Blockwise Direct-Search Methods

LI Haitian

The Hong Kong Polytechnic University

Joint work with Cunxin Huang and Zaikun Zhang

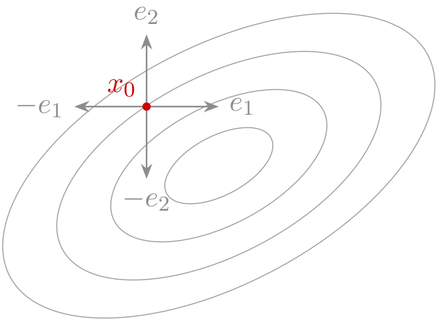
2026 优化算法软件与应用博士研究生前沿研讨会

Zeroth-order / Derivative-free optimization (DFO)

$$x \longrightarrow \boxed{f} \longrightarrow f(x)$$

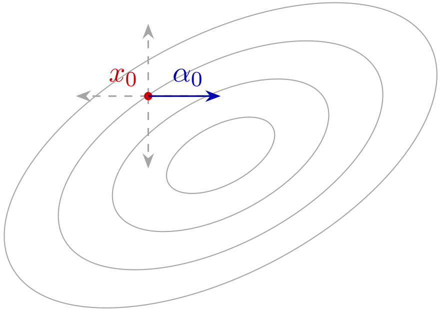
- Minimize f using function values, not derivatives.
- For many problems, $f(x)$ comes from running a simulation, not from an explicit formula.
- Function evaluations are typically the dominant cost: use as few as possible.

A simple DFO method: direct search



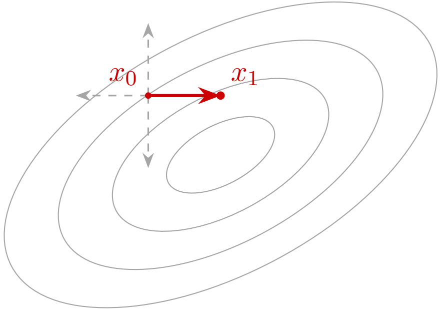
$\mathcal{D} = \{e_1, -e_1, e_2, -e_2\}$, and DS keeps one shared step size α_k.

A simple DFO method: direct search



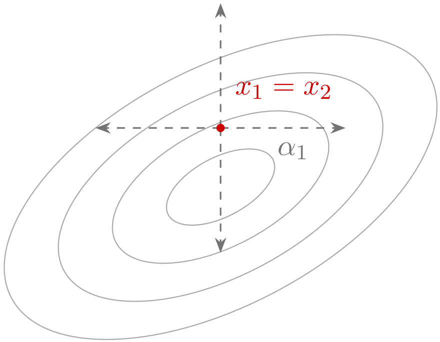
The e_1 trial gives enough decrease.

A simple DFO method: direct search



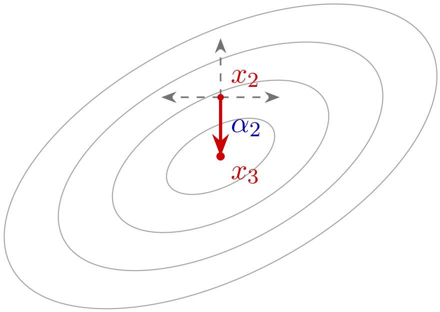
Accept and expand: $x_1 = x_0 + \alpha_0 e_1$, $\alpha_1 = 2\alpha_0$.

A simple DFO method: direct search



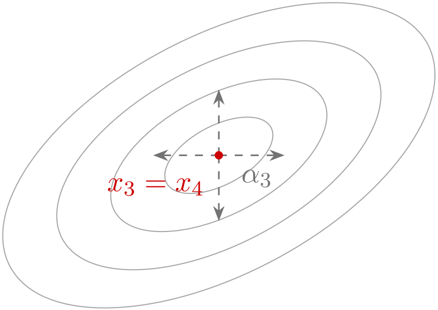
Fail and shrink: $x_2 = x_1$, $\alpha_2 = \frac{1}{2}\alpha_1$.

