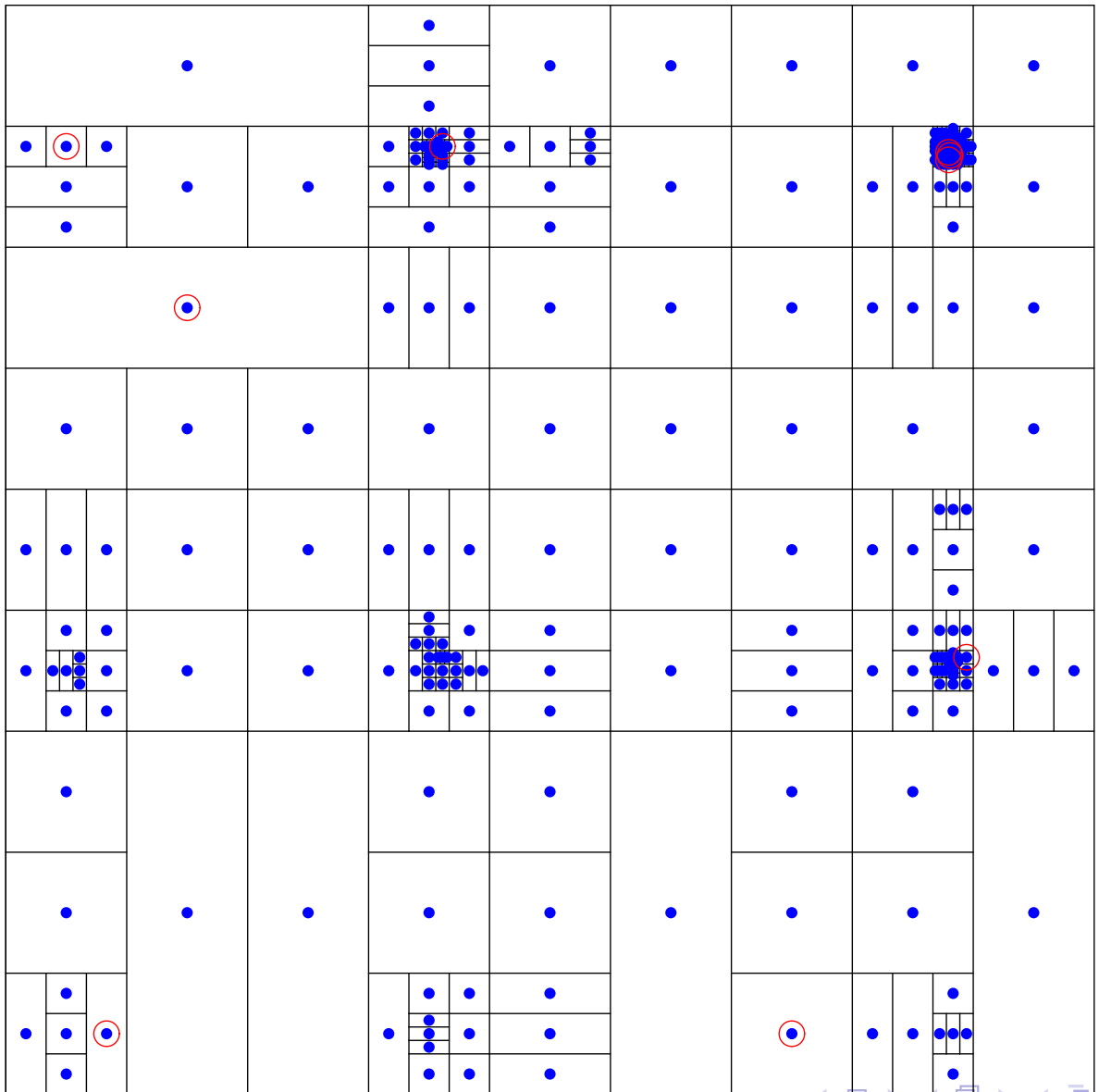
SOO



SOO



SOO



Performance of SOO

Theorem 4.

For any semi-metric ℓ such that

- *f is locally smooth w.r.t. ℓ*
- *The ℓ-diameter of cells of depth h is $c\gamma^h$*
- *The near-optimality dimension of f w.r.t. ℓ is $d = 0$,*

by choosing $h_{\max}(n) = \sqrt{n}$, the expected loss of SOO is

$$r_n \leq c\gamma^{\sqrt{n}/C - 1}$$

In the case $d > 0$ a similar statement holds with $\mathbb{E}r_n = \widetilde{O}(n^{-1/d})$.

Performance of SOO

Remarks:

- Since the algorithm does not depend on ℓ, the analysis holds for the best possible choice of the semi-metric ℓ satisfying the assumptions.
- **SOO does almost as well as DOO optimally fitted** (thus "adapts" to the unknown local smoothness of f).

Numerical experiments

Again for the function $f(x) = (\sin(13x)\sin(27x) + 1)/2$ we have:

n	SOO	uniform grid	DOO with ℓ_1	DOO with ℓ_2
50	3.56×10^{-4}	1.25×10^{-2}	2.53×10^{-5}	1.20×10^{-2}
100	5.90×10^{-7}	8.31×10^{-3}	2.53×10^{-5}	1.67×10^{-7}
150	1.92×10^{-10}	9.72×10^{-3}	4.93×10^{-6}	4.44×10^{-16}

The case $d = 0$ is non-trivial!

Example:

- f is locally α-smooth around its maximum:

$$f(x^*) - f(x) = \Theta(\|x^* - x\|^\alpha),$$

  for some $\alpha > 0$.
- SOO algorithm does not require the knowledge of ℓ,
- Using $\ell(x, y) = \|x - y\|^\alpha$ in the analysis, all assumptions are satisfied (with $\gamma = 3^{-\alpha/D}$ and $d = 0$, thus the loss of SOO is $r_n = O(3^{-\sqrt{n}\alpha/(CD)})$ (stretched-exponential loss),
- This is almost as good as DOO optimally fitted!

