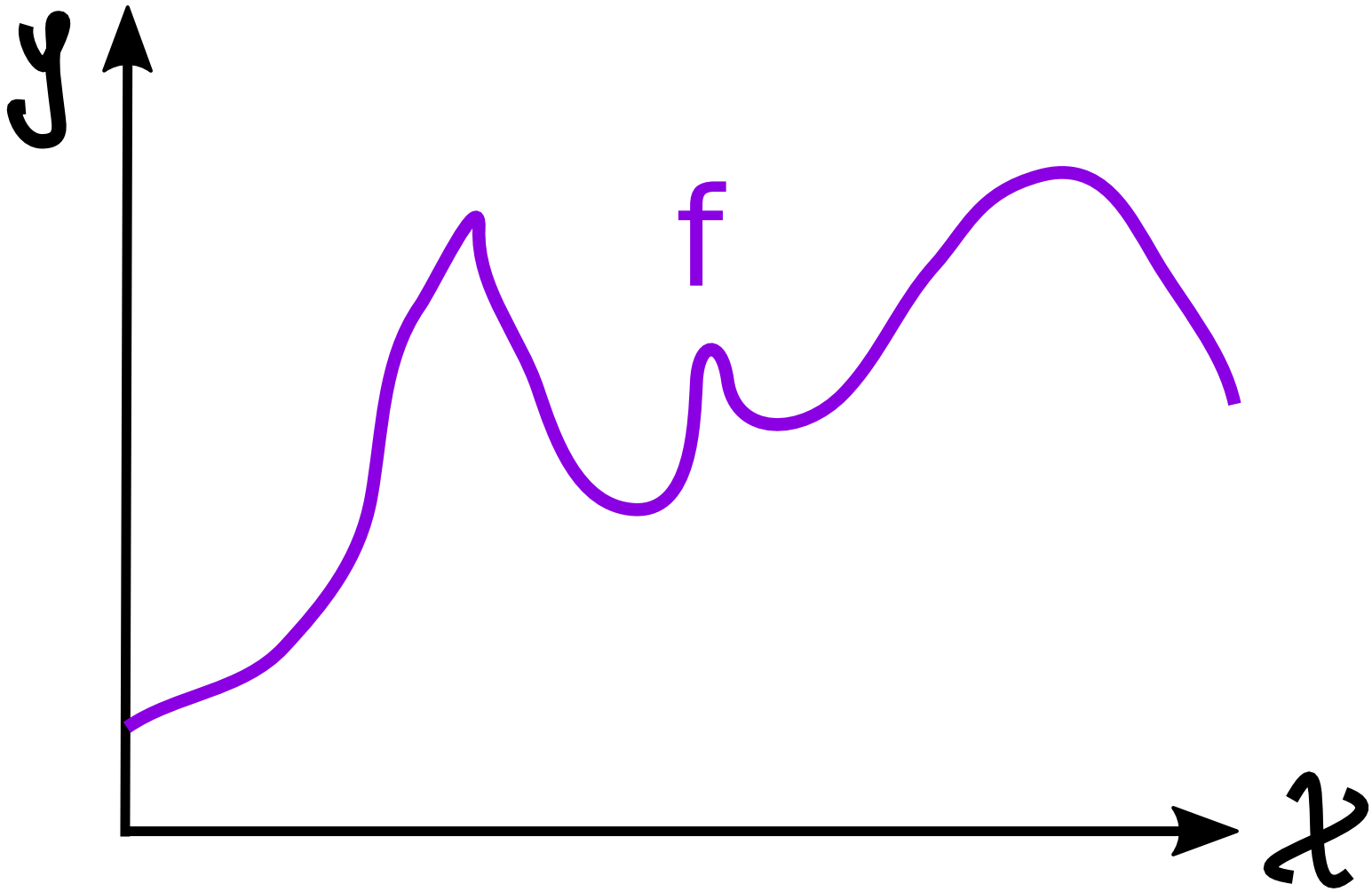
General parallel optimization WITHOUT a metric

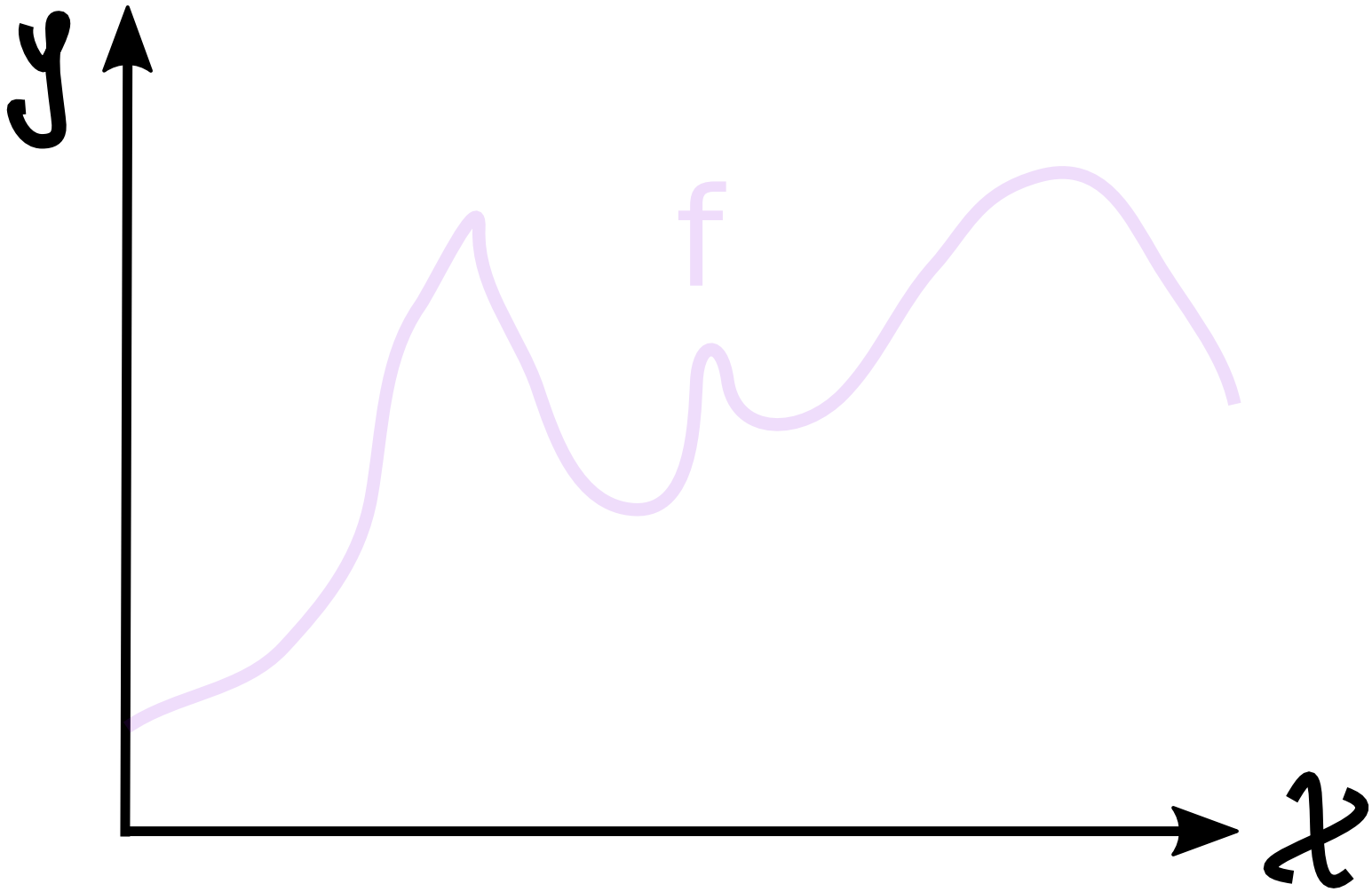
Xuedong Shang, Emilie Kaufmann, Michal Valko