A simple DFO method: direct search



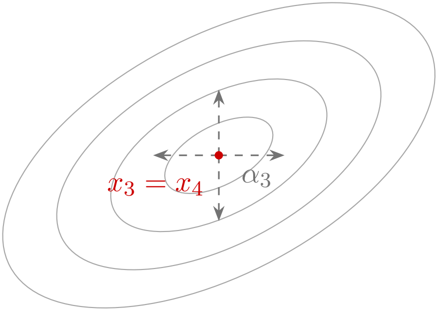
With the smaller step size, the $-e_2$ trial is accepted.

A simple DFO method: direct search



Fail and shrink again: $x_4 = x_3$, $\alpha_4 = \frac{1}{2}\alpha_3$.

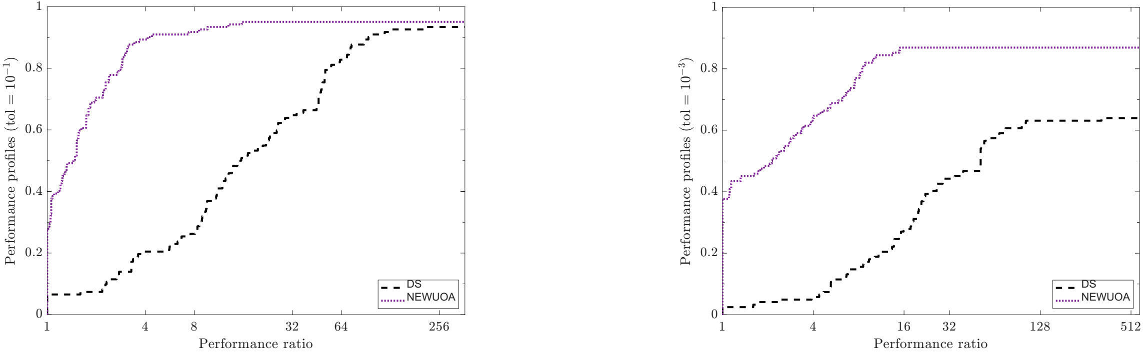
A simple DFO method: direct search



Fail and shrink again: $x_4 = x_3$, $\alpha_4 = \frac{1}{2}\alpha_3$.

It is not reasonable to have one single stepsize for all directions!

Unsatisfactory performance of classical direct search

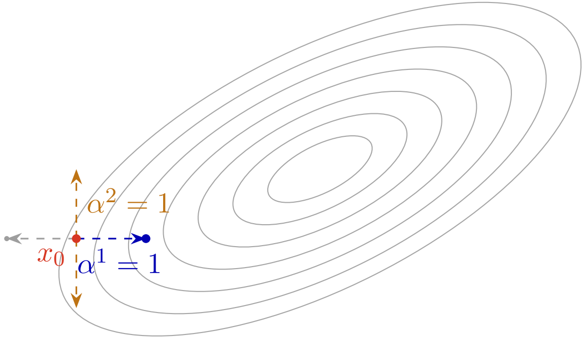


(a) $\tau = 10^{-1}$ (b) $\tau = 10^{-3}$

122 unconstrained CUTEst problems, $6 \leq n \leq 50$

- NEWUOA is a widely recognized benchmark in the DFO community.
- Classical DS leaves a clear performance gap.

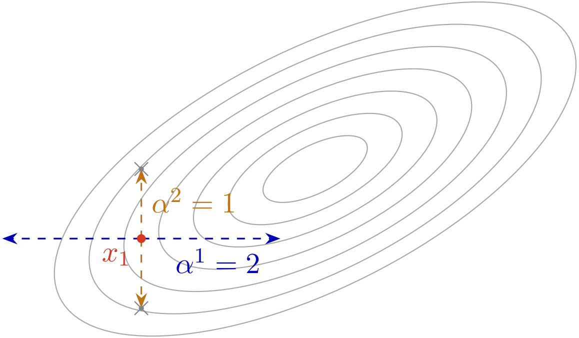
An improved direct-search method?