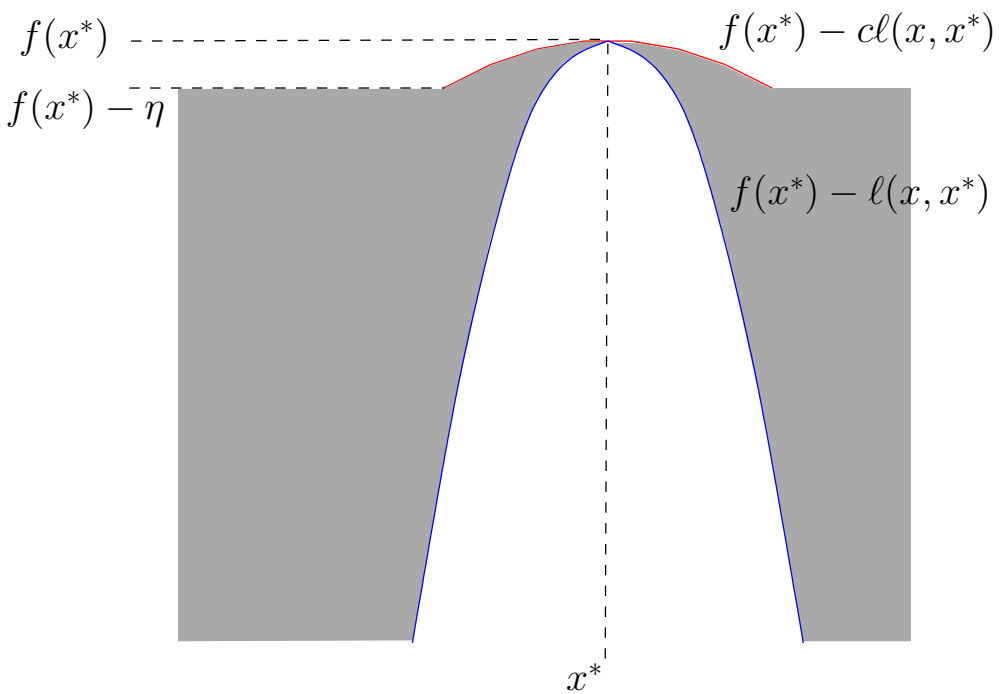
(Extends to the case $f(x^*) - f(x) \approx \sum_{i=1}^{D} c_i |x_i^* - x_i|^{\alpha_i}$)

The case $d = 0$

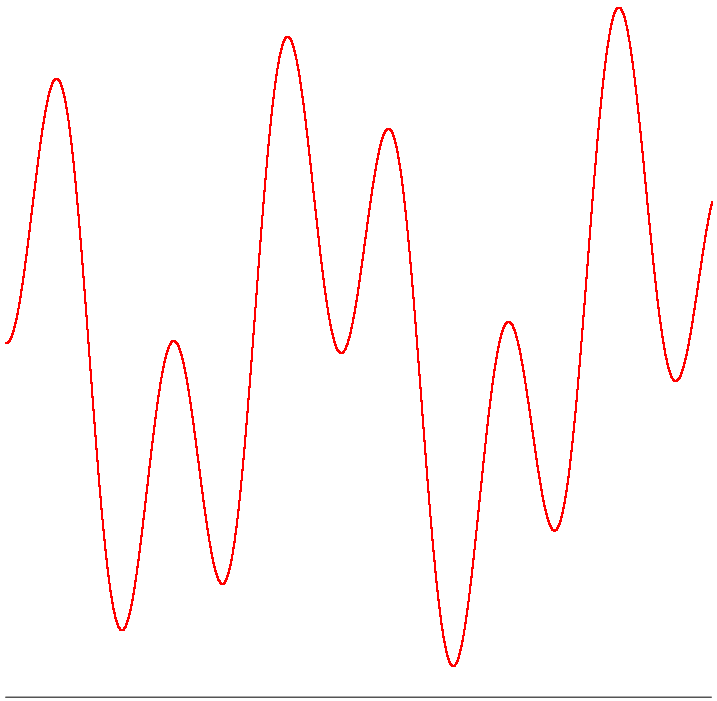
More generally, any function whose **upper- and lower envelopes around x^* have the same shape**: $\exists c > 0$ and $\eta > 0$, such that

$$\min(\eta, c\ell(x, x^*)) \leq f(x^*) - f(x) \leq \ell(x, x^*), \quad \text{for all } x \in \mathcal{X}.$$

has a near-optimality $d = 0$ (w.r.t. the metric ℓ).