ALT - 23 March 2019

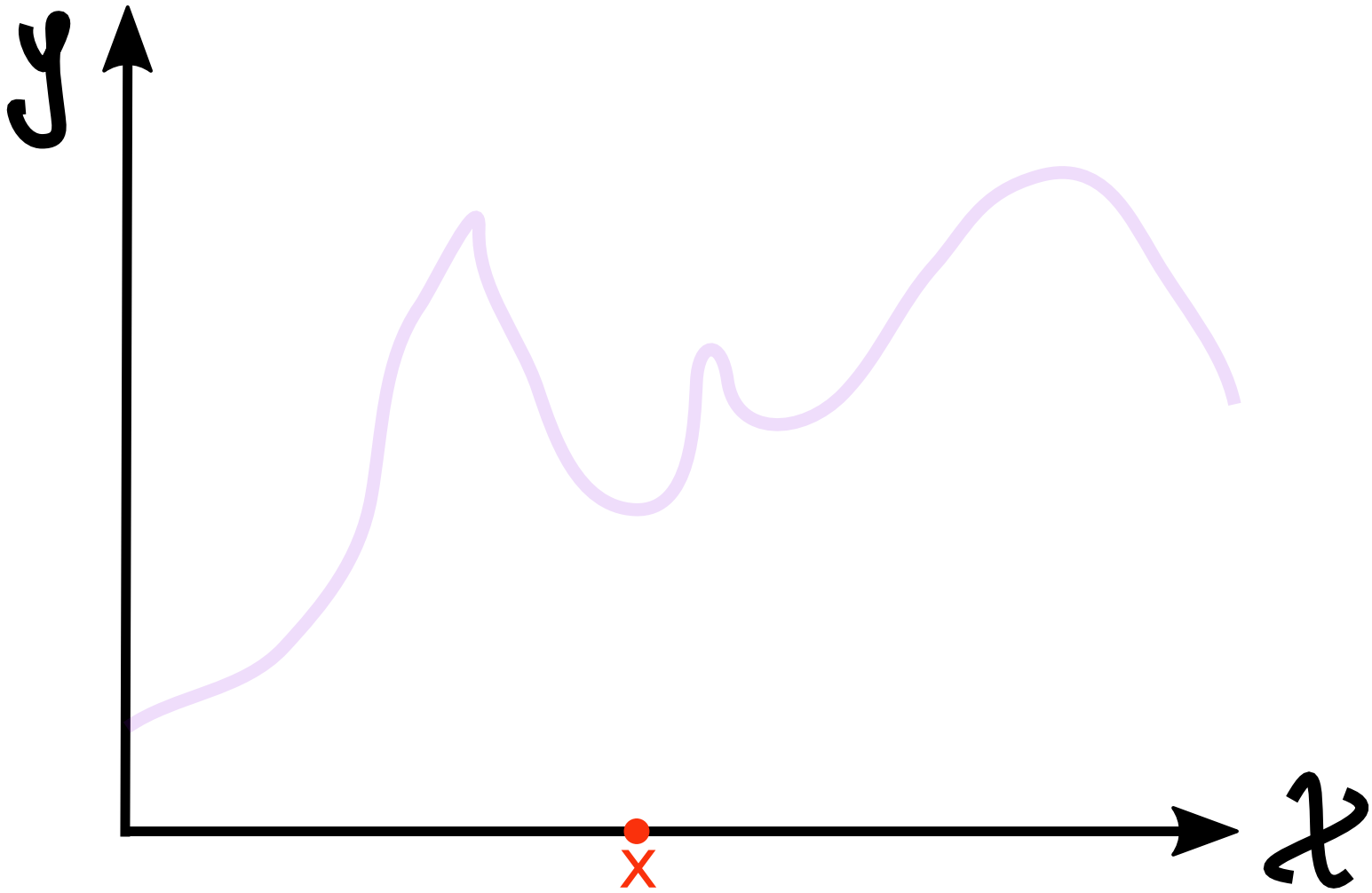
Black box optimization



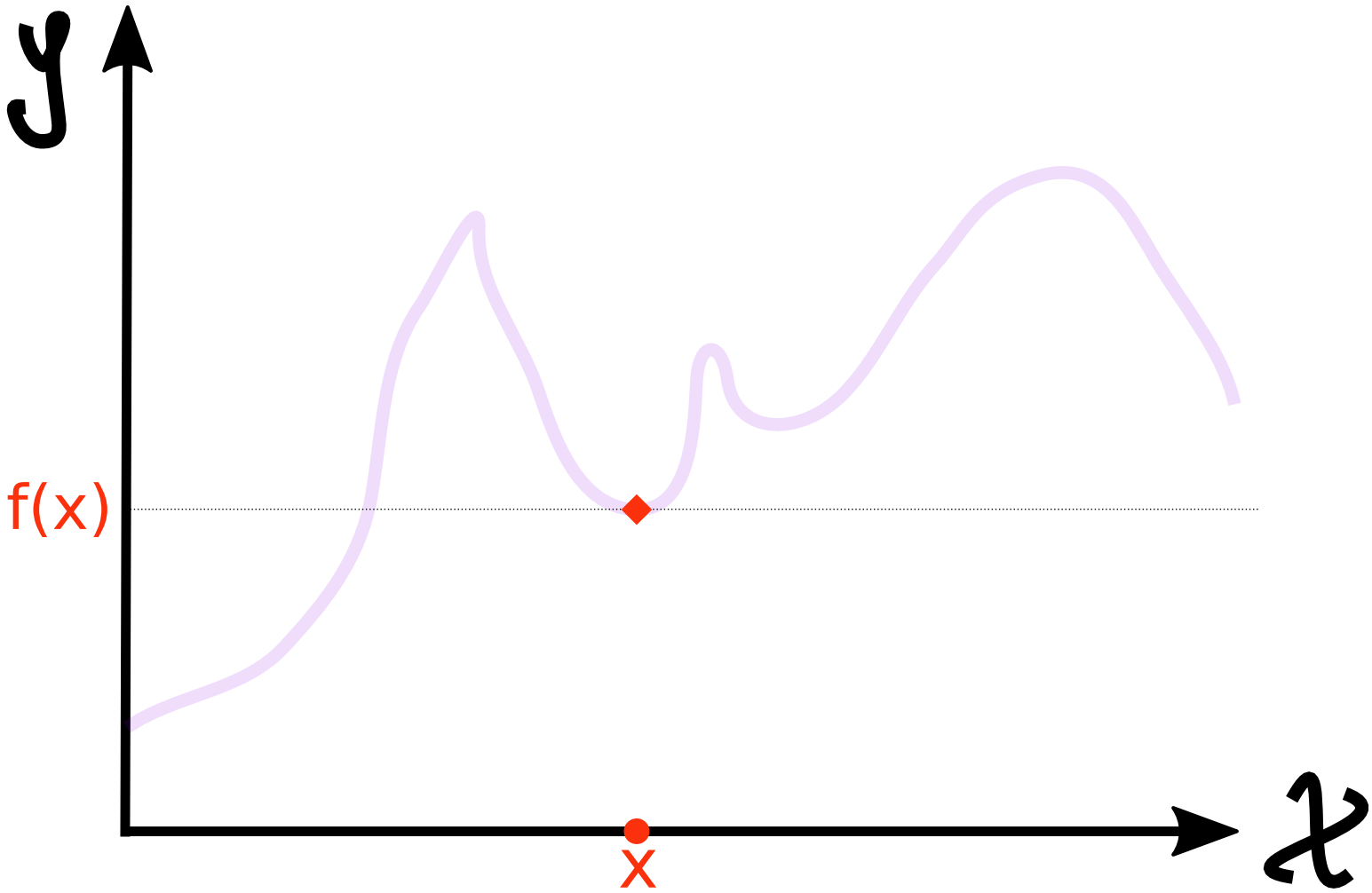
Black box optimization



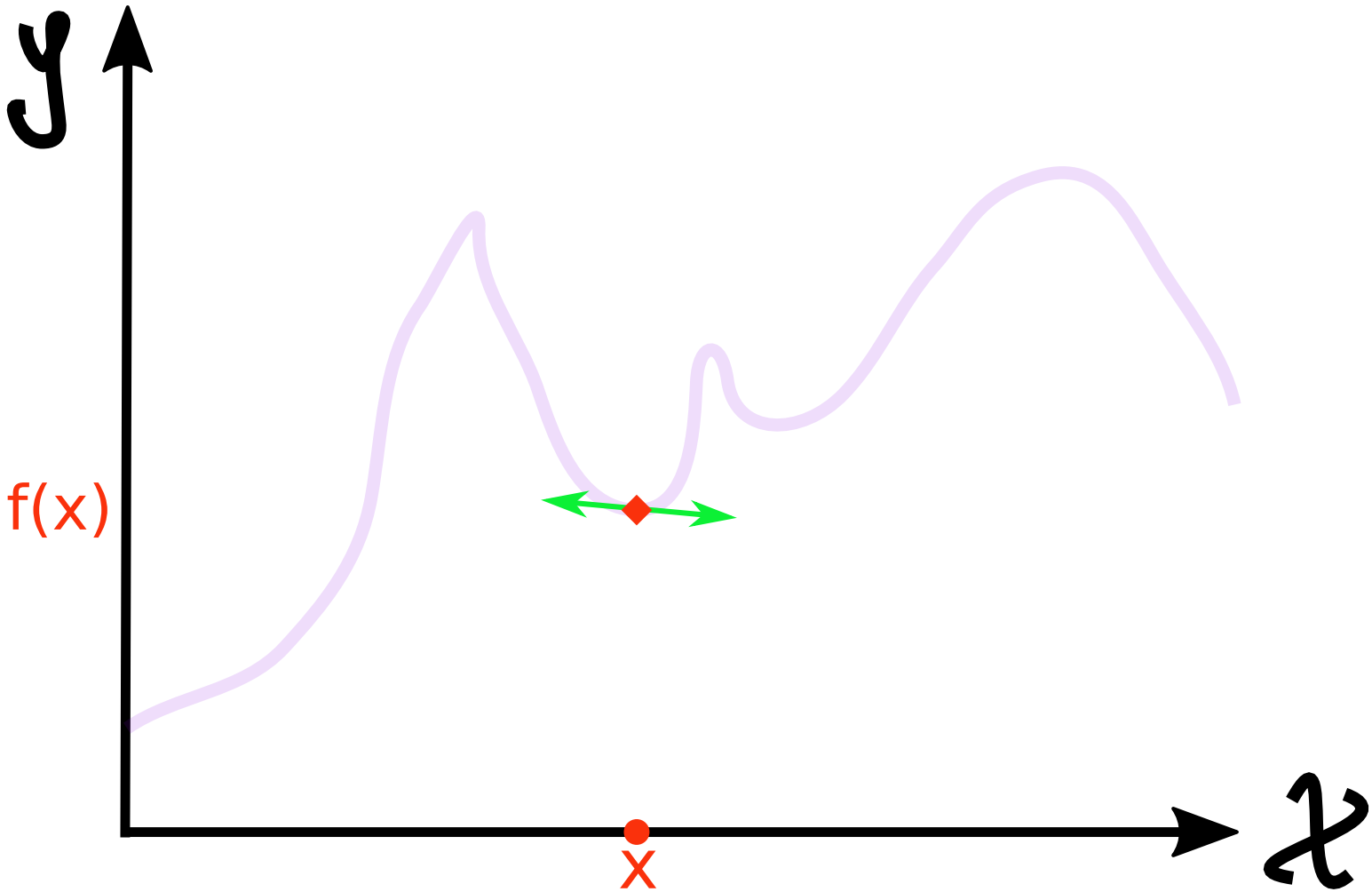
Black box optimization