$\mathcal{D}^1 = \{e_1, -e_1\}, \qquad \mathcal{D}^2 = \{e_2, -e_2\}.$

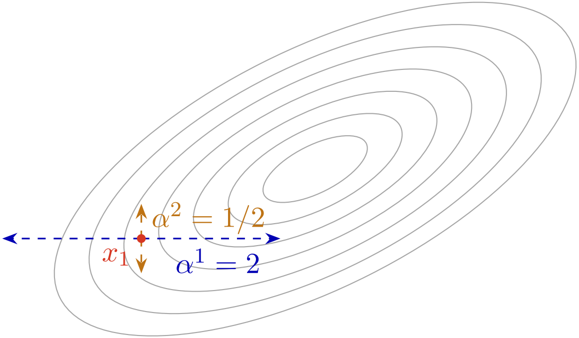
Block 1: the e_1 trial gives enough decrease.

An improved direct-search method?



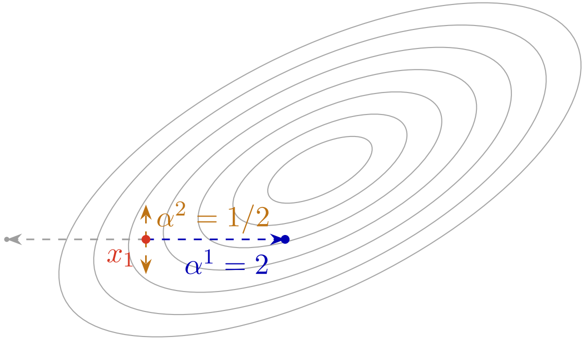
Block 2: both trials fail.

An improved direct-search method?



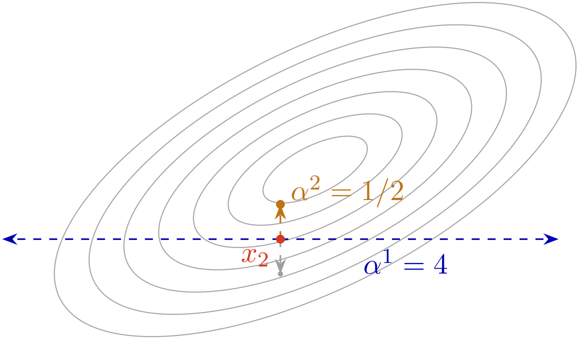
After one round: $\alpha^1 = 2$ and $\alpha^2 = 1/2$.

An improved direct-search method?



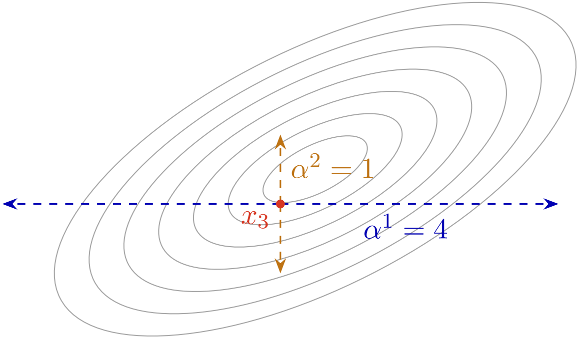
Block 1: the e_1 trial succeeds again.

An improved direct-search method?



Block 2: the e_2 trial gives enough decrease.

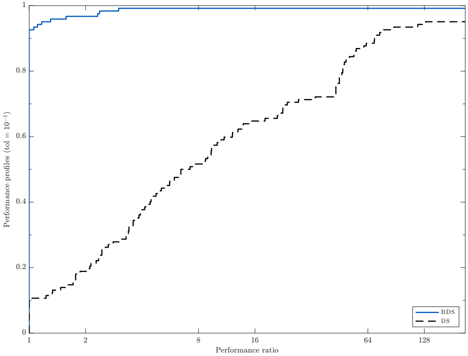
An improved direct-search method?



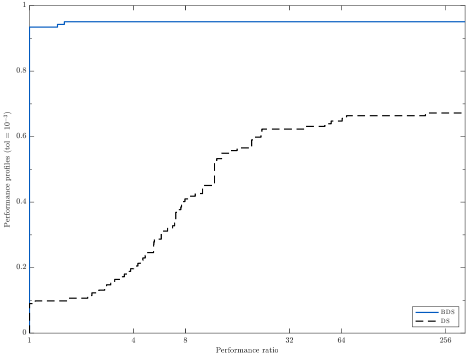
After two rounds: $\alpha^1 = 4$ and $\alpha^2 = 1$.

Each block learns its own scale.