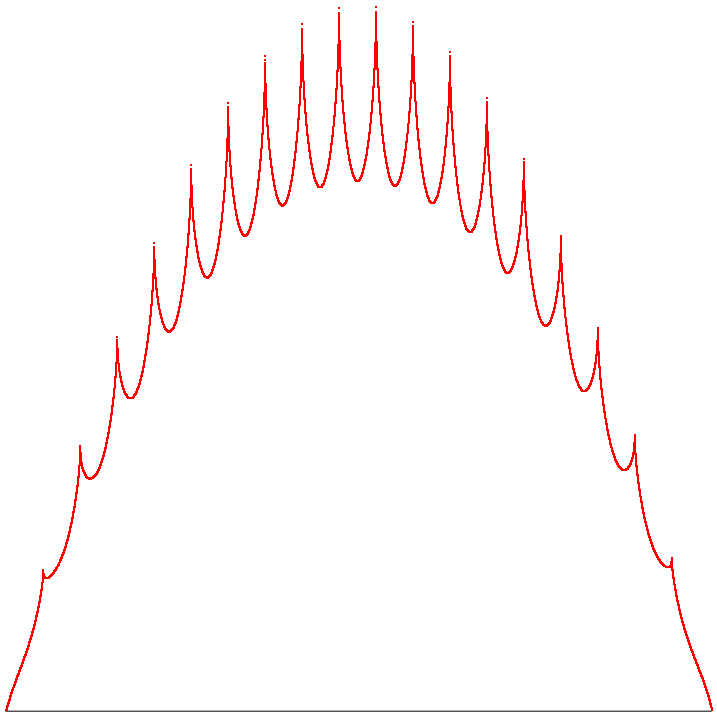


Example of functions for which $d = 0$



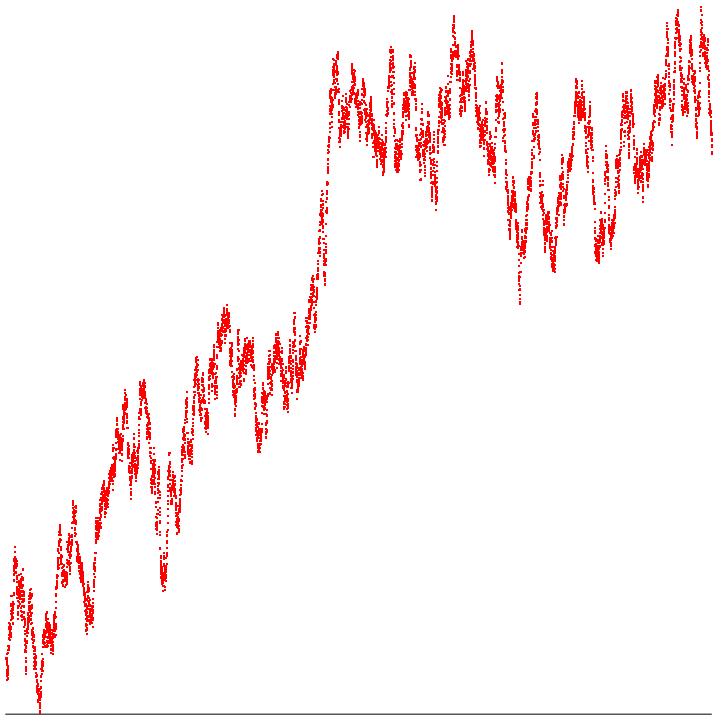
$$\ell(x, y) = c\|x - y\|^2$$

Example of functions with $d = 0$



$$\ell(x, y) = c\|x - y\|^{1/2}$$

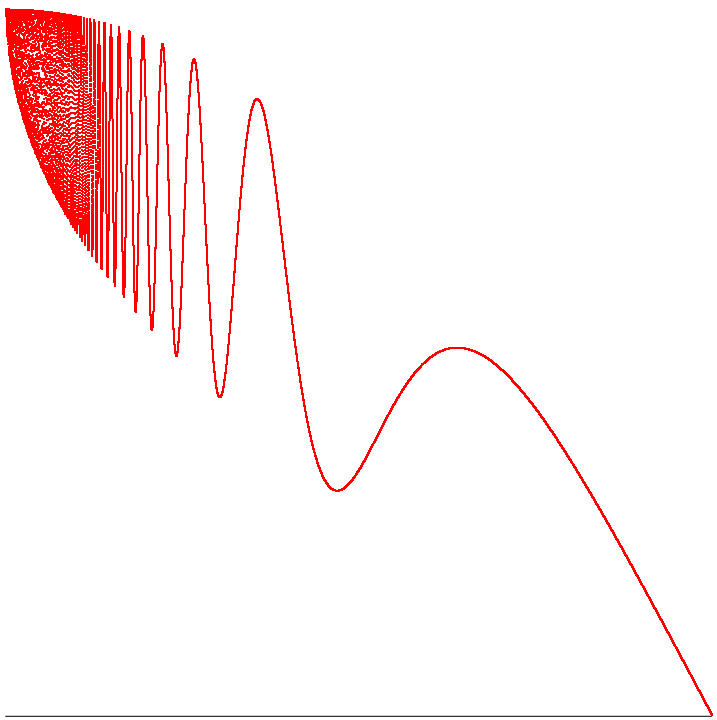
$d = 0$?



$$\ell(x, y) = c\|x - y\|^{1/2}$$

$d > 0$

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



The lower-envelope is of order $1/2$ whereas the upper one is of order 2. We deduce that $d \geq 3/2$.

Sketch of proof

- For any ℓ, define
  $I_h = \{$ cells X_i of depth h such that $f(x_i) + \text{diam}(h) \geq f(x^*)\}$
- Once the optimal cell of depth h has been expanded, it takes at most $|I_{h+1}|$ cell expansions of depth $h+1$ before the optimal cell is expanded.
- Thus $n \leq h_{\max}(n) \sum_{h=0}^{min(h_{\max}(n), h_n^*)} |I_h|$, where h_n^* is the depth of the node containing x^*.
- Assuming $d = 0$, $|I_h| \leq C$, and using $h_{\max}(n) = \sqrt{n}$, we have $\sqrt{n} = C \min(h_{\max}(n), h_n^*) = C h_n^*$
- Finally the value of the returned point $x(n)$ is at least as good as that of the optimal expanded node i_n^* containing x^*:

$$f(x(n)) \geq f(x_{i_n^*}) \geq f(x^*) - \text{diam}(h_n^*) \geq f(x^*) - c\gamma^{\sqrt{n}/C},$$

where we used that the diameters are $c\gamma^h$.

Comparison SOO versus DIRECT algorithms

- **SOO is much more general than DIRECT**: the function is only **locally smooth** and the space is **semi-metric**.
- **Finite-time analysis of SOO** (whereas only a consistency property $\lim_{n\to\infty} r_n = 0$ is available for DIRECT in [Finkel and Kelley, 2004])
- **SOO is a rank-based algorithm**: any transformation of the values while preserving their rank will not change anything in the algorithm. Thus extends to the optimization of function givens pair-wise comparisons.
- SOO is easier to implement...

Stochastic SOO (StoSOO)

A simple way to extends SOO to the case of stochastic rewards is the following:

- Select a cell i (and sample f at x_i) according to SOO based on the values

$$\widehat{\mu}_{i,t} + c\sqrt{\frac{\log n}{T_k(t)}},$$

  (where $\widehat{\mu}_{i,t}$ is the arerage rewards received at x_i and $T_i(t)$ is the number of rewards received at state x_i),
- Expand the cell X_i only if $T_i(t) \geq k$, where k is a parameter.