Black box optimization

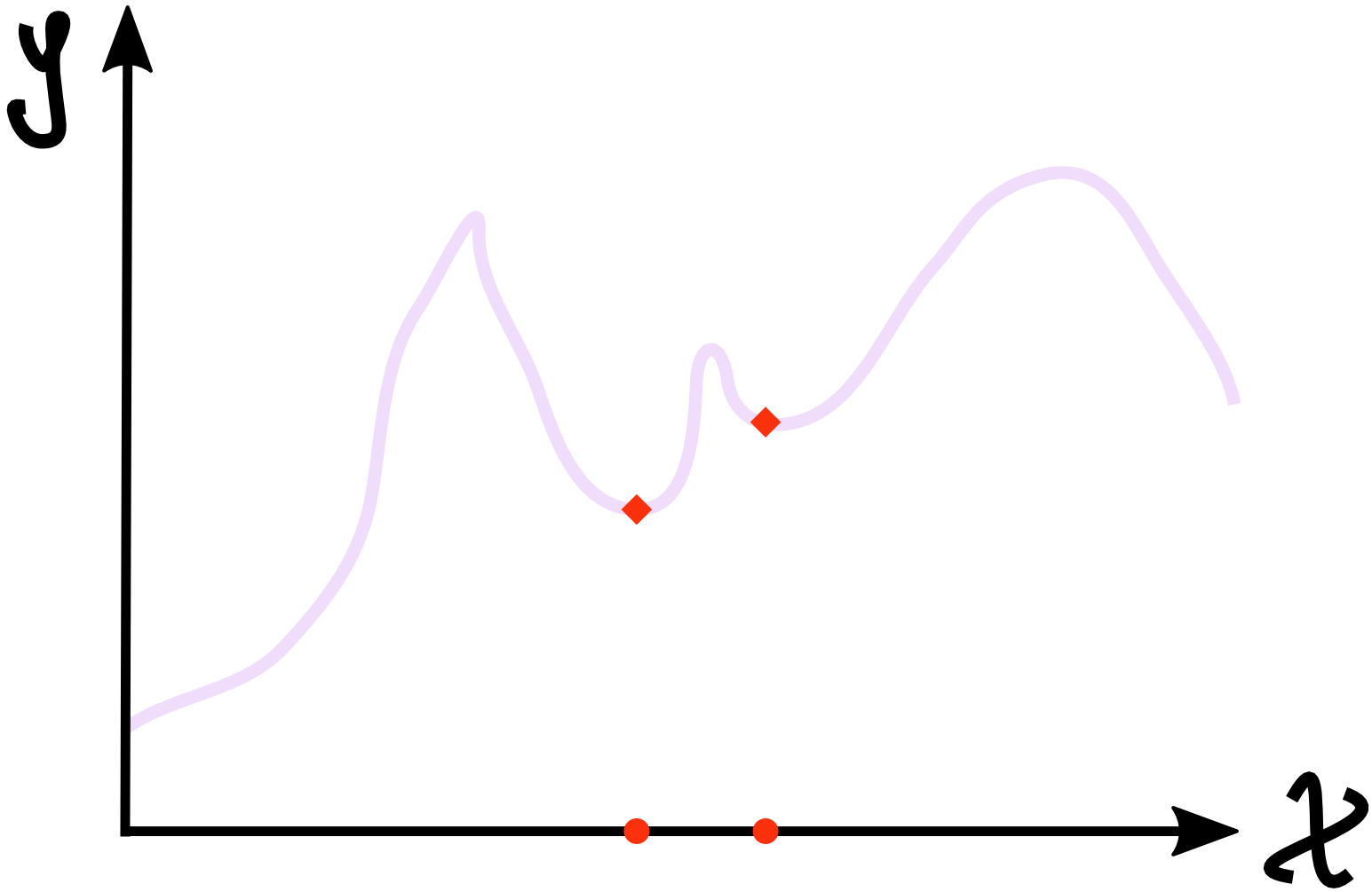


Black box optimization

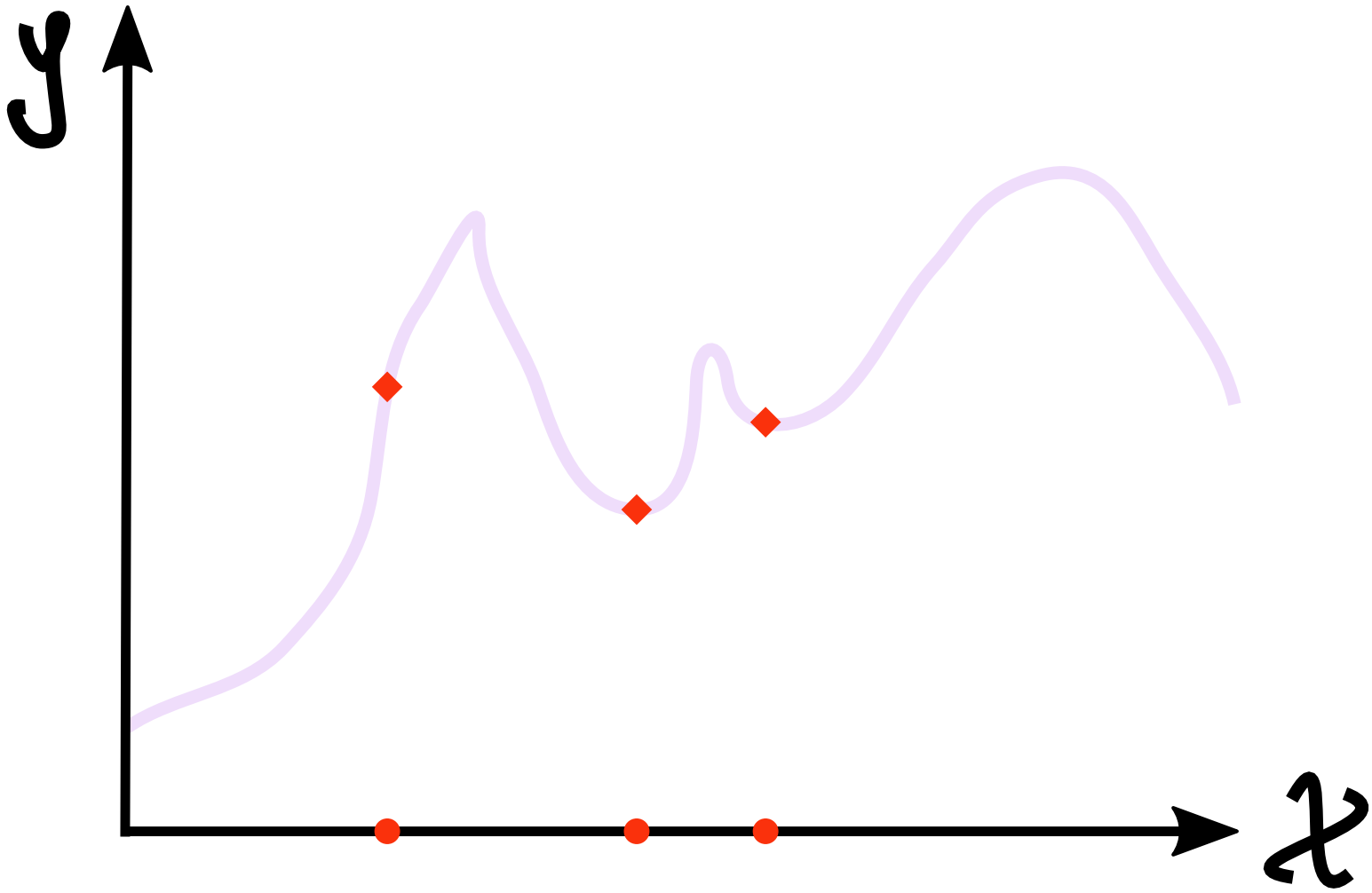


Any gradient information? No! Also called zero-order optimization.

