Step size selection in Frank Wolf

Overview

Problem statement

$$\text{Minimize } f(\mathbf{x})$$

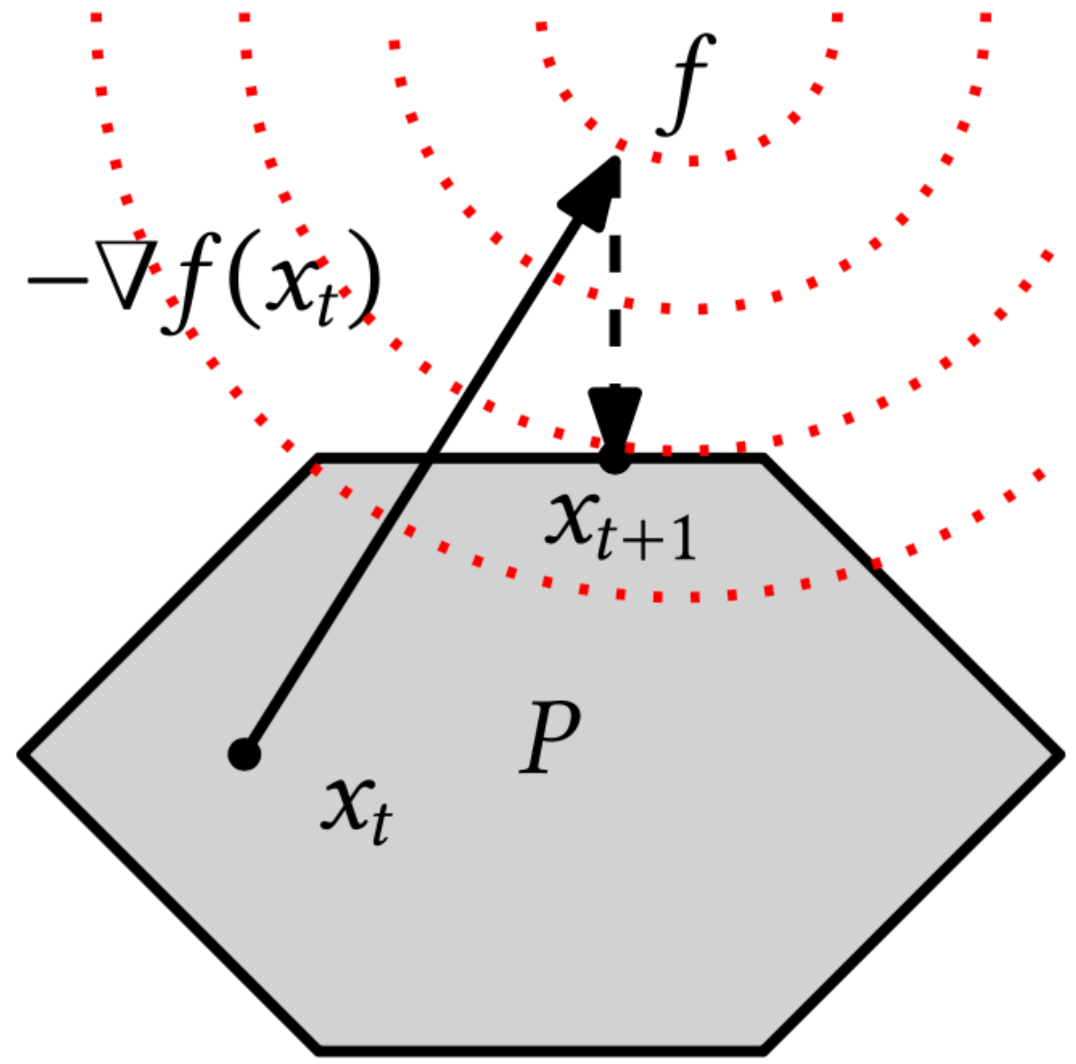
$$\text{Subject to } \mathbf{x} \in \mathscr{D}$$

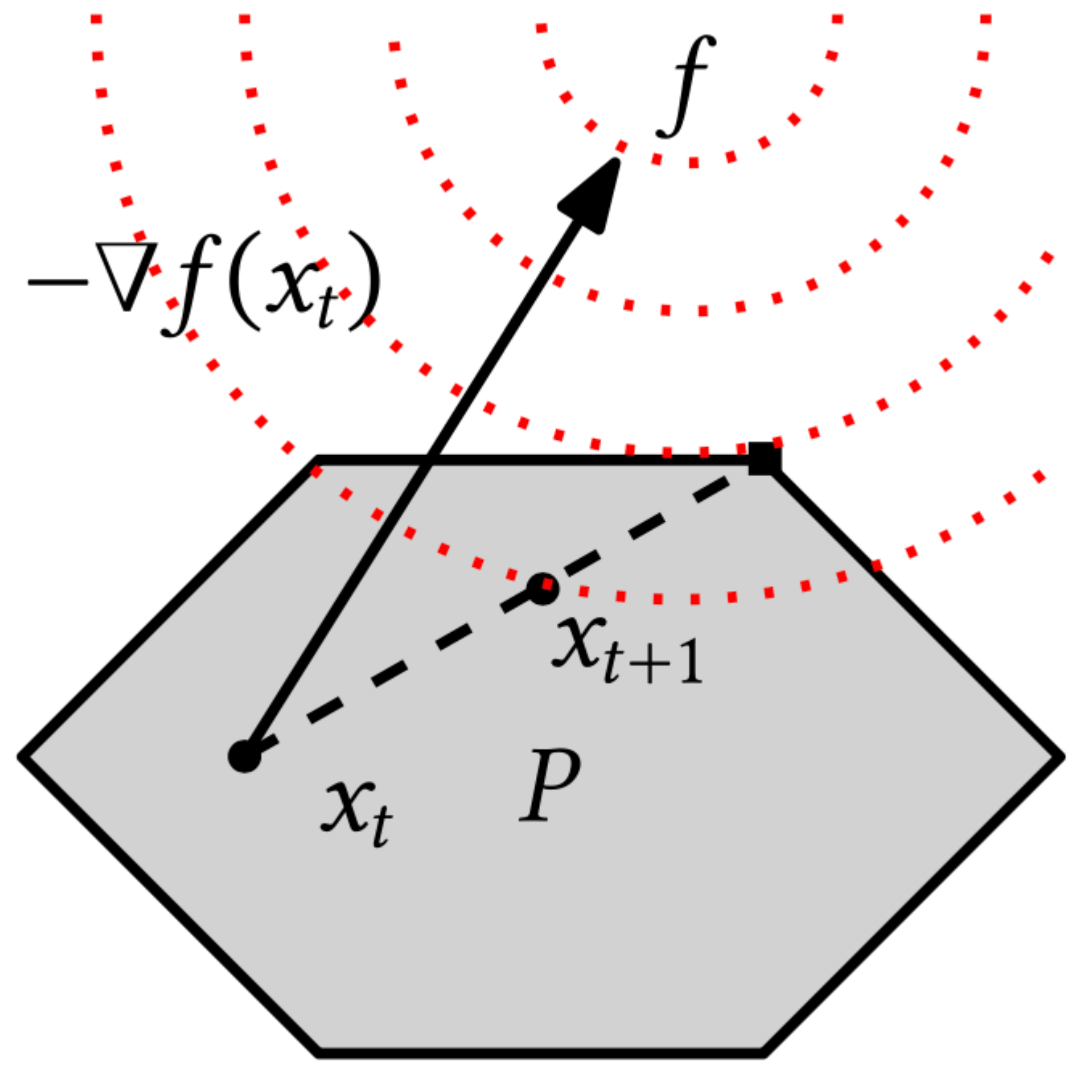
- $\mathscr{D}$ - **compact**, **convex** set in a vector space
- $f : \mathscr{D} \to \mathbf{R}$ is a **convex**, L**-smooth**, function