Blocking substantially improves direct search



(a) $\tau = 10^{-1}$



(b) $\tau = 10^{-3}$

122 unconstrained CUTEst problems, $6 \leq n \leq 50$

A General Framework for Blockwise Direct Search (BDS)

Algorithm 1: Blockwise Direct Search

Input: $x_0 \in \mathbb{R}^n$, $0 < \theta < 1 \leq \gamma$, $\alpha_0^1, \ldots, \alpha_0^m > 0$, $c > 0$, and block direction sets $\mathcal{D}^1, \ldots, \mathcal{D}^m$.

for $k = 0, 1, \ldots$ **do**

    Select a block $i_k \in \{1, \ldots, m\}$

    if there exists $d \in \mathcal{D}^{i_k}$ such that $f(x_k + \alpha_k^{i_k} d) < f(x_k) - c(\alpha_k^{i_k})^2$ **then**

        Set $x_{k+1} = x_k + \alpha_k^{i_k} d$ and $\alpha_{k+1}^{i_k} = \gamma \alpha_k^{i_k}$

    else

        Set $x_{k+1} = x_k$ and $\alpha_{k+1}^{i_k} = \theta \alpha_k^{i_k}$

- Step sizes of unvisited blocks remain unchanged: $\alpha_{k+1}^j = \alpha_k^j$ for every $j \neq i_k$.

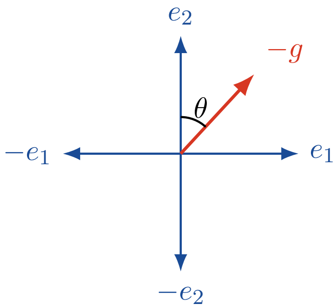
Flexible block selection

Every block is selected infinitely often.

$$\#\{k : i_k = i\} = \infty \qquad \text{for every block } i.$$

- No frequency or order is prescribed.
- Blocks may be selected deterministically or randomly.

Directional requirement: $\bigcup_i \mathcal{D}^i$ positively spans $\mathbb{R}^n$



- Every $v \in \mathbb{R}^n$ is a nonnegative combination of directions in the union: $v = \sum_r \lambda_r d^r$, $\lambda_r \geq 0$.
- Hence, for every nonzero gradient g, some direction in the union satisfies $g^\top d < 0$: a descent direction.
- **Positive spanning sets are abundant:** every basis $\{b_1, \ldots, b_n\}$ generates $\{\pm b_1, \ldots, \pm b_n\}$.

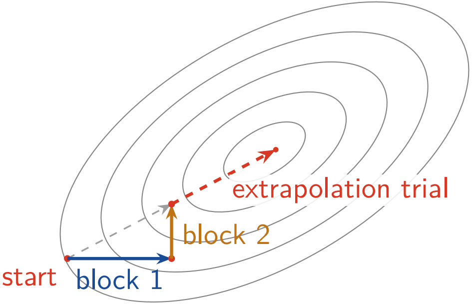
Convergence: L-smooth, convex, bounded initial level set

- The iterates have a cluster point $\bar{x}$.
- Every block step size vanishes: $\alpha_k^i \to 0$ for every block i.
- If $\bar{x}$ were nonstationary, convexity and Bregman distance would imply that some block step size is eventually nondecreasing.

  This contradicts $\alpha_k^i \to 0$.

Directional extrapolation in BDS: a 2D illustration

For $n = 2$, visit $\{e_1, -e_1\}$ and $\{e_2, -e_2\}$ alternately.

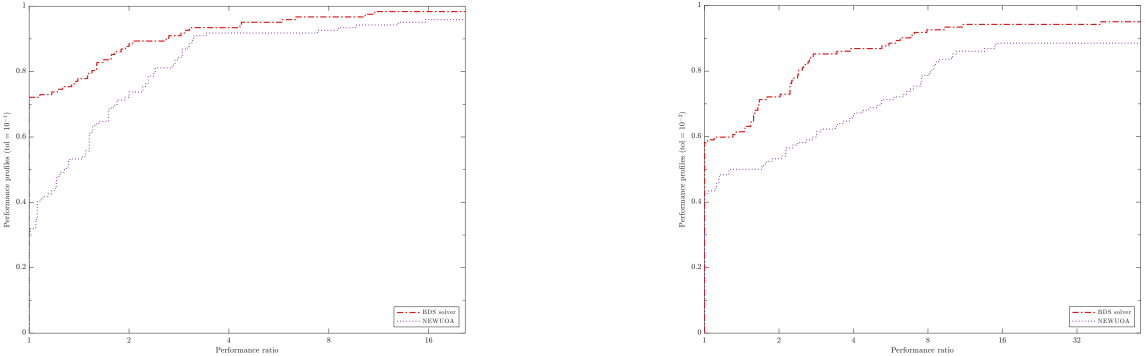


- Optional extrapolation trial: continue along the combined displacement of the two block moves.
- No line search along the extrapolation direction is performed.