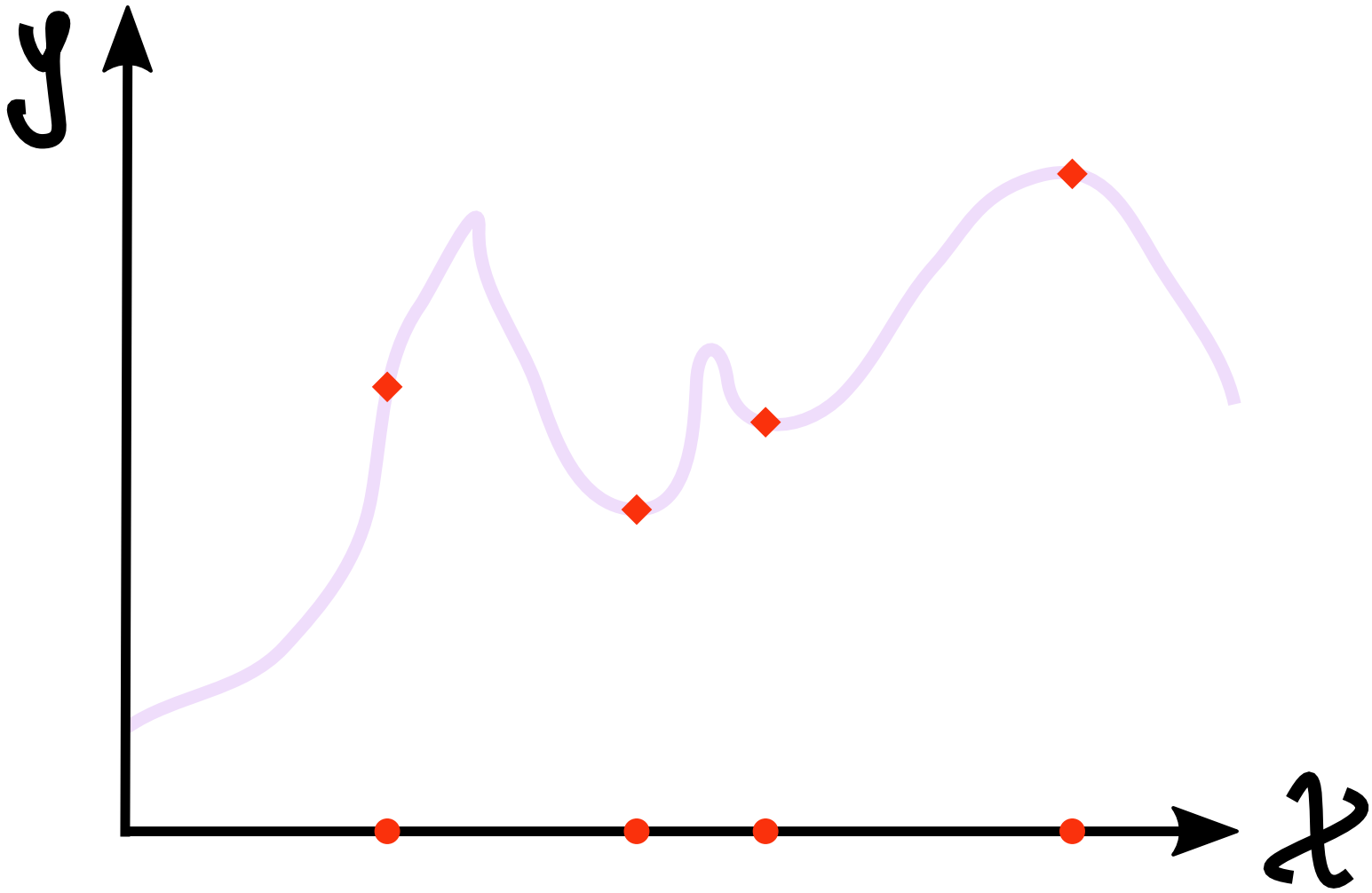
Black box optimization



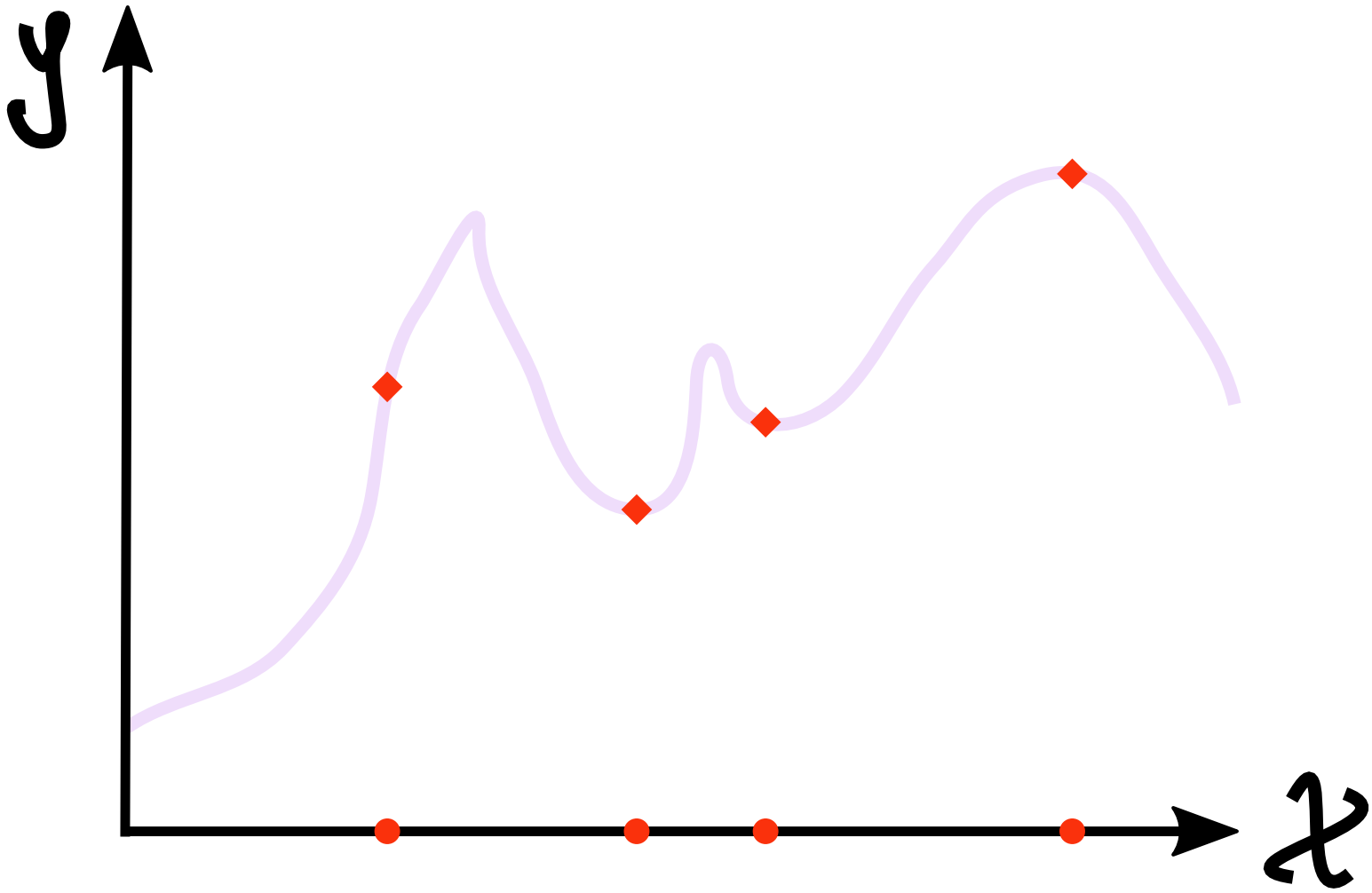
Black box optimization



Black box optimization