Remark: This really looks like UCT, except that

- several cells are selected at each round,
- a cell is refined only when we received k samples.

Intuition for StoSOO

With high probability, StoSOO acts as a ϵ-pertubed version of SOO where:

- The values $f(x_i)$ are perturbed by ϵ, where $\epsilon = O(\sqrt{\frac{\log n}{k}})$
- There are only $m = n/k$ evaluations to the function.

Thus the loss of StoSOO is

$$\mathbb{E}r_n(StoSOO) = \mathbb{E}r_m(SOO) + O(\sqrt{\frac{\log n}{k}})$$

Performance of StoSOO

Theorem 5 (Valko, Carpentier, Munos, 2013).

For any semi-metric ℓ such that

- *f is locally smooth w.r.t. ℓ*
- *The ℓ-diameters of the cells decrease exponentially fast with their depth,*
- *The near-optimality dimension of f w.r.t. ℓ is $d = 0$,*

by choosing $k = \frac{n}{(\log n)^3}$, $h_{\max}(n) = (\log n)^{3/2}$, the expected loss of StoSOO is

$$\mathbb{E} r_n = O\Big(\frac{(\log n)^2}{\sqrt{n}}\Big).$$

Comments about StoSOO

In the (rather general) case $d = 0$, StoSOO gives a $\tilde{O}(1/\sqrt{n})$ loss
In comparison to HOO/Zooming:

- HOO optimally fitted gives $O\left(\sqrt{\frac{\log n}{n}}\right)$ loss
- HOO with underestimation of the right smoothness deteriorates
- HOO with overestimation of the right smoothness may not converge.

Thus **StoSOO is almost as good as HOO optimally fitted**.

Range of application of DOO/HOO and SOO/StoSOO

Illustrations in Euclidean spaces $[0,1]^D$ only.

But there are plenty of other semi-metric spaces:

- Trees, graphs, combinatorial spaces, structured spaces, ...
- Ex: space of policies for MDPs

We only require:

- the search space $\mathcal{X}$ to be equipped with a semi-metric ℓ
- a nested (hierarchical) partitioning of the space, such that the ℓ-diameters of the cells decrease with their depth
- f to satisfy a local smoothness property w.r.t. ℓ
- ℓ may or may not be known.

Online planning in a MDP

Setting:

- Assume we have a model of the MDP.
- The state space is large: no way to approximate the value function
- Search for the best policy given an initial state, using a finite numerical budget (number of calls to the model)

Protocol:

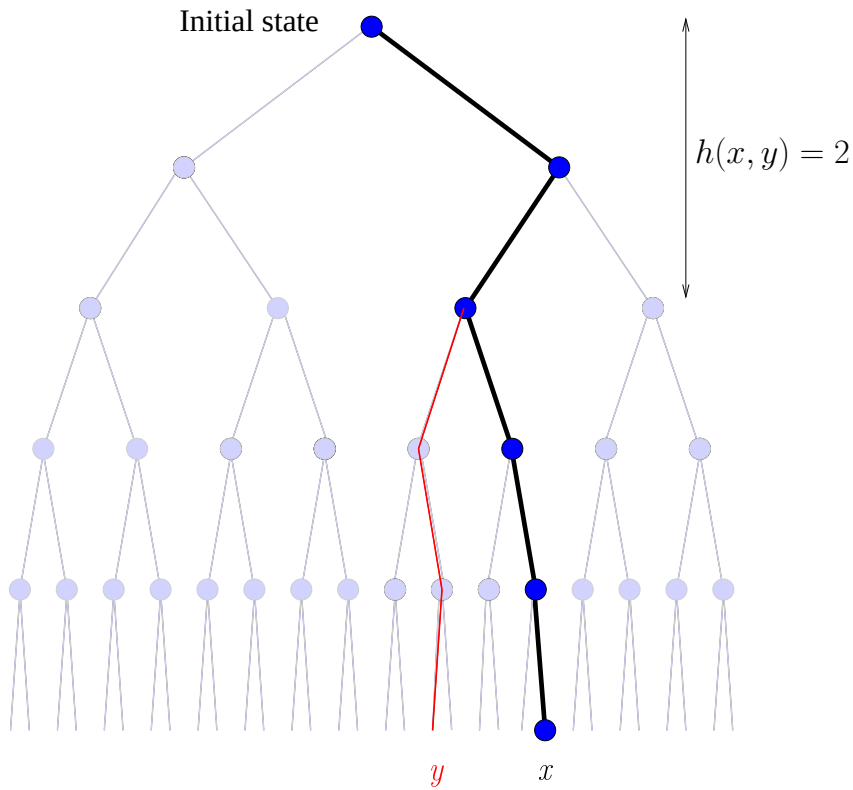
- From current state s_k, perform planning using n calls to the model and recommend action $a(n)$,
- Play $a(n)$, observe next state s_{k+1}, and repeat

$$\text{Loss:} \qquad r_n \stackrel{\text{def}}{=} \max_{a \in A} Q^*(s_k, a) - Q^*(s_k, a(n)).$$

Deterministic transitions and rewards

(infinite time horizon and discounted setting)

- A policy x is an infinite path
- Value $f(x) = \sum_{s \geq 0} \gamma^s r_s(x)$, where the rewards are in $[0, 1]$
- Metric: $\ell(x, y) = \frac{\gamma^{h(x,y)}}{1-\gamma}$
- Prop: $f(x)$ is Lipschitz w.r.t. ℓ
- $\rightarrow$ Use optimistic search