Other practical mechanisms

- Initialization: choose initial step sizes blockwise from the scale of x_0.
- Stopping: reuse existing trial values without extra function evaluations.
- Failed simulations: if a trial errors or returns NaN, assign $+\infty$ and continue.

Performance of the complete BDS solver



(a) $\tau = 10^{-1}$ (b) $\tau = 10^{-3}$

122 unconstrained CUTEst problems, $6 \leq n \leq 50$

Noise model and experimental setting

Observed function value:

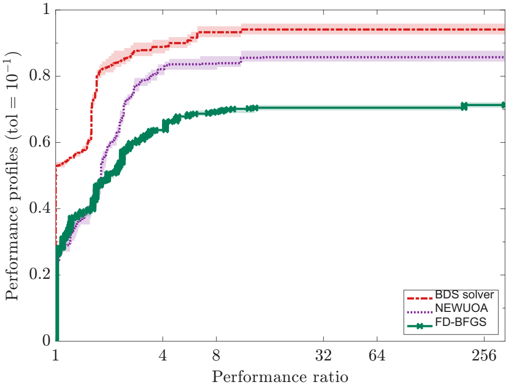
$$\widetilde{f}_\sigma(x) = f(x) + \max\{|f(x)|, 1\}\,\sigma Z, \qquad Z \sim \mathcal{N}(0,1).$$

Z is sampled at every function evaluation.

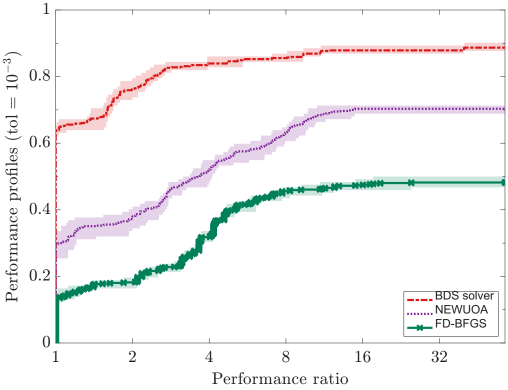
In our experiments:

- problem set: 122 unconstrained CUTEst problems
- dimensions: $6 \leq n \leq 50$
- noise level: $\sigma = 10^{-3}$
- budget: $500n$ function evaluations
- runs per noisy problem: 5

Performance of the BDS solver under noise



(a) $\tau = 10^{-1}$



(b) $\tau = 10^{-3}$

Adaptive forward-difference step for FD-BFGS: $h(x) = \sqrt{\sigma \max\{|\widetilde{f}(x)|, 1\}}$.

Conclusions

- Independent block scales substantially improve direct search.
- For smooth convex objectives, BDS converges under very general algorithmic conditions.
- The complete BDS solver is practical and competitive.
- BDS is robust to noise without specialized noise handling.

BDS on GitHub

Conclusions

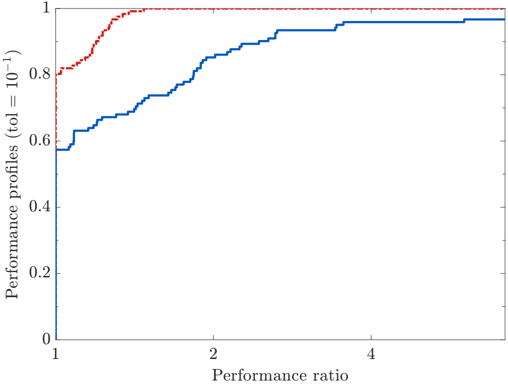
- Independent block scales substantially improve direct search.
- For smooth convex objectives, BDS converges under very general algorithmic conditions.
- The complete BDS solver is practical and competitive.
- BDS is robust to noise without specialized noise handling.