Linear Minimization Oracle

$$s = \arg\min_{s \in \mathscr{D}} \quad s^T \nabla f(x_k)$$

- Finds a vector s in feasible set $\mathscr{D}$, which aligns most with $\nabla f(x_k)$.
- Vector s has the largest projection on $-\nabla f(x_k)$. Usually a vertex of the domain.





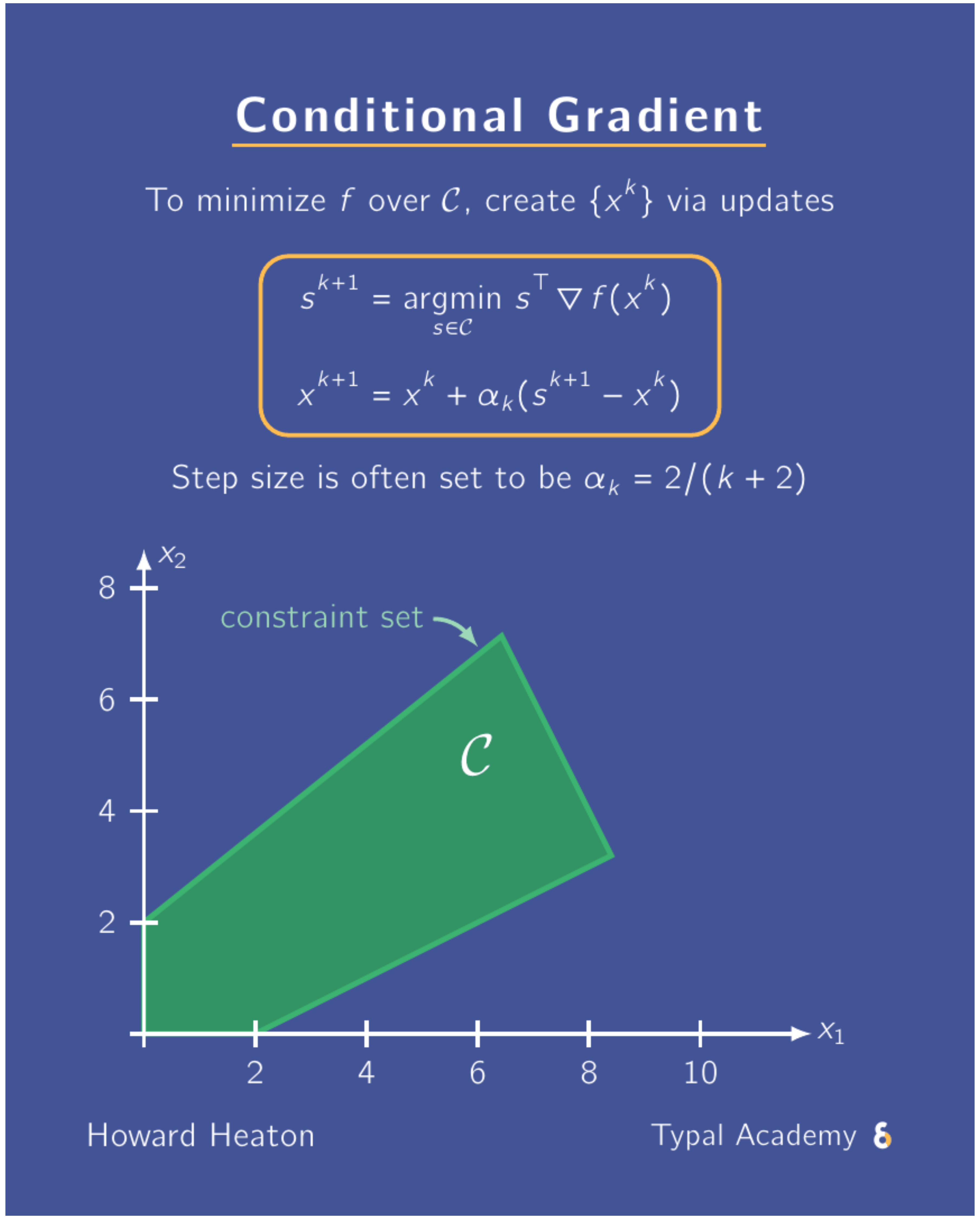
Algorithm

1. $s_k = \arg\min_{s \in \mathscr{D}} \ s^T \nabla f(x_k)$

2. $x_{k+1} = (1 - \gamma_k) x_k + \gamma_k s_k$

, where $\gamma_k \in [0,1]$ is a step-size.