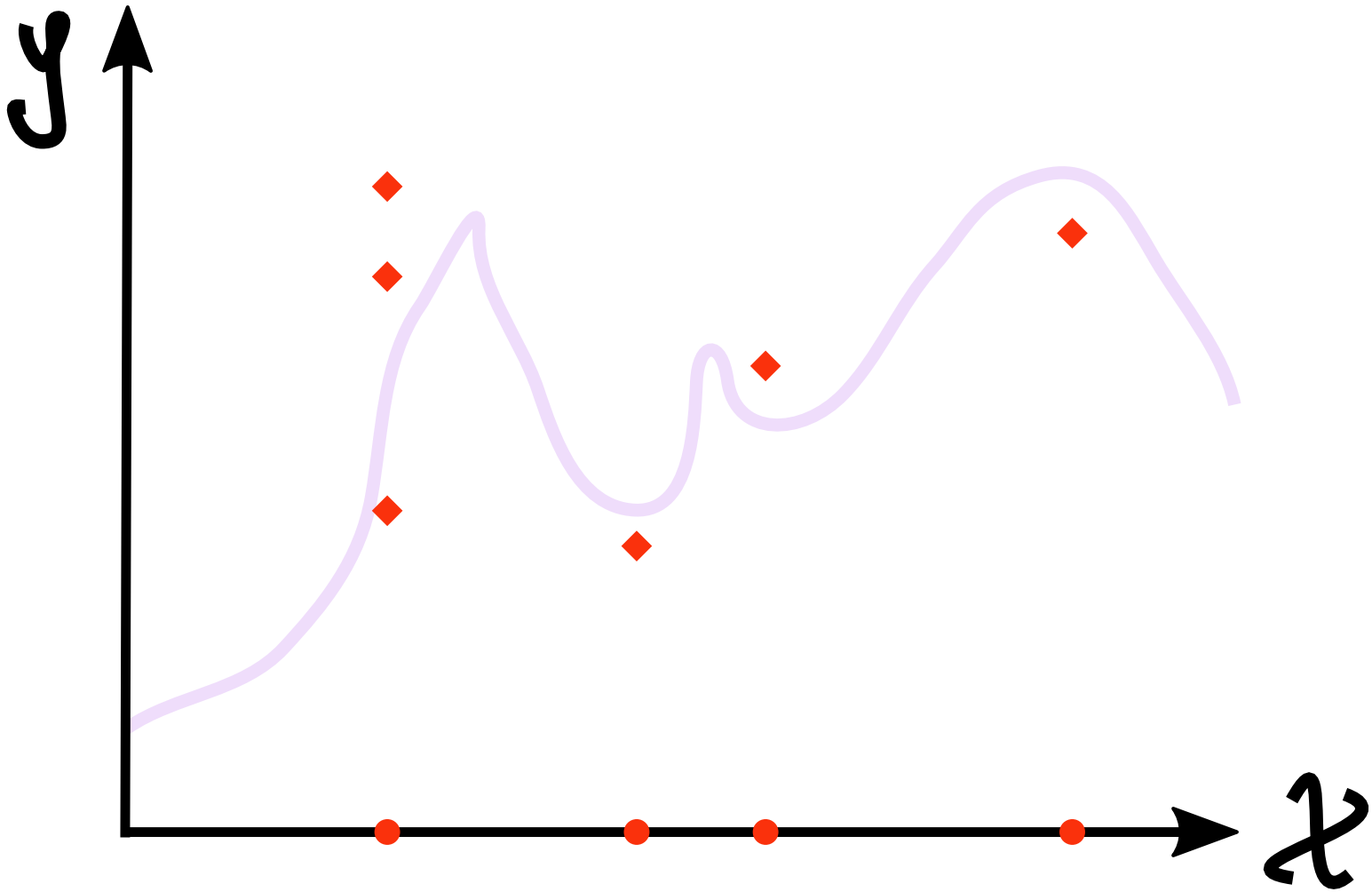


Black box optimization



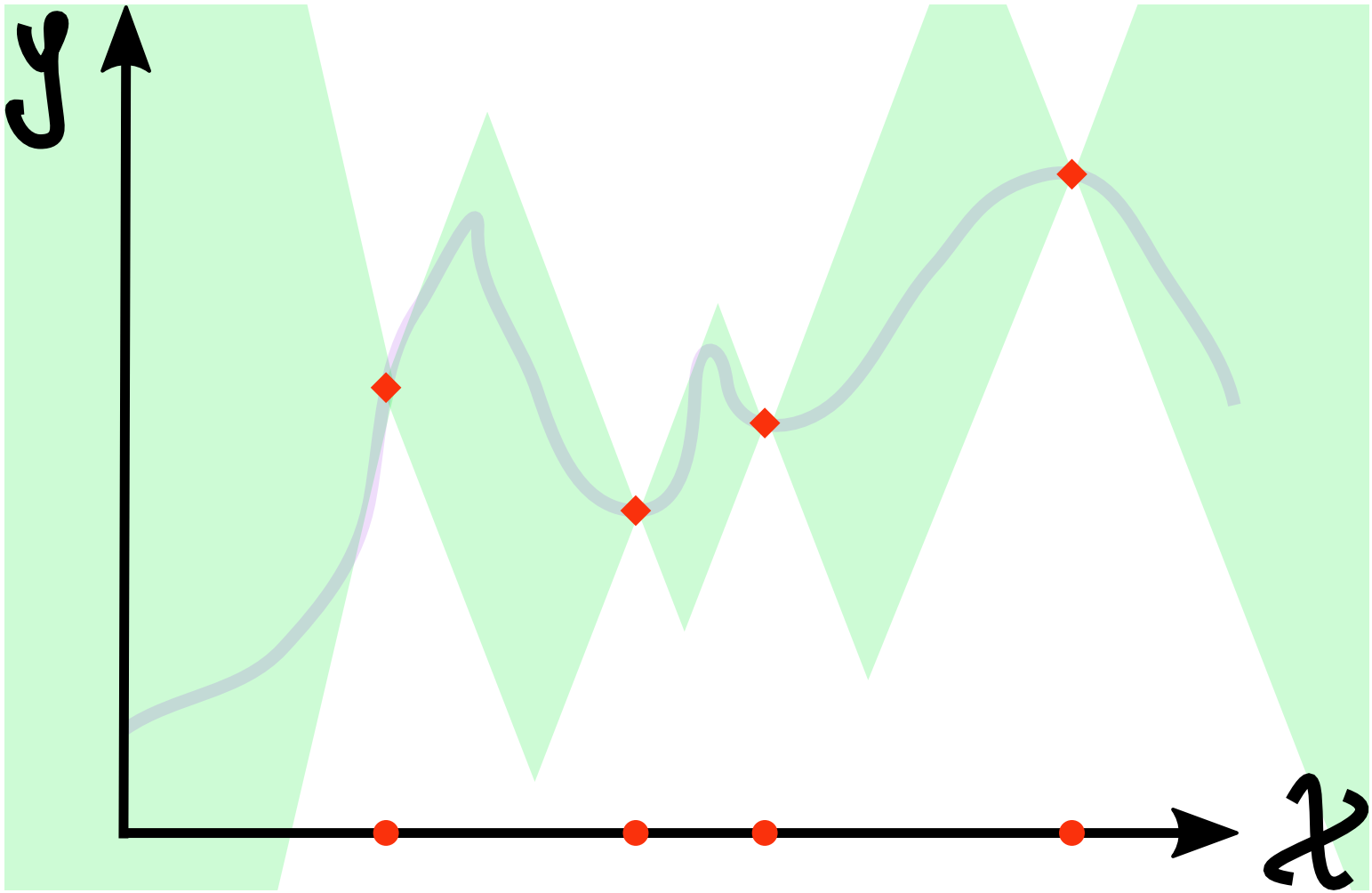
How to choose the next query?