BDS on GitHub

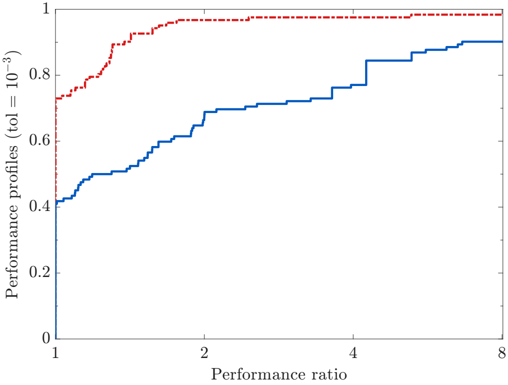
Thank you!

Appendix

Appendix: acceleration ablation



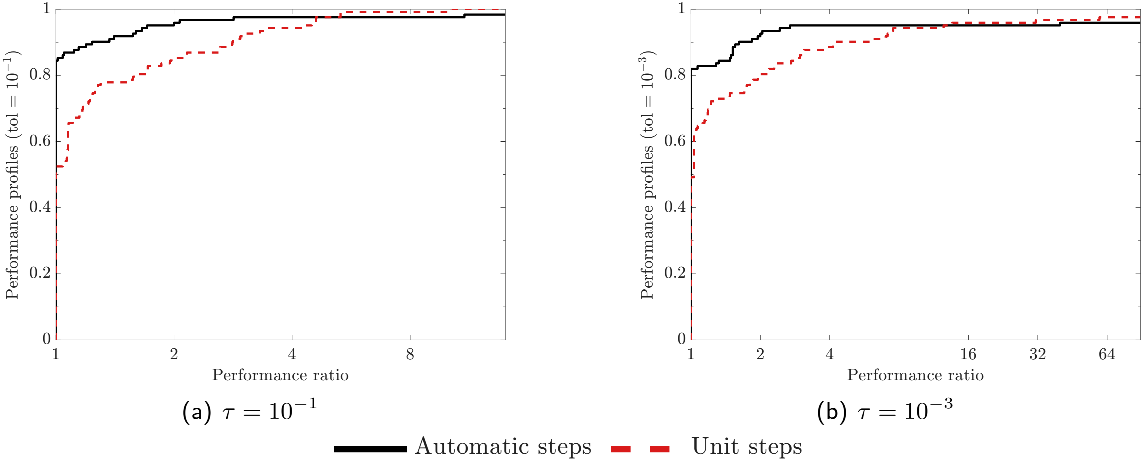
(a) $\tau = 10^{-1}$



(b) $\tau = 10^{-3}$

Accelerated BDS   BDS

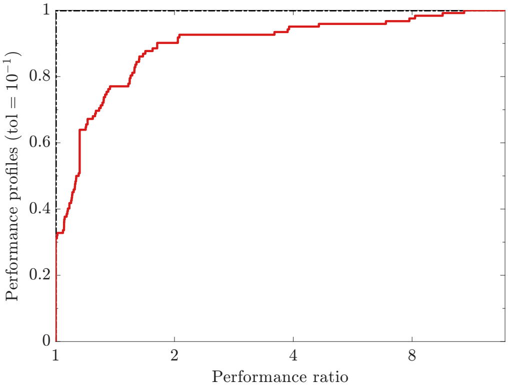
Appendix: initial step-size selection



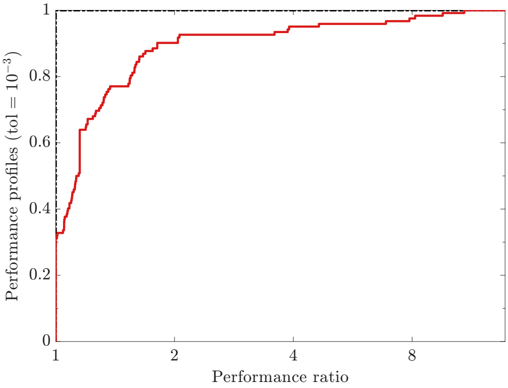
(a) $\tau = 10^{-1}$

(b) $\tau = 10^{-3}$

Appendix: termination without extra evaluations



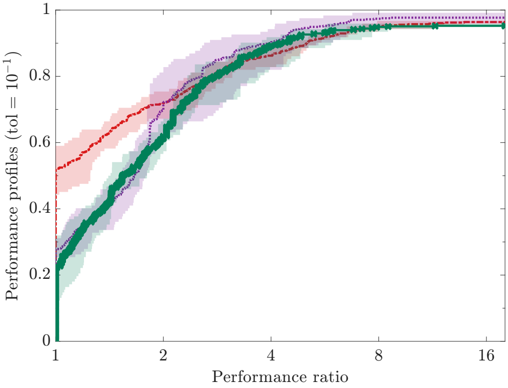
(a) $\tau = 10^{-1}$



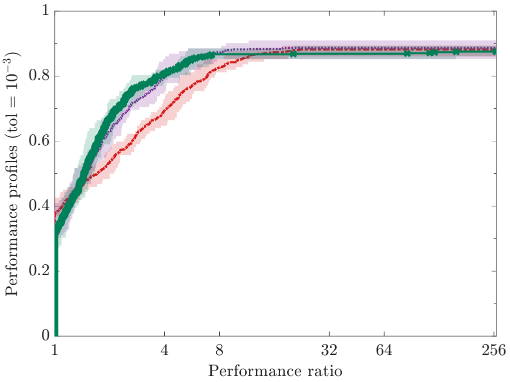
(b) $\tau = 10^{-3}$

Additional stopping — No additional stopping

Appendix: performance under orthogonal transformations



(a) $\tau = 10^{-1}$

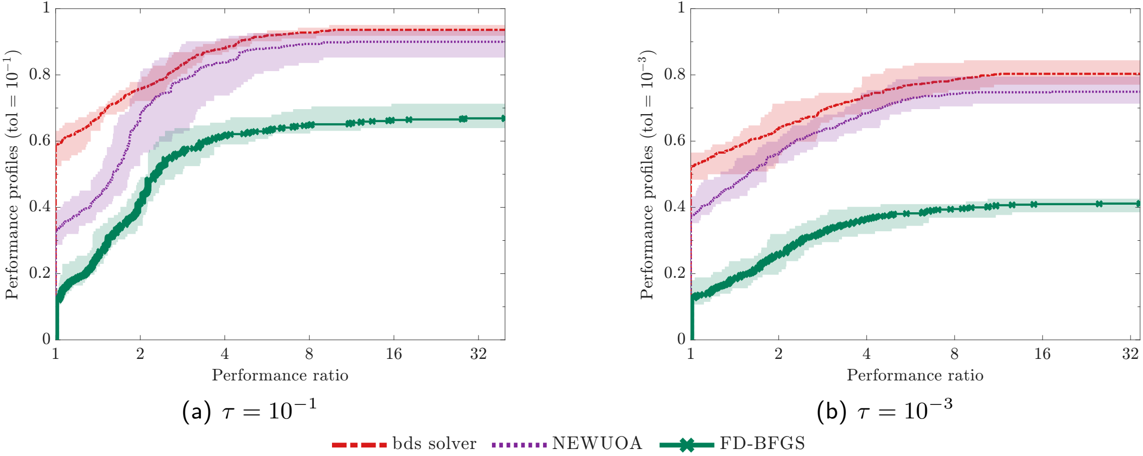


(b) $\tau = 10^{-3}$

bds solver NEWUOA FD-BFGS

Appendix: transformed problems under noise

Noise level: $\sigma = 10^{-3}$



(a) $\tau = 10^{-1}$ (b) $\tau = 10^{-3}$

Appendix: selected references

- C. Huang, H. Li, and Z. Zhang, *Blockwise Direct Search Methods*, manuscript, 2026.
- M. J. D. Powell, *The NEWUOA software for unconstrained optimization without derivatives*, in *Large-Scale Nonlinear Optimization*, 2006.
- A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, SIAM, 2009.
- J. J. Moré and S. M. Wild, *Benchmarking derivative-free optimization algorithms*, SIAM J. Optim., 2009.
- A. Beck and L. Tetruashvili, *On the Convergence of Block Coordinate Descent Type Methods*, SIAM J. Optim., 2013.