- Both $x_k, s_k \in \mathscr{D}$. Convex combination of them is going to remain in the set. $x_{k+1} \in \mathscr{D}$



Properties

- Convergence rate $O(1/k)$
- No need to do projection step (linear optimization vs. quadratic)
- Solve high dimensional problems
- Sparse solutions

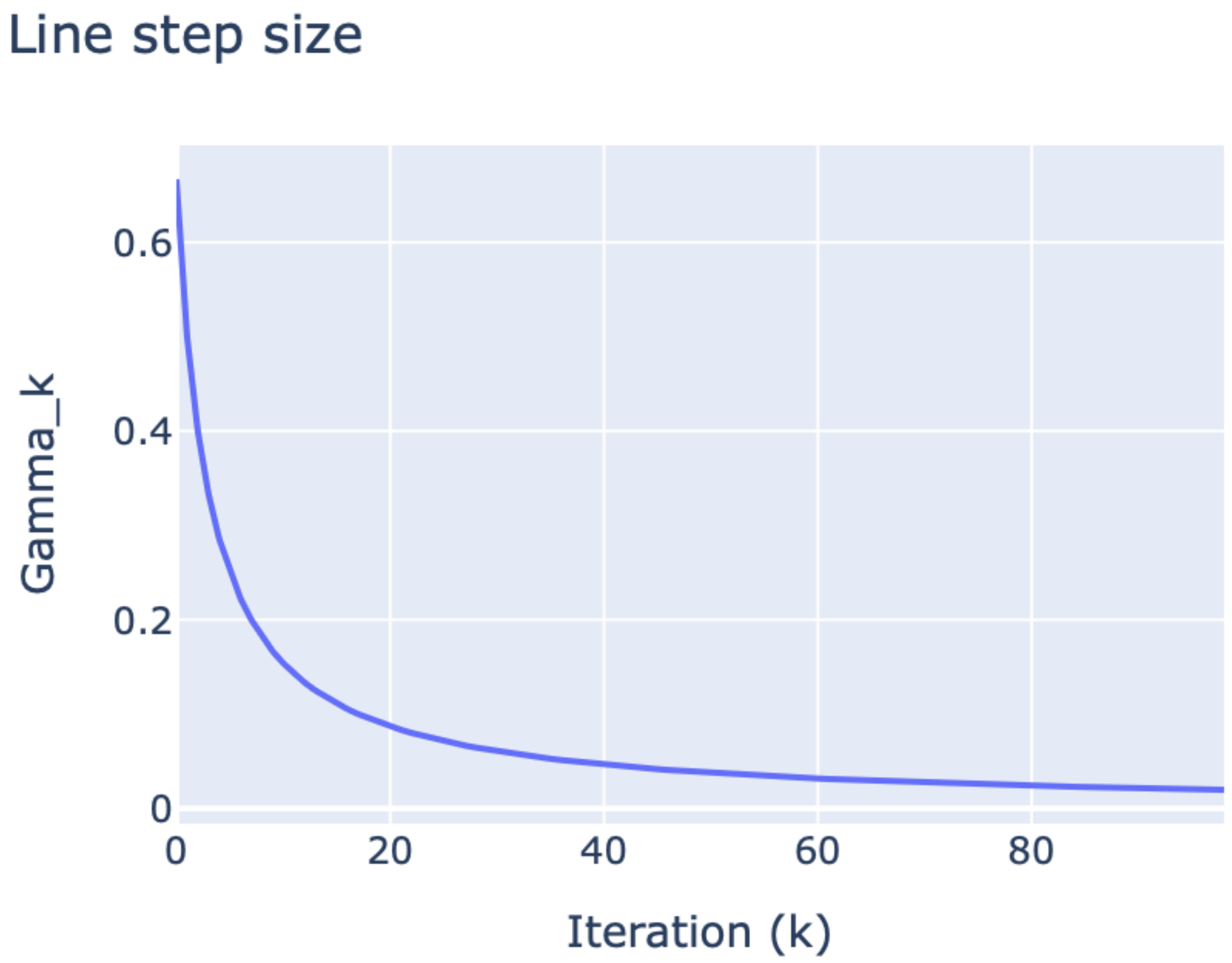
- Designed for smooth, convex f
- Poor performance near optimum
- Complex, non-linear boundaries increase computation costs.

Step size

Line step size

$$\gamma_k = \frac{2}{k+2}$$

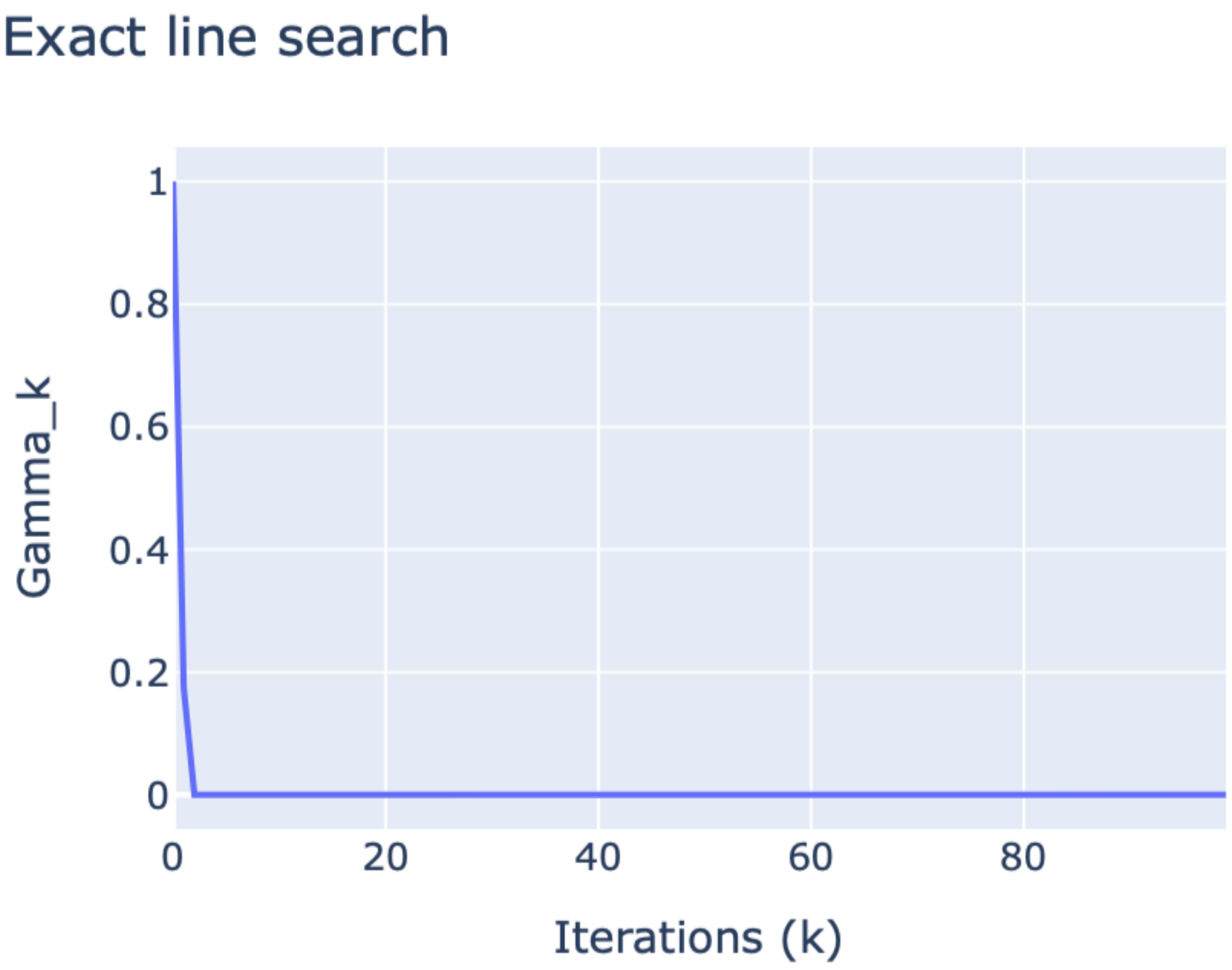
- Straightforward and cheap to compute.
- Slow convergence near optimum
- No function adaptation