Black box optimization



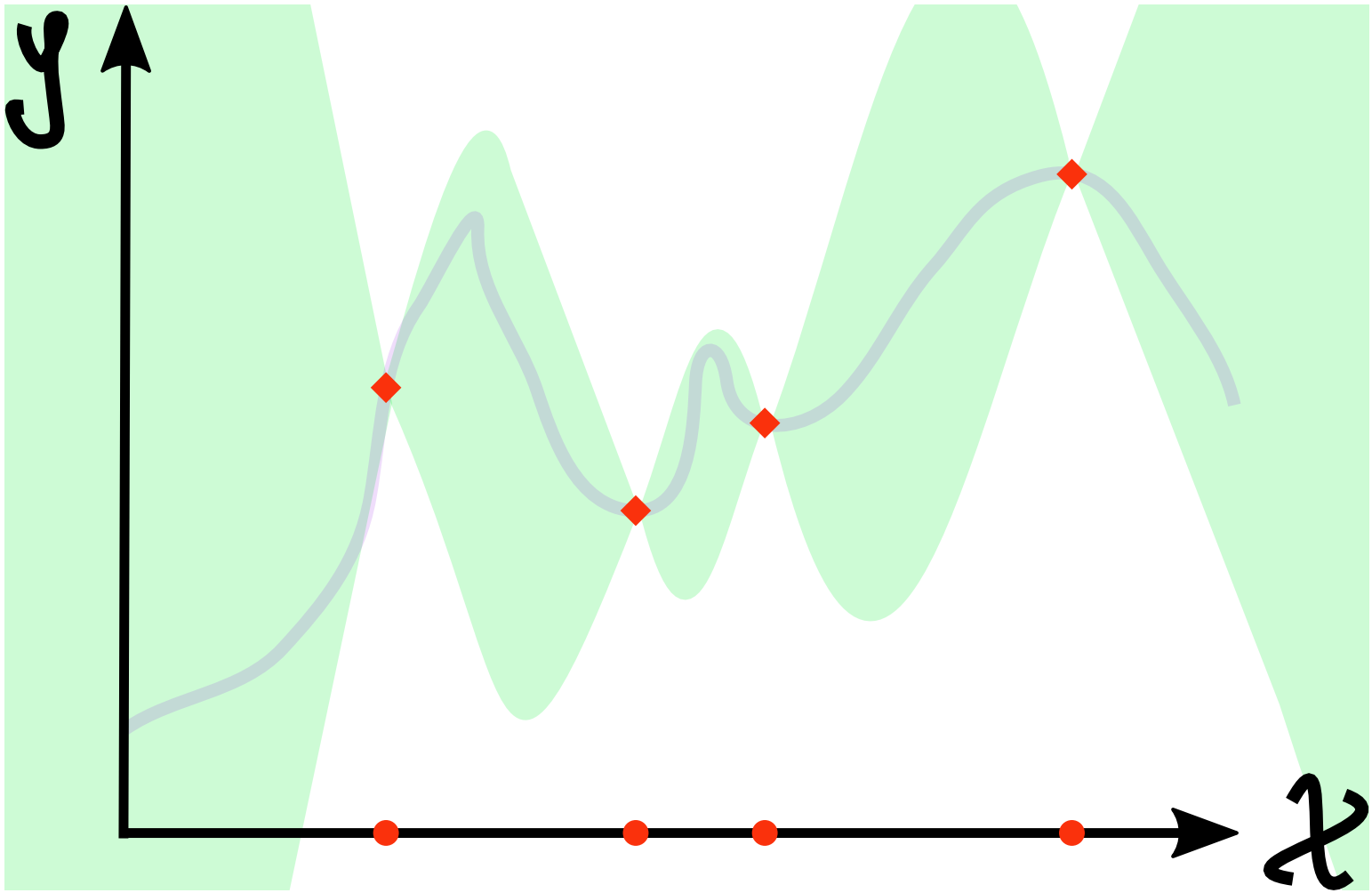
How to choose the next query?

Black box optimization



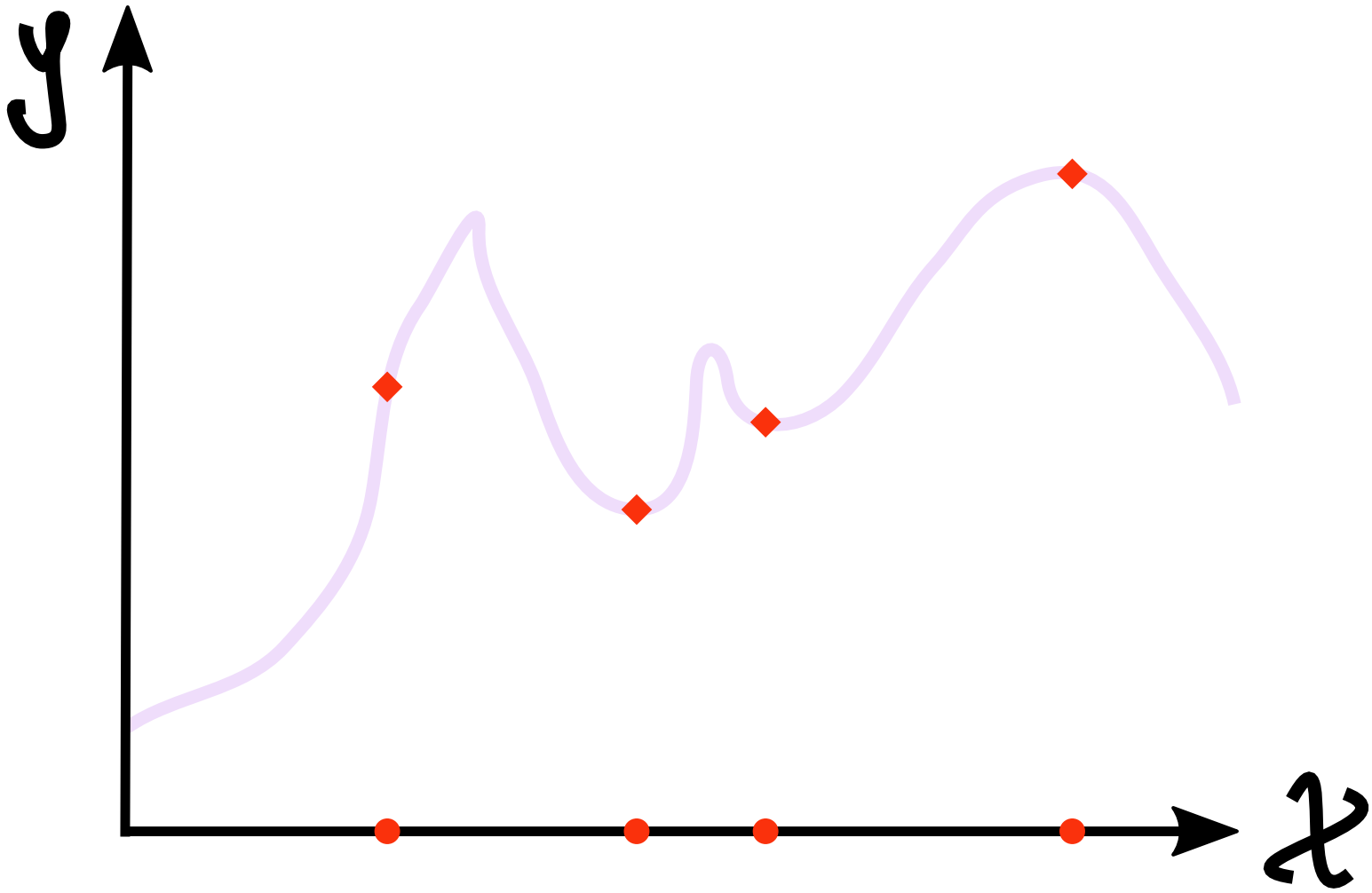
Assumption: Known Lipschitz constant? No!

Black box optimization



Assumption: Bayesian Gaussian priors? No!

Black box optimization



We want minimal assumptions! What is the smoothness of f?

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

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

Challenge: f has an **unknown smoothness**.

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

$$\mathbb{E}[y_t|x_t] = f(x_t) \qquad |y_t - x_t| \leq 1$$

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

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

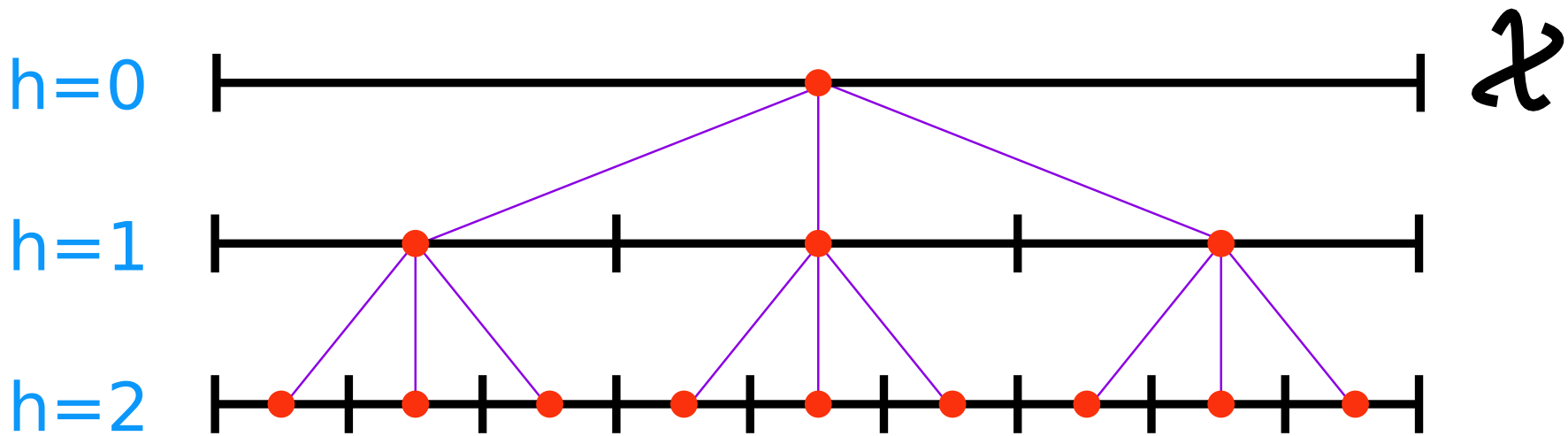
Minimal assumptions

- We want minimal assumptions.