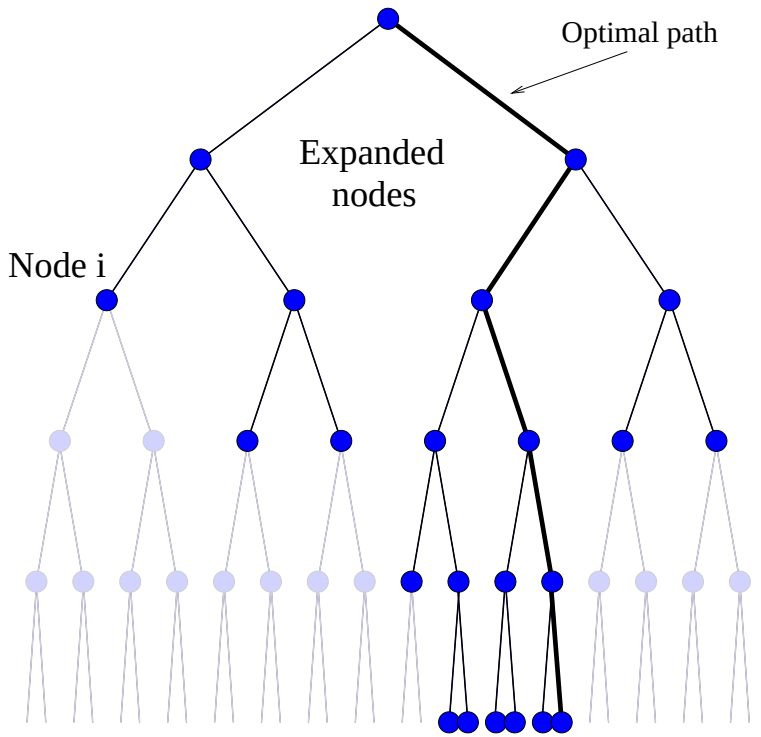


OPD algorithm

Optimistic Planning in Deterministic systems:

- Define the B-values:

$$B_i \stackrel{\text{def}}{=} \sum_{s=0}^{d(i)} \gamma^s r_s + \frac{\gamma^{d(i)+1}}{1-\gamma}$$



- We have $B_i \geq \max_{x \ni i} f(x)$
- For each round $t = 1$ to n, expand the node with highest B-value
- Observe reward, update B-values

Recommend the first action $a(n)$ of the best policy.

Performance of OPD

Define β such that $\mathbb{P}(\text{Random path is } \epsilon\text{-optimal}) = O(\epsilon^\beta)$.

Define $\kappa \stackrel{\text{def}}{=} K\gamma^\beta \in [1, K]$. Then κ is the branching factor of the set of near-optimal sequences:

$$I = \left\{\text{sequences of length } h \text{ such that } \sum_{s=0}^{h} \gamma^s r_s \geq V^* - \frac{\gamma^{h+1}}{1-\gamma}\right\}.$$

Property: κ relates to the near-opt. dimension $d = \frac{\log \kappa}{\log 1/\gamma}$ (the set of ϵ-optimal paths is covered by $O(\epsilon^{-d})$ ℓ-balls of radius ϵ)

Loss of OPD [Hren and Munos, 2008]:

$$r_n = O(n^{-\frac{1}{d}}) = O\left(n^{-\frac{\log 1/\gamma}{\log \kappa}}\right).$$

Performance depends on the quantity of near-optimal policies

κ-minimax lower bounds

Let $\mathcal{M}_\kappa$ the set of problems with coefficient κ.
Upper-bound of OPD uniformly over $\mathcal{M}_\kappa$

$$\sup_{M \in \mathcal{M}_\kappa} r_n(\mathcal{A}_{OPD}, M) = O\left(n^{-\frac{\log 1/\gamma}{\log \kappa}}\right).$$

We can prove that we have a κ-minimax lower-bound:

$$\sup_{\mathcal{A}} \inf_{M \in \mathcal{M}_\kappa} r_n(\mathcal{A}, M) \geq \Omega\left(n^{-\frac{\log 1/\gamma}{\log \kappa}}\right).$$

Sketch of proof:
OPD only expands nodes in I. Reciproquely, I is the set of nodes that need to be expanded in order to find the optimal path.

Extension to stochastic rewards

Dynamics are still deterministic thus the space of policies is still the set of sequences of actions.

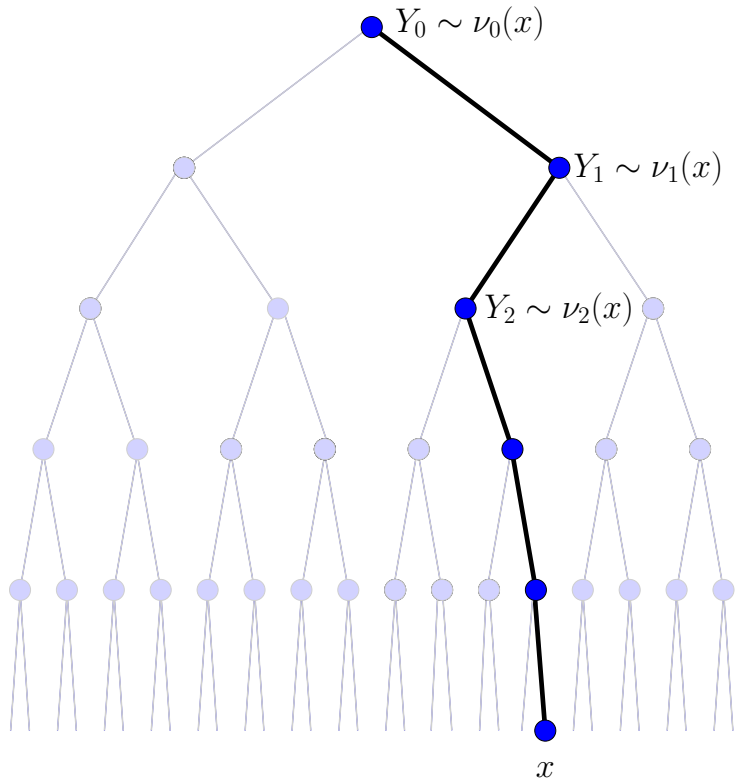
Stochastic rewards:

- Reward along a path x:

$$\sum_{s \geq 0} \gamma^s Y_s,$$

where $Y_s \sim \nu_s(x)$ where $\nu_s(x)$ is the reward distribution ($\in [0,1]$) after s actions along path x.