Exact line-search

$$\gamma_k = \arg\min f((x_k + \gamma_k(s_k - x_k))$$

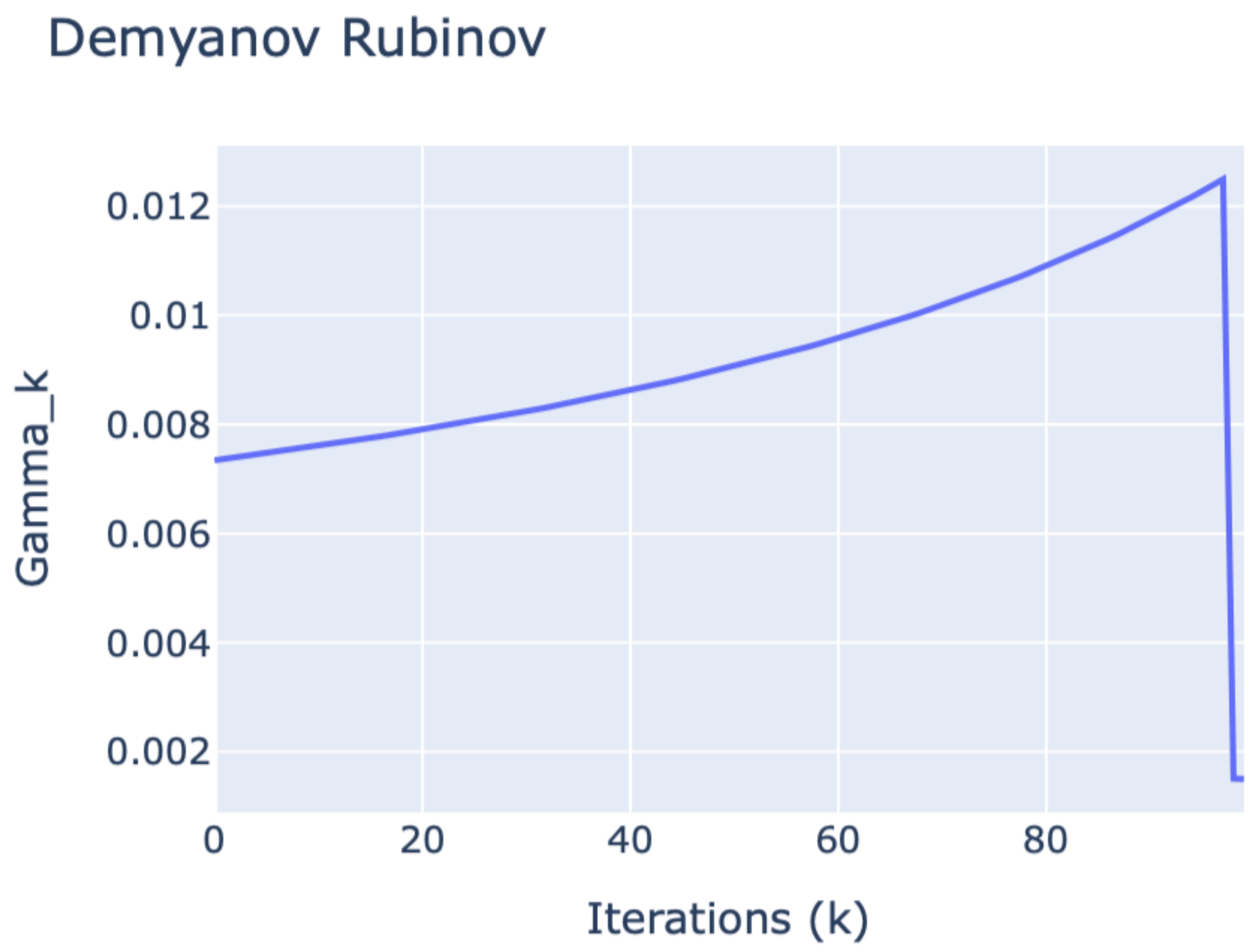
- Ensures highest decrease per iteration
- Costly optimization problem



Demyanov-Rubinov

$$\gamma_k = \min\{\frac{-\nabla f(x)^T(s_k - x_k)}{L\|s_k - x_k\|^2}, 1\}$$

- Goes to zero as we approach the optimum
- Responsive for geometry of f
- Require access to L
- Unstable for small denominator (near optimum)