- The smoothness of f is defined **with respect to** a partitioning $\mathcal{P}$ of the search space $\mathcal{X}$. **No metric!** **(Grill et al., 2015)**

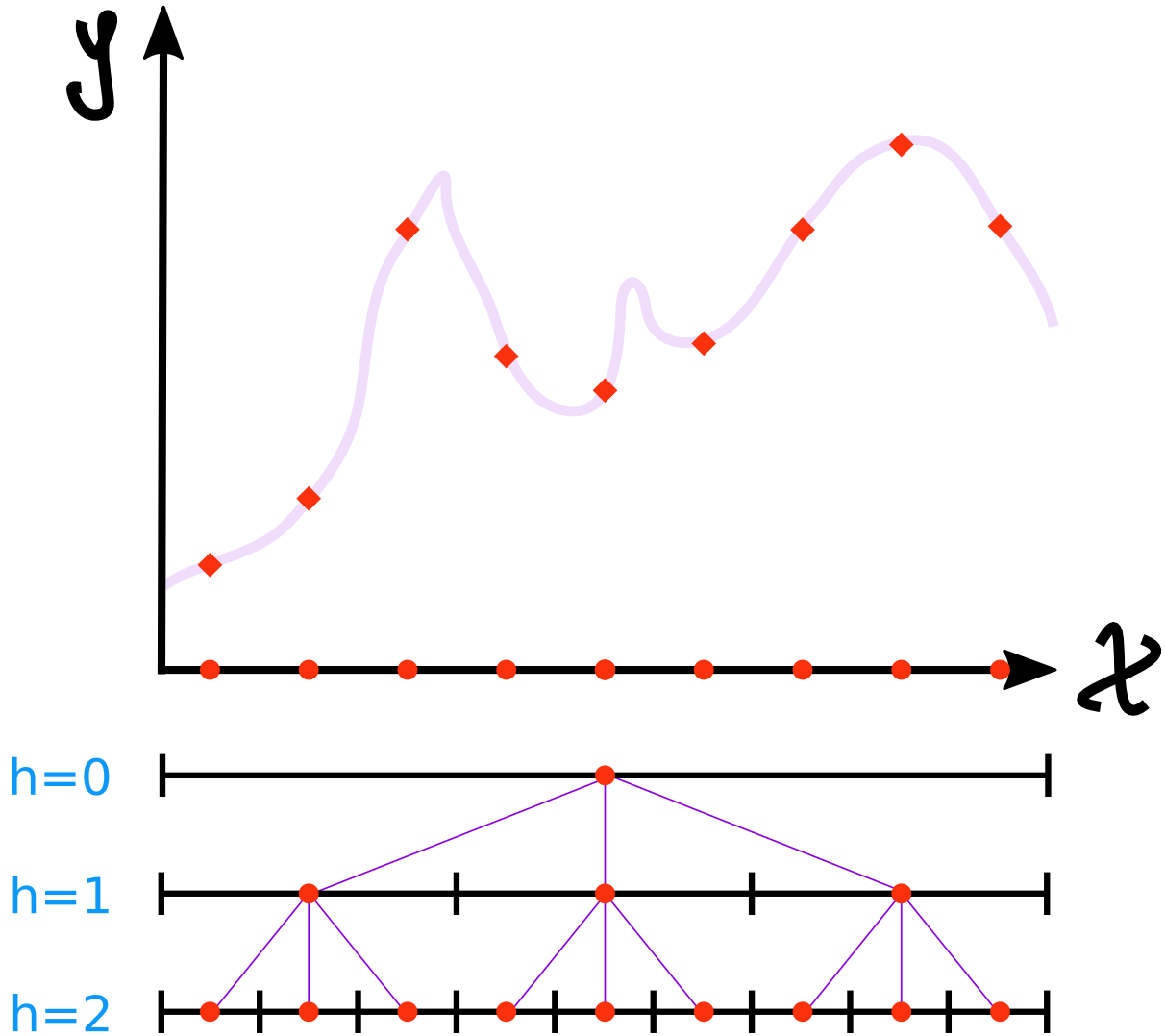
Minimal assumptions . Step 1 . Partitioning

- 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 $\mathcal{X}$.



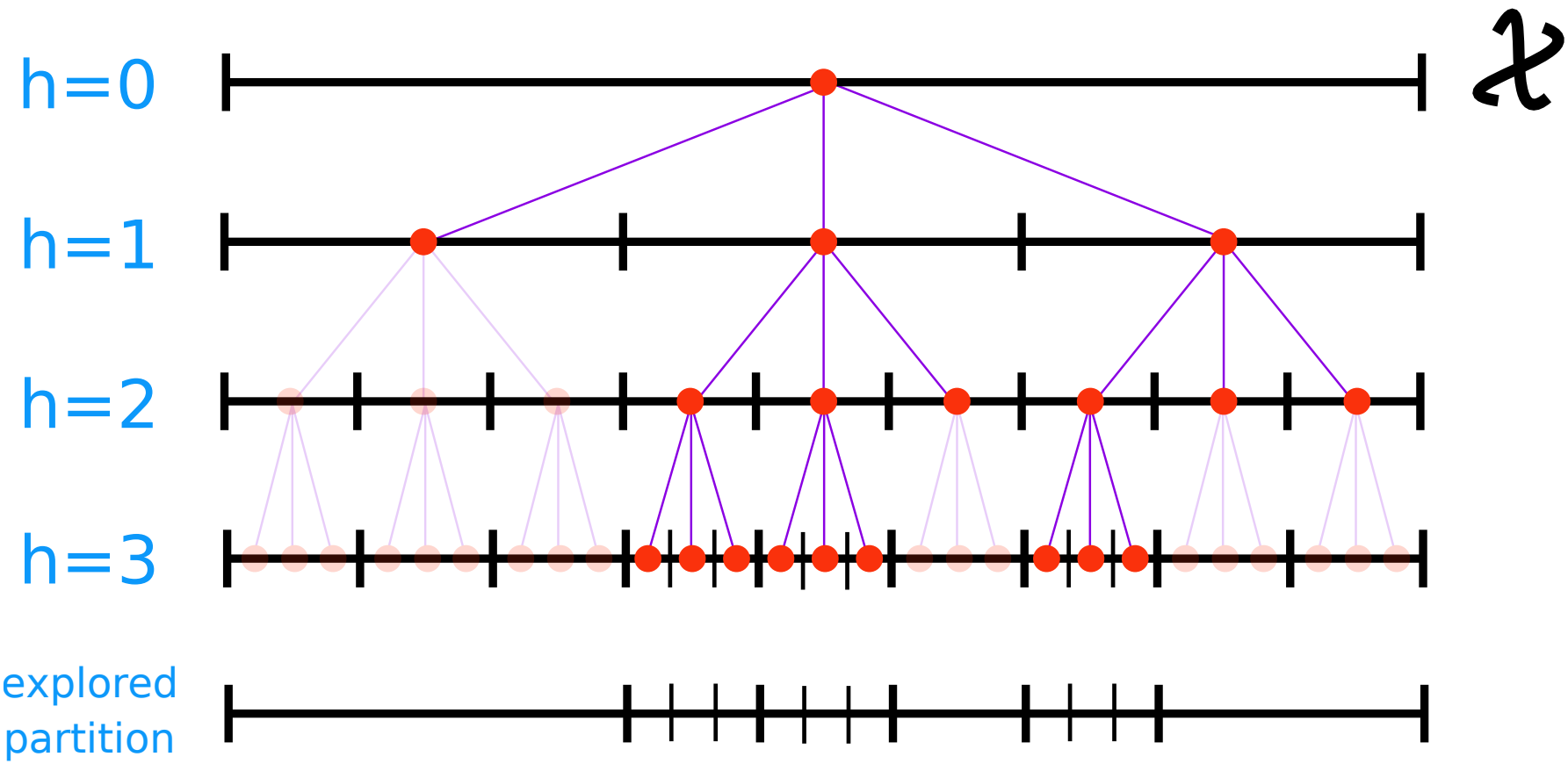
An example of partitioning in one dimension with $K = 3$.