- Write $r_s(x) = \mathbb{E}_{X \sim \nu_s(x)}[X]$ and $f(x) = \sum_{s \geq 0} \gamma^s r_s(x)$



Then $f(x)$ is Lipschitz w.r.t. $\ell(x,y) = \frac{\gamma^{h(x,y)}}{1-\gamma}$ and one can think of applying HOO.

Using HOO for planning

Apply HOO to the search space $\mathcal{X}$:

- Assign a B-value to each finite sequence
- At each round t, select a finite sequence x_t maximizing the B-value.
- Observe sample reward $\sum_{s \geq 0} \gamma^s Y_s(x_t)$ of the path x_t and use it to update the B-values.
- The loss is

$$O(n^{-\frac{1}{d+2}}).$$

Problem: HOO does not make full use of the tree structure: It uses the sample reward of the whole path x but not the individual reward samples $Y_s(x)$ collected along the path x.

Optimistic sampling using the tree structure

Open Loop Optimistic Planning (OLOP) [Bubeck, Munos, 2010]:

- At round t, play path x_t (up to depth $h = \frac{1}{2}\frac{\log n}{\log 1/\gamma}$)
- Observe sample rewards $Y_s(x_t)$ of each node along the path x_t
- Compute empirical rewards $\hat{\mu}_t(x_{1:s})$ for each node $x_{1:s}$ of depth $s \leq h$
- Define bound for each path x:

$$B_t(x) = \min_{1 \leq j \leq h} \left[\sum_{s=0}^{j} \gamma^s \left(\hat{\mu}_t(x_{1:s}) + \sqrt{\frac{2 \log n}{T_{x_{1:s}}(t)}} \right) + \frac{\gamma^{j+1}}{1-\gamma} \right]$$

- Select path $x_{t+1} = \operatorname{argmax}_x B_t(x)$

This algorithm fully uses the tree structure of the rewards.

Performance of OLOP

Define

- $\beta \geq 0$ such that $\mathbb{P}(\text{Random path is } \epsilon\text{-optimal}) = O(\epsilon^\beta)$.
- or $\kappa \overset{\text{def}}{=} K\gamma^\beta \in [1, K]$ the branching factor of the set of near-optimal sequences.
- or the near-optimality dimension, $d = \frac{\log \kappa}{\log 1/\gamma}$.

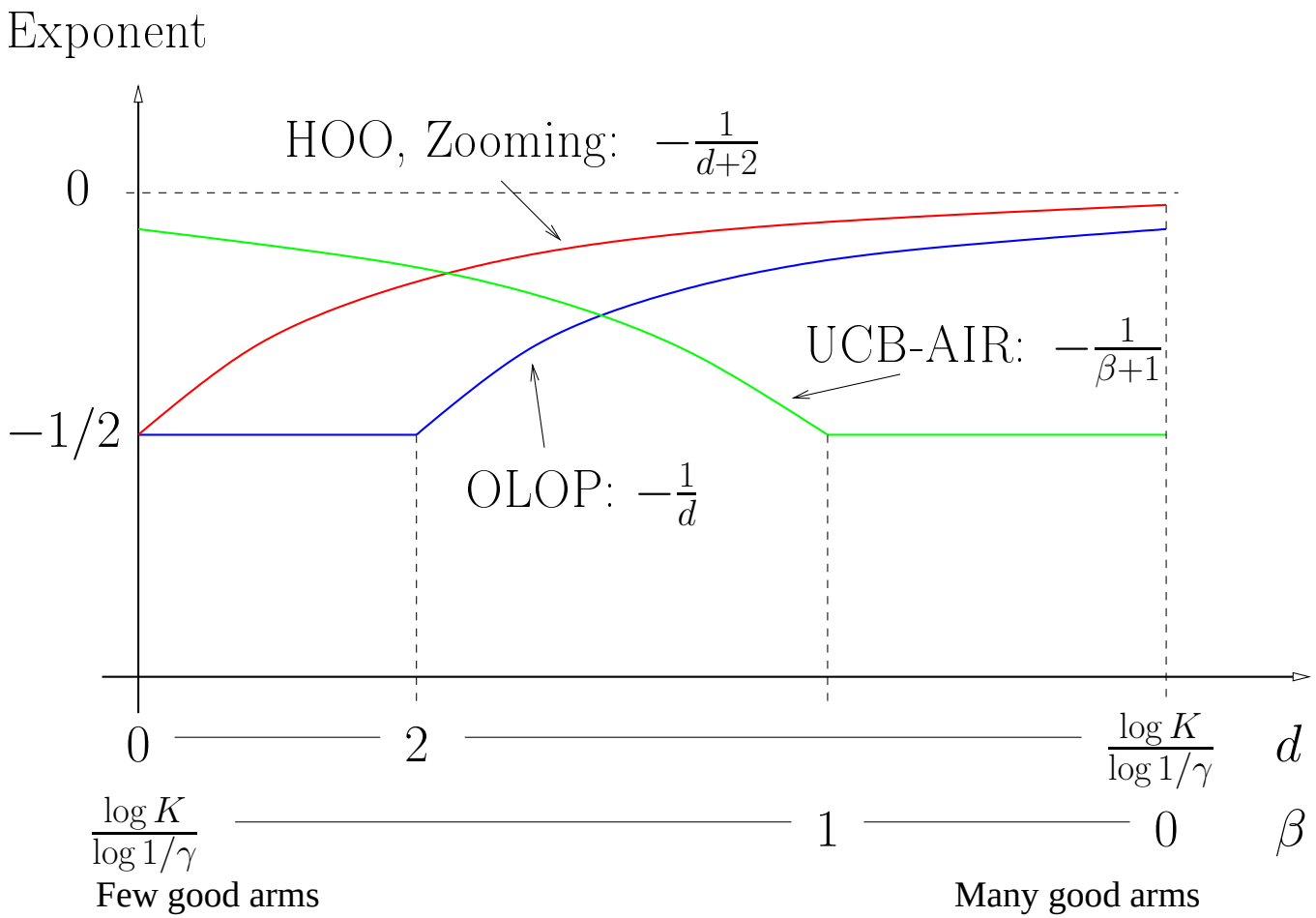
Theorem 6 (Loss of OLOP).