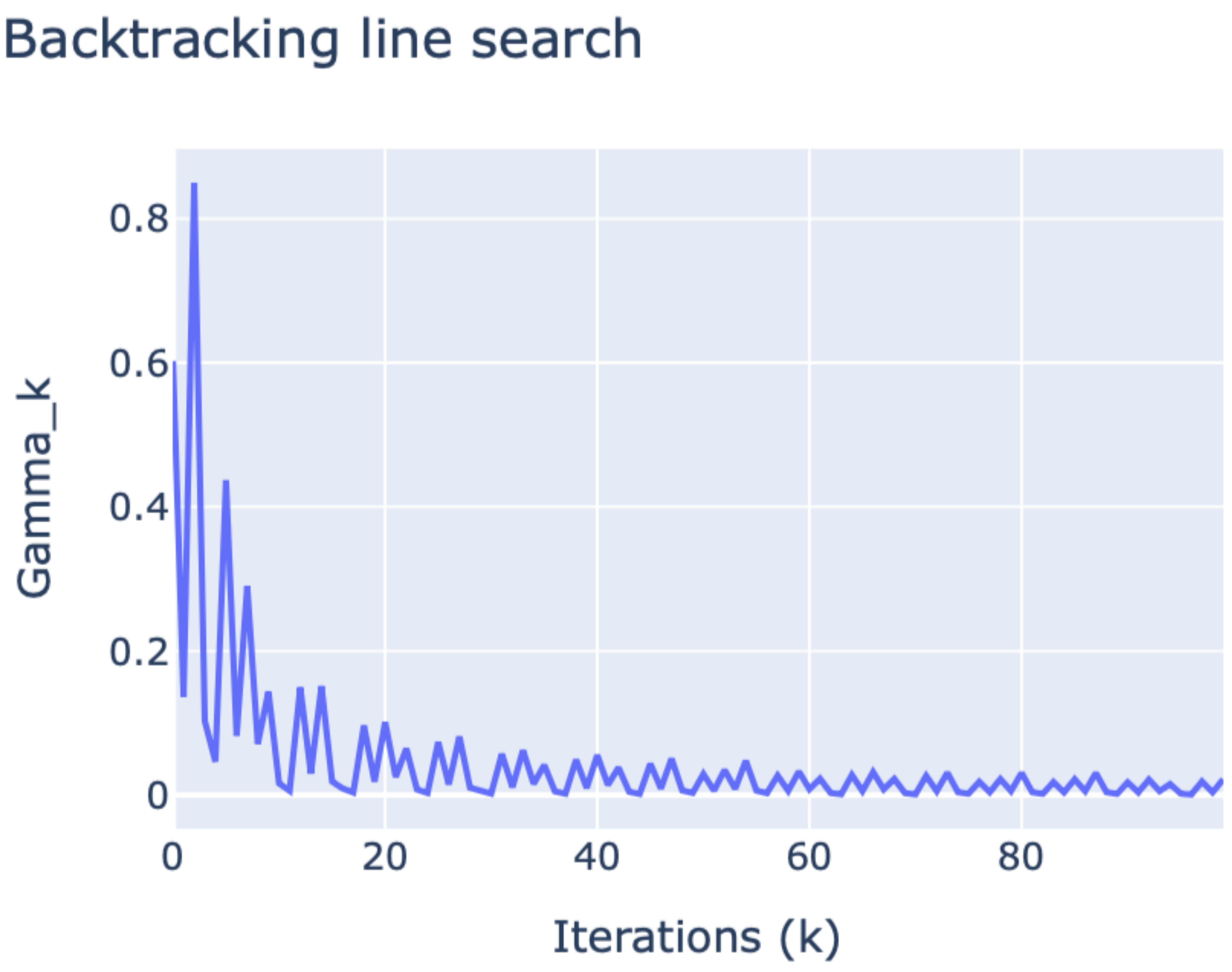


Backtracking line search

$$\gamma_k = \min\{\frac{-\nabla f(x)^T(s_k - x_k)}{M||s_k - x_k||^2}, 1\}$$

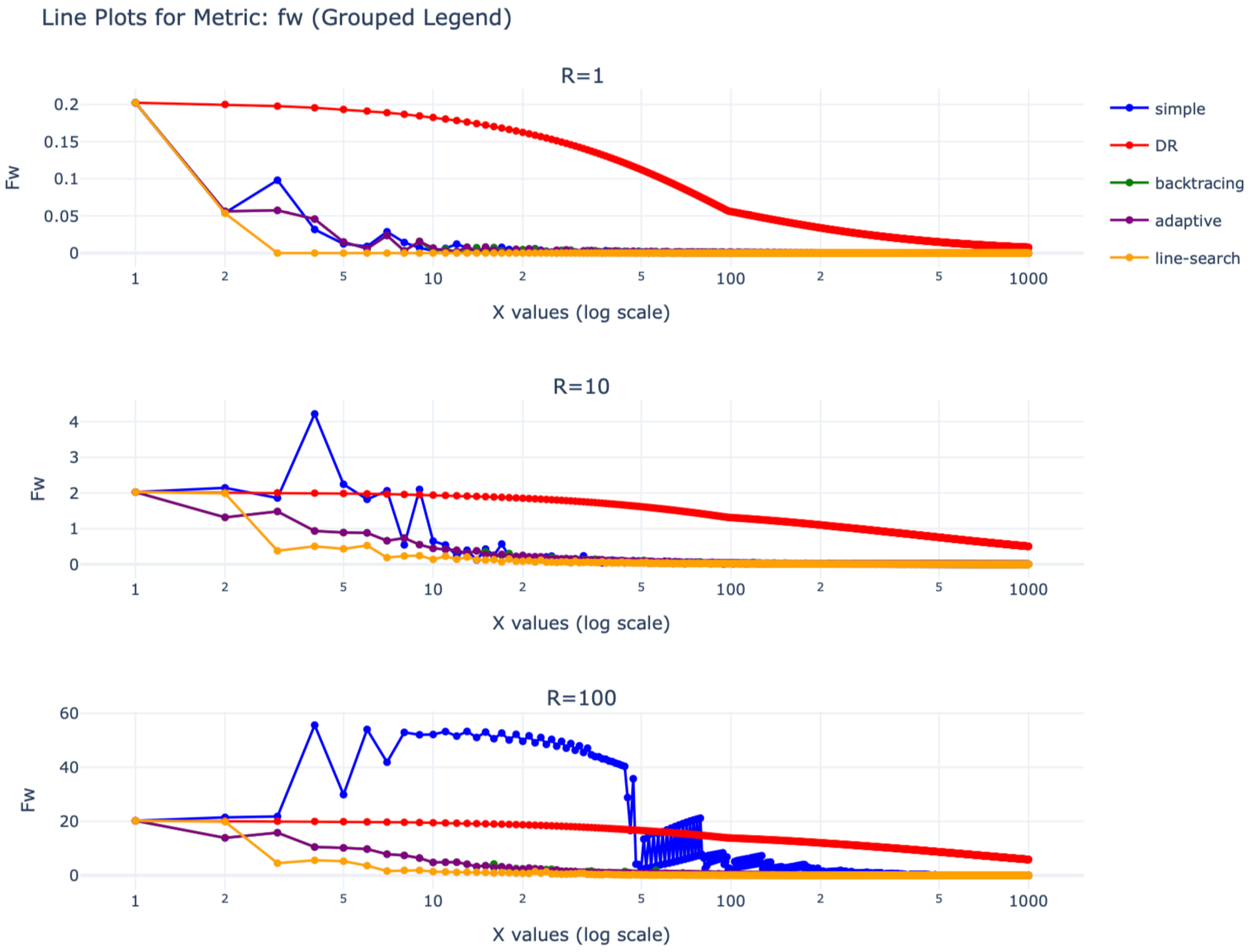