Use the partitioning to explore (uniformly) f



Tree search

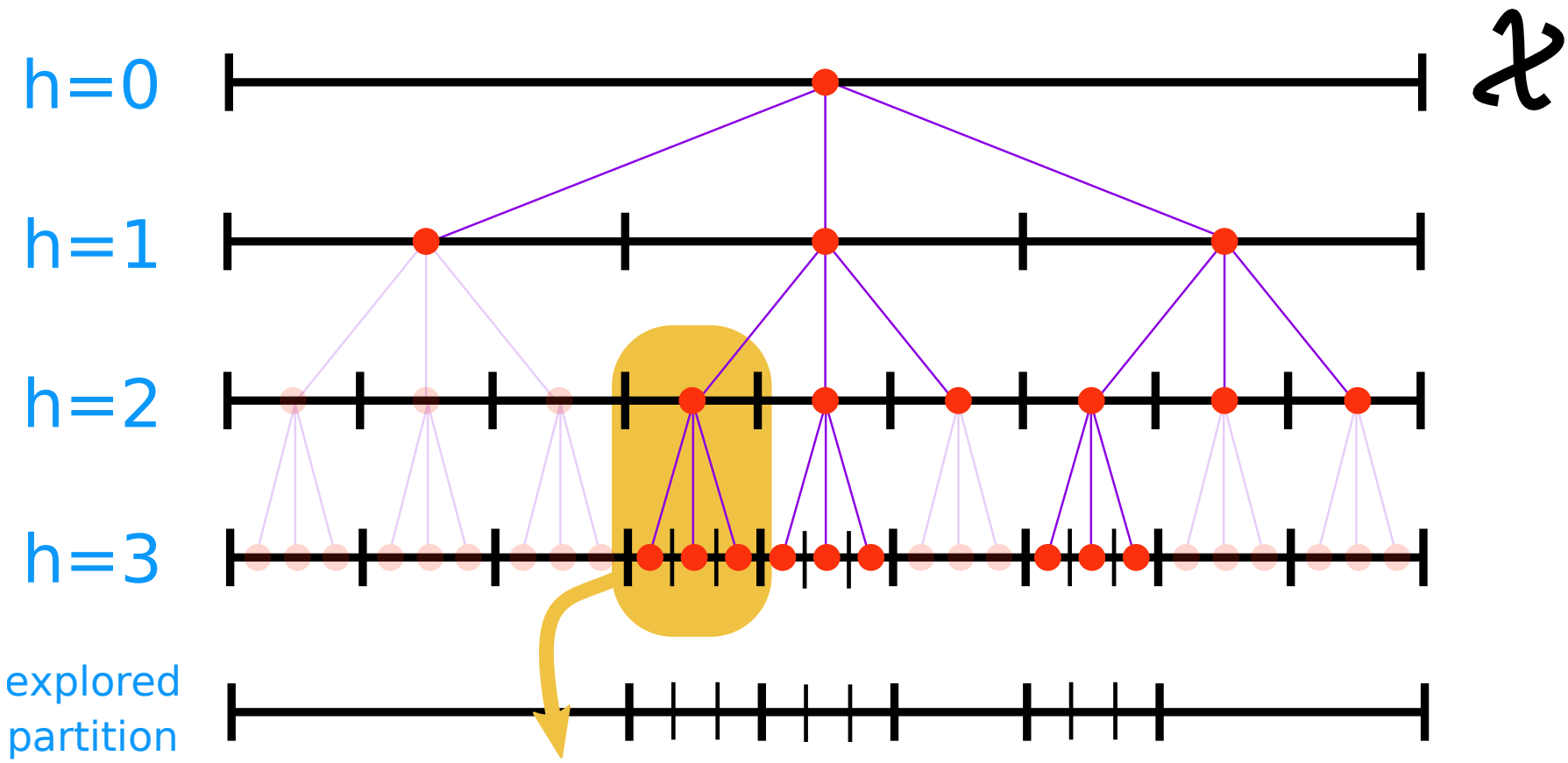
Optimizing becomes a **tree search** on the partition $\mathcal{P}$.



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

Tree search

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



Opening a cell means *evaluating* all its K children cells.

The assumption and the smoothness

Assumption (Grill et al., 2015)

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

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

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

The smoothness and the near-optimal dimension

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

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

- $\mathcal{N}_h(\varepsilon)$ is the number of cells $\mathcal{P}_{h,i}$ of depth h such that $\sup_{x \in \mathcal{P}_{h,i}} f(x) \geq f(x^\star) - \varepsilon$.
- $\mathcal{N}_h(3\nu\rho^h)$ explodes exponentially w.r.t d.

Previous work

Previous algorithms that depend on a metric:

smoothness	**global**	**local**
(ν, ρ) known	`Zooming`, `HOO`	`DOO`, `HCT`
(ν, ρ) unknown	`TaxonomyZoom`	`StoSOO`, `SOO`, `ATB`

We tackle **unknown** smoothness (ν, ρ) **without a metric**:

- `POO` **(Grill et al., 2015)** $\rightsquigarrow$ requires a base algorithm that has upper-bounded **cumulative regret**
- `GPO` (our algorithm) $\rightsquigarrow$ requires a base algorithm that has upper-bounded **simple regret**

How it works?

$\rightsquigarrow$ We run several instances of the base algorithm over $n/2$.

The GPO algorithm

Parameters: base algorithm $\mathcal{A}$, n, $\mathcal{P} = \{\mathcal{P}_{h,i}\}$, $\rho_{\max}$, $\nu_{\max}$

Initialization: $D_{\max} \leftarrow \ln K / \ln(1/\rho_{\max})$

Compute $N = \lceil (1/2) D_{\max} \ln((n/2)/\ln(n/2)) \rceil$

For $i = 1, \ldots, N$

- $s \leftarrow \left(\nu_{\max}, {\rho_{\max}}^{2N/(2i+1)}\right)$
- Run $\mathcal{A}(s)$ for $\lfloor n/(2N) \rfloor$ time steps $\rightarrow \widetilde{x}_s$

Output

The GPO algorithm

How it works?

$\rightsquigarrow$ We run several instances of the base algorithm over $n/2$.

$\rightsquigarrow$ We use another $n/2$ to do a **cross-validation**.

The GPO algorithm

Parameters: base algorithm $\mathcal{A}$, n, $\mathcal{P} = \{\mathcal{P}_{h,i}\}$, $\rho_{\max}$, $\nu_{\max}$

Initialization: $D_{\max} \leftarrow \ln K / \ln(1/\rho_{\max})$

Compute $N = \lceil (1/2) D_{\max} \ln((n/2)/\ln(n/2)) \rceil$

For $i = 1, \ldots, N$

- $s \leftarrow \left(\nu_{\max}, {\rho_{\max}}^{2N/(2i+1)}\right)$
- Run $\mathcal{A}(s)$ for $\lfloor n/(2N) \rfloor$ time steps $\rightarrow \widetilde{x}_s$
- Get $\lfloor n/(2N) \rfloor$ evaluations of $f(\widetilde{x}_s) \rightarrow$ average $V[s]$

$s^\star \leftarrow \arg\max_s V[s]$

Output $x(n) \leftarrow \widetilde{x}_{s^\star}$

The GPO algorithm

Theorem

If for all (ν, ρ) the $\mathcal{A}(\nu, \rho)$ algorithm has its simple regret bounded as

$$\mathbb{E}\left[r_n^{\mathcal{A}(\nu,\rho)}\right] \leq \alpha C\left((\log n/n)^{1/(d+2)}\right), \tag{1}$$

for any function f satisfying our minimal assumptions with parameters (ν, ρ), then there exists a constant β that is independent of $\nu_{\max}$ and $\rho_{\max}$ such that

$$\mathbb{E}\left[r_n^{\texttt{GPO}(\mathcal{A})}\right] \leq \beta D_{\max}(\nu_{\max}/\nu^\star)^{D_{\max}}\left((\log^2 n)/n)^{1/(d+2)}\right),$$

for any function f satisfying our minimal assumptions with parameters $\nu^\star \leq \nu_{\max}$ and $\rho^\star \leq \rho_{\max}$.

How it works?

$\rightsquigarrow$ The question now is whether there exists a base algorithm that has **simple regret** guarantee (1) under our minimal assumptions.

How it works?

$\rightsquigarrow$ The question now is whether there exists a base algorithm that has **simple regret** guarantee (1) under our minimal assumptions.

$\rightsquigarrow$ It is not clear whether HOO satisfies our needs. Worse, it is not even clear that HOO has any regret bound under our minimal assumptions, contrary to what is claimed by **Grill et.al. (2015)**.

How it works?

$\rightsquigarrow$ The question now is whether there exists a base algorithm that has **simple regret** guarantee (1) under our minimal assumptions.

$\rightsquigarrow$ It is not clear whether HOO satisfies our needs. Worse, it is not even clear that HOO has any regret bound under our minimal assumptions, contrary to what is claimed by **Grill et.al. (2015)**.

$\rightsquigarrow$ HCT does! **(Azar et.al., 2014)**: requires a refined analysis.

The GPO algorithm

Theorem

The simple regret of HCT after n function evaluations under our minimal assumptions satisfies

$$\mathbb{E}[r_n^{HCT(\nu,\rho)}] \leq \alpha C \left((\log n/n)^{1/(d+2)} \right).$$

Takeaway messages:

- A general meta-algorithm that adapts to unknown local smoothness that only requires the base algorithm to have some simple regret guarantee.
- Refined `HCT` analysis showing that it is a valid candidate.

Thank you!