After n calls to the generative model,

$$\mathbb{E}r_n = f(x^*) - \mathbb{E}f(x(n)) = \begin{cases} \widetilde{O}\left(n^{-1/d}\right) & \text{if } d \geq 2 \\ \widetilde{O}\left(1/\sqrt{n}\right) & \text{if } d < 2 \end{cases}$$

Much better than HOO! As good as OPD for $d \geq 2$.

Comparison: OLOP, HOO, Zooming, UCB-AIR



Optimistic Planning in MDPs

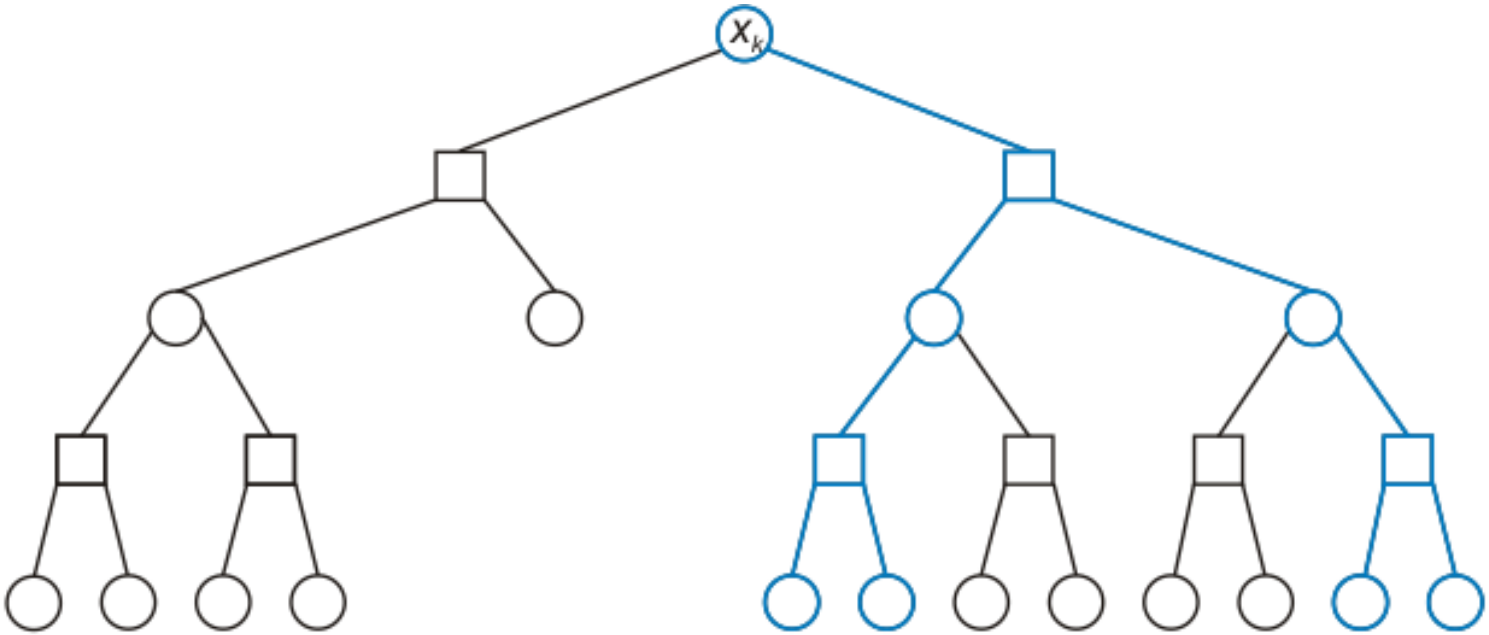
Stochastic transitions, but assume that the number of next states is finite.

Here a policy is no more a sequence of actions

OP-MDP [Buşoniu and Munos, 2012]:

- The root = current state.
- For $t = 1$ to n:
  - Compute the B-value of all nodes of the current sub-tree
  - Compute the optimistic policy
  - Select a leaf of the optimistic sub-tree and expand it (generates transitions to next states using the model)
- Return first action of the best policy