M_t is approximation of L

- Doesn't require L
- Adaptive and stable progress
- Require multiple evaluation of f



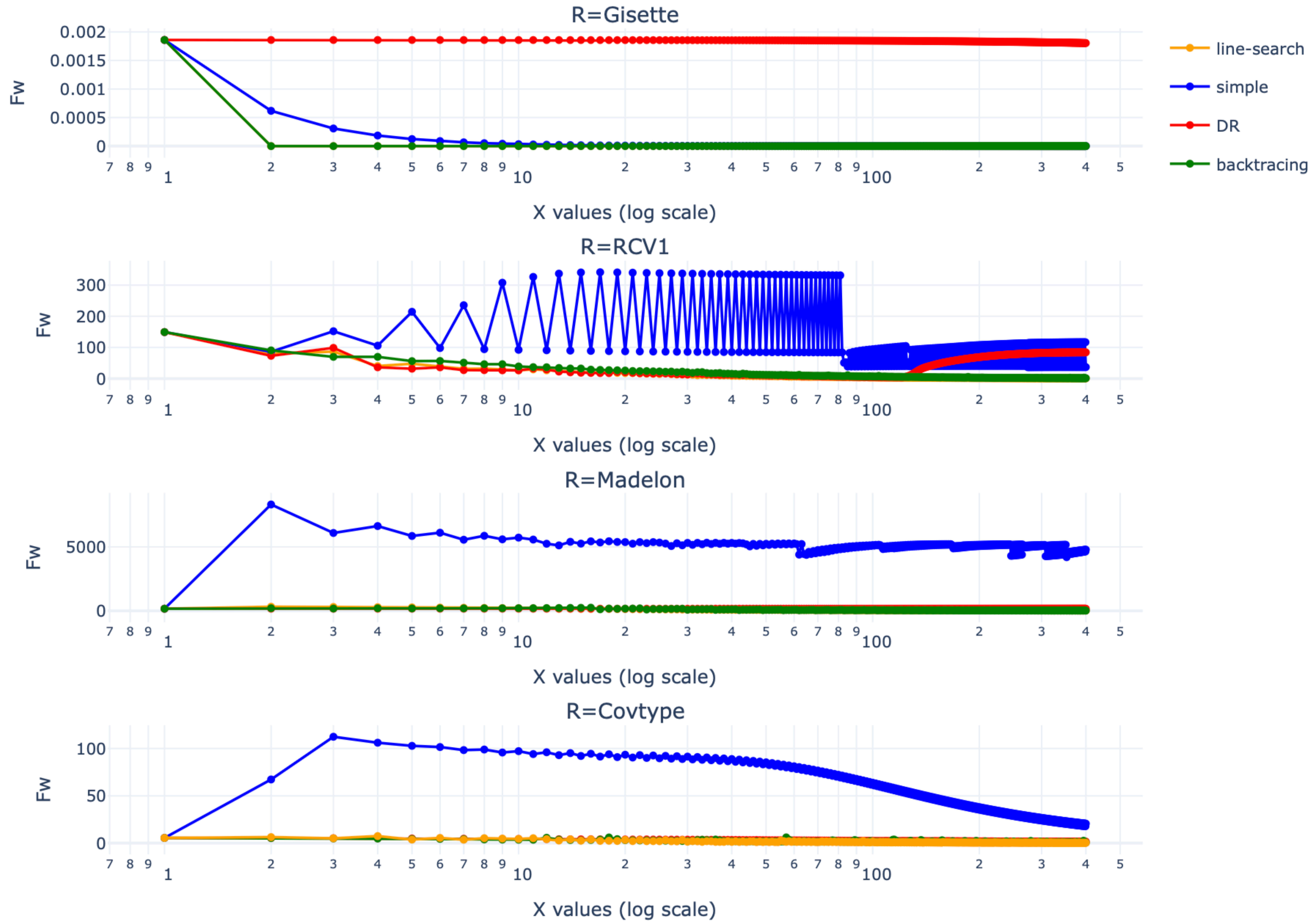
Experiments

HW (Mushrooms)



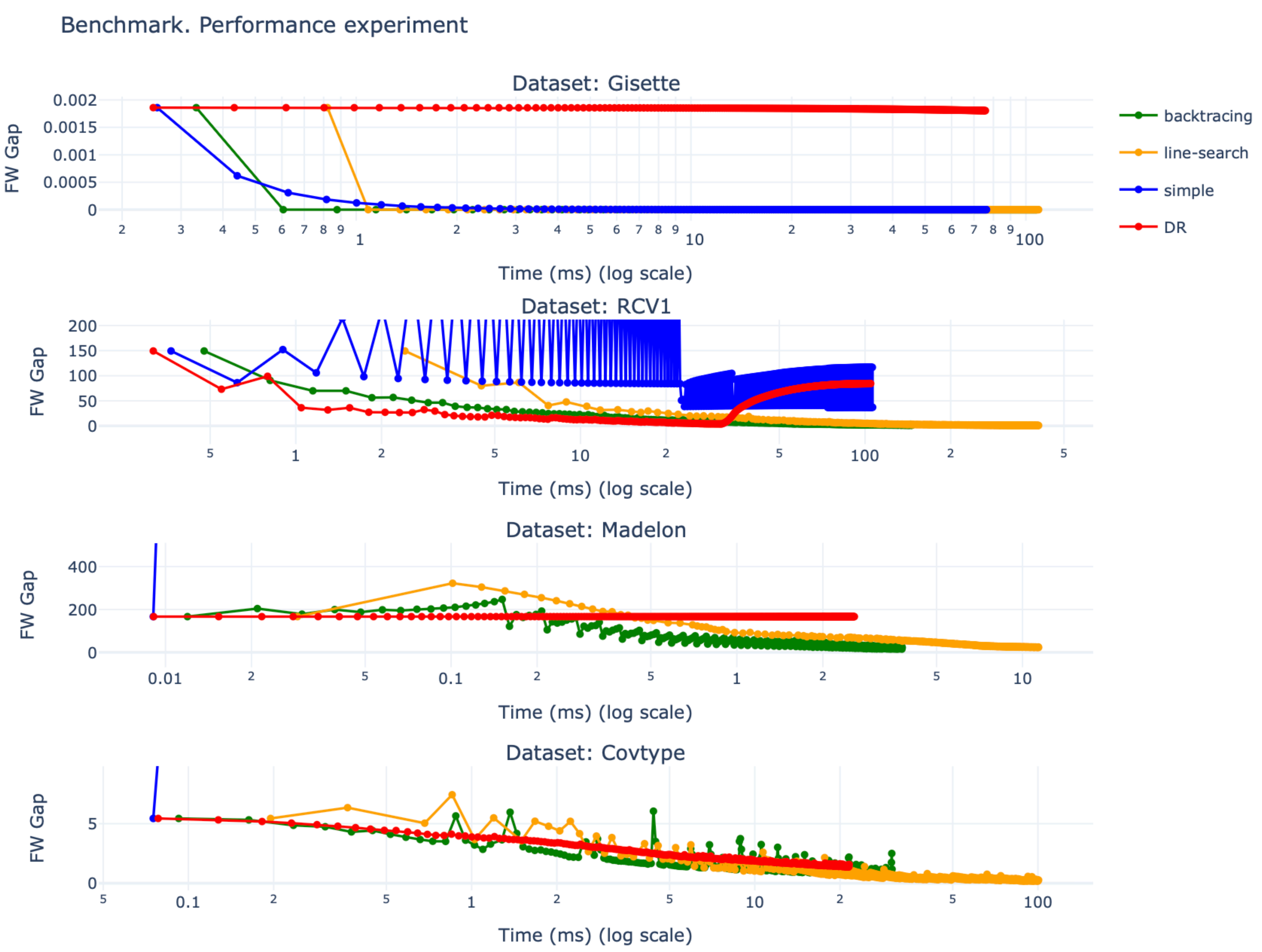
Benchmark

- Gisette: Binary classification on 5000 features $R = 6e^{-}3$
- RCV1: Binary classification on 47236 features $R = 2e^4$
- Madelon: Binary classification on 500 features $R = 20$
- Covtype: Binary classification on 54 features $R = 200$