Illustration of OP-MDP



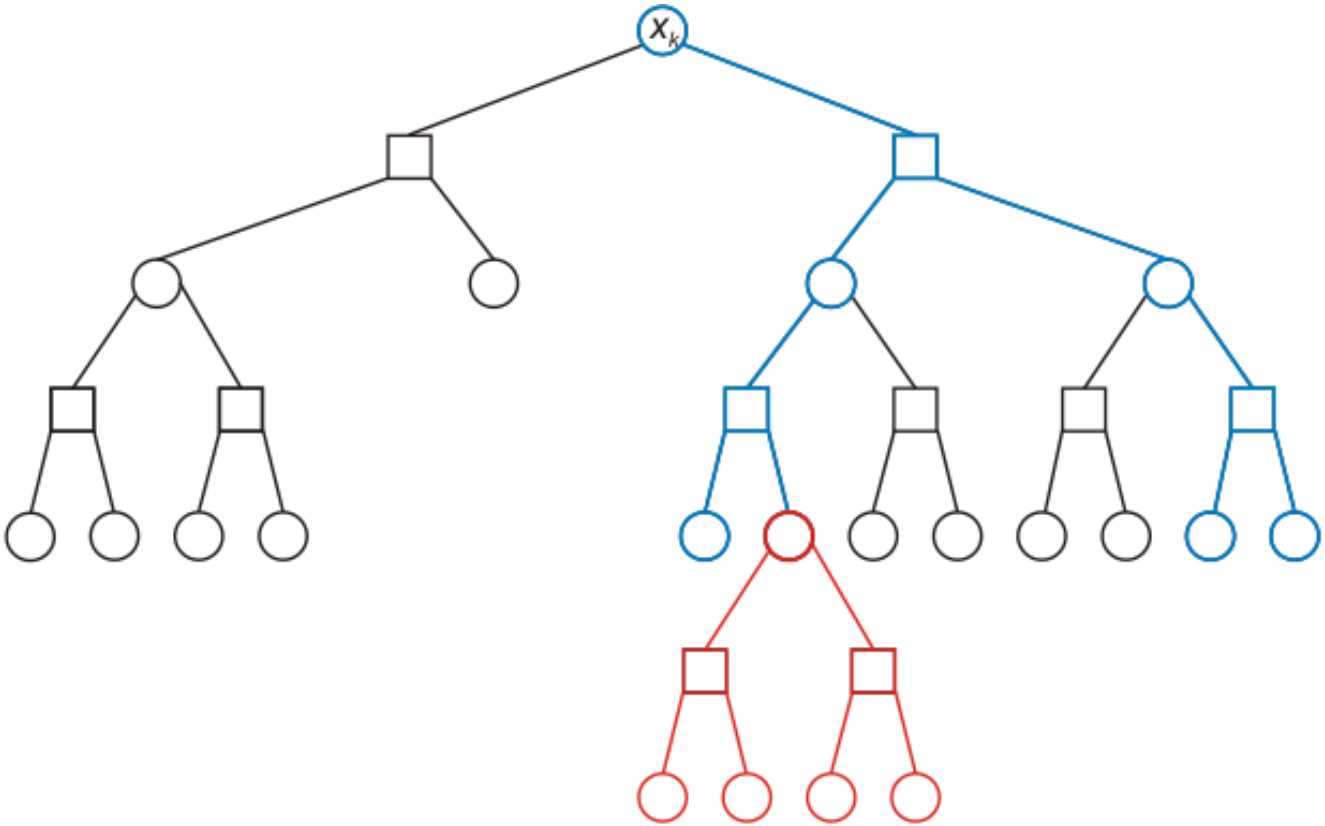
B-values: upper-bounds on the optimal value function $V^*(s)$:

$$B(s) = \frac{1}{1-\gamma} \text{ for leaves}$$

$$B(s) = \max_a \sum_{s'} p(s'|s,a)\left[r(s,a,s') + \gamma B(s')\right]$$

Compute the **optimistic policy** π^+.

Optimistic Planning in MDPs



Expand leaf in π^+ with largest contribution: $\arg\max_{s \in \mathcal{L}} P(s) \frac{\gamma^{d(s)}}{1-\gamma}$, where $P(s)$ is the probability to reach s when following π^+.

Performance analysis of OP-MDP

Define X_ϵ the set of states

- whose "contribution" is at least ϵ
- and that belong to an ϵ-optimal policy

Near-optimality planning dimension: Define the measure of complexity of planning in the MDP as the smallest $d \geq 0$ such that $|X_\epsilon| = O(\epsilon^{-d})$.

Theorem 7.

The performance of OP-MDP is $r_n = O(n^{-1/d})$.

The performance depends on the quantity of states that contribute significantly to near-optimal policies

Illustration of the performance

Reminder: $r_n = O(n^{-1/d})$.

Uniform rewards and probabilities $d = \frac{\log K + \log N}{\log 1/\gamma}$ (uniform planning)

Structured rewards, uniform probabilities $d = \frac{\log N}{\log 1/\gamma}$ (uniform planning for a single policy)

Uniform rewards, concentrated probabilities $d \to \frac{\log K}{\log 1/\gamma}$ (planning in deterministic systems)

Structured rewards, concentrated probabilities $d \to 0$ (exponential rate)

Remarks: d is small when

- Structured rewards
- Heterogeneous transition probabilities

Towards d-minimax lower bounds

Let $\mathcal{M}_d$ the set of MDPs with near-optimality planning dim. d
Upper-bound of OP-MDP uniformly over $\mathcal{M}_\beta$

$$\sup_{M \in \mathcal{M}_d} r_n(\mathcal{A}_{OP-MDP}, M) \leq O(n^{-1/d}).$$

We conjecture that we have a d-minimax lower-bound:

$$\sup_{\mathcal{A}} \inf_{M \in \mathcal{M}_d} \mathbb{E} r_n(\mathcal{A}, M) \geq \Omega(n^{-1/d}).$$

Conclusions on optimistic planning

- Perform optimistic search in policy space.
- In deterministic dynamics, deterministic rewards, can be seen as a direct application of optimistic optimization
- In stochastic rewards, the structure of the reward function can help estimation of paths given samples from other paths
- In MDPs the **near-optimality planning dimension** is a new measure of the quantity of states that need to be expanded (*the set of states that significantly contribute to near-optimal policies*)
- Fast rates when the MDP has structured rewards and heterogeneous transition probabilities.
- Applications to Bayesian Reinforcement learning and planning in POMDPs.

Possible extensions in optimistic planning

- Extends OP-MDP to the case when only a generative model of the dynamics is available
- Extension to a possibly infinite number of next-states
- Apply SOO / StoSOO ideas for planning: Although the value function is Lipschitz w.r.t. the metric $\ell(x, y) = \frac{\gamma^{h(x,y}}{1-\gamma}$, it may possess additional smoothness around the maximum with a higher-order semi-metric.

Main message of this tutorial

The **"optimism in the face of uncertainty" principle:**

- applies in a large class of decision making problems in stochastic and deterministic environments (in unstructured and structured problems)
- provides an efficient exploration of the search space by exploring the most promising areas first
- provides a natural transition from global to local search
- Performance depends on the "smoothness" of the function around the maximum w.r.t. some metric,
  - a measure of the quantity of near-optimal solutions,
  - and our knowledge or not of this metric.

Conclusions

Thanks !!!

Slides and a review paper are available from my web page:

`http://chercheurs.lille.inria.fr/~munos/`