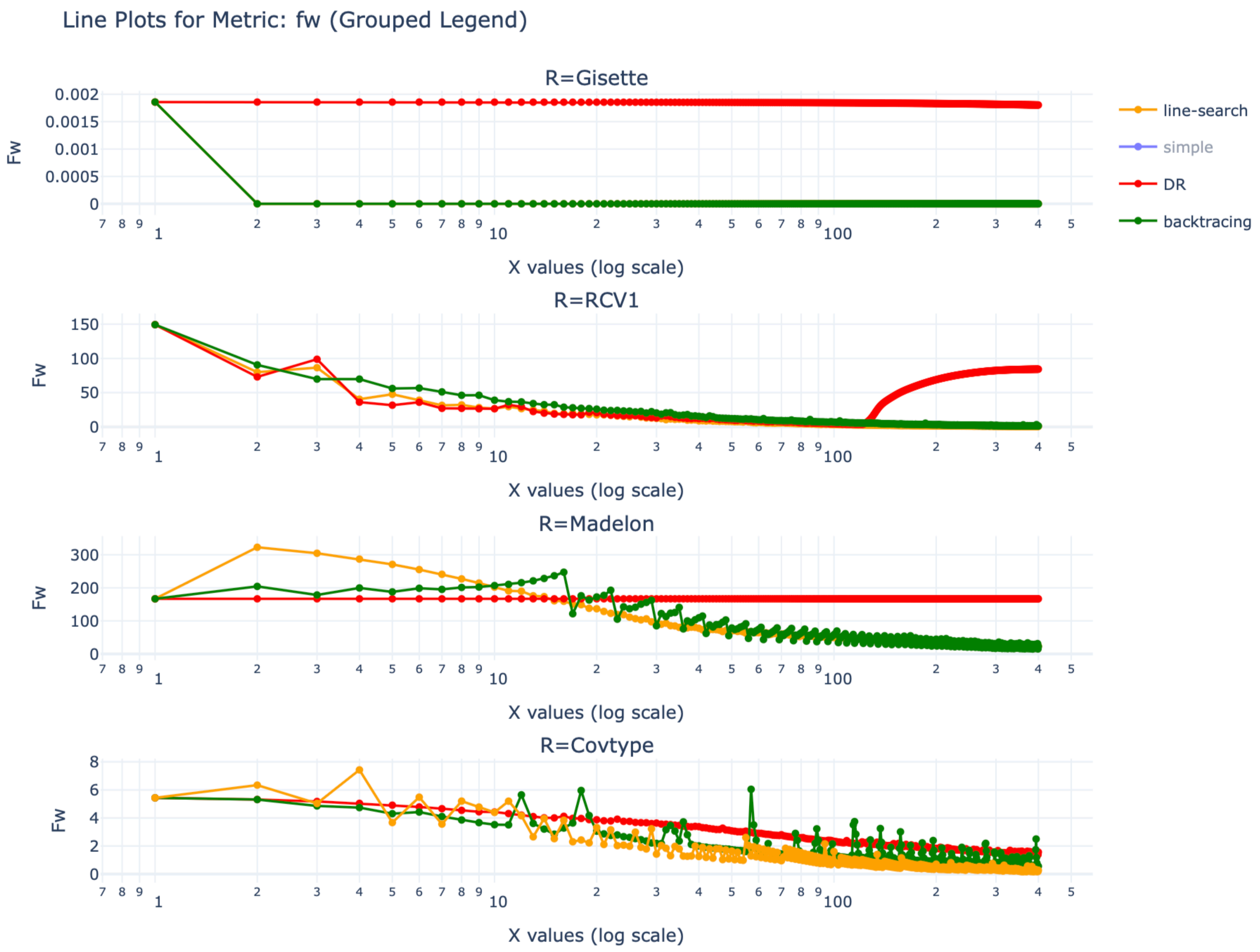
Performance

	Simple	DR	Backtracking	Line-Search
Gisette	1:16 5.24 it/s	1:15 5.27 it/s	1:49 3.66 it/s	1:49 3.67 it/s
RCV1	1:43 3.87 it/s	1:41 3.94 it/s	2:25 2.75 it/s	6:41 1 it/s
Madelon	0:02 157.22 it/s	0:02 159 it/s	0:03 107 it/s	0:11 35.84 it/s
Covtype	0:21 18.84 it/s	0:21 18.67 it/s	0:30 13.02 it/s	1:39 4.02 it/s

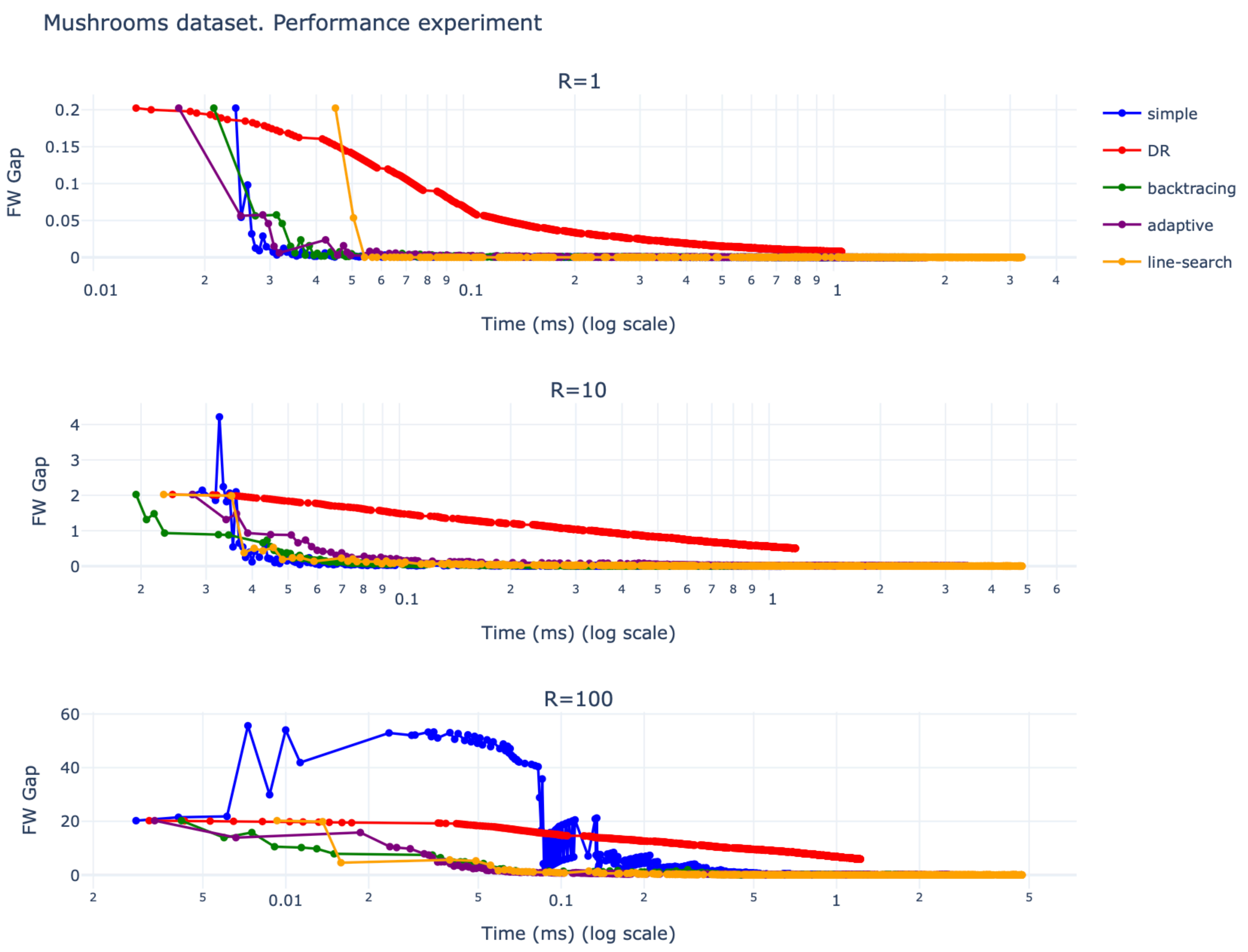


Benchmark. Performance experiment
Dataset: Gisette
Dataset: RCV1
Dataset: Madelon
Dataset: Covtype
FW Gap
Time (ms) (log scale)
backtracing
line-search
simple
DR

Appendix



Line Plots for Metric: fw (Grouped Legend)
R=Gisette
R=RCV1
R=Madelon
R=Covtype
Fw
X values (log scale)
line-search
simple
DR
backtracing



Mushrooms dataset. Performance experiment
R=1
R=10
R=100
FW Gap
Time (ms) (log scale)
simple
DR
backtracing
adaptive
line-search

Cancer

- Breast Cancer: