Secant Line Search for Frank-Wolfe Algorithms

IOL Lab at ZIB and TUB

Deborah Hendrych (ZIB, TUB), Sebastian Pokutta (ZIB, TUB), Mathieu Besançon (Inria, UGA), David Martínez-Rubio (Carlos III)

Motivation

We consider constrained optimization problems of the form:

$$\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) \tag{P}$$

where f is a smooth convex function and $\mathcal{X}$ is a compact convex set.
Frank-Wolfe (FW) algorithms [2, 3] are popular for such problems, especially when projection onto $\mathcal{X}$ is expensive. A key component of FW variants is the choice of step size γ_t to update the iterate:

$$\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t - \gamma_t \boldsymbol{d}_t \tag{1}$$

where $\boldsymbol{d}_t$ is a descent direction and $\gamma_t \in [0, \gamma_{\max}]$.
The challenge: Finding an effective, efficient, and adaptive step-size strategy that:

- Makes good progress (effective)
- Has low computational cost (efficient)
- Adapts to local smoothness (adaptive)

Most existing strategies achieve only 2 out of 3 requirements.

Existing step-size strategies

Open-loop strategies: $\gamma_t = \frac{\ell}{\ell+t}$ with $\ell \geq 2$

- Very cheap to compute
- Can achieve higher convergence rates in some settings [1]
- Not adaptive to function geometry

Short-steps: $\gamma_t = \frac{\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{d}_t\rangle}{L\|\boldsymbol{d}_t\|^2}$

- Requires knowledge of Lipschitz constant L
- Conservative step sizes
- Not adaptive to local smoothness

Line search strategies: Solve $\min_{\gamma\in[0,\gamma_{\max}]} f(\boldsymbol{x}_t - \gamma \boldsymbol{d}_t)$

- Adaptive backtracking line search [4] (current gold standard)
- Golden ratio search
- High computational cost due to many function evaluations

Our approach: Secant Line Search (SLS)

We propose a new step-size strategy based on the **secant method** for solving the line search problem. The key insight is to reformulate the line search as a root-finding problem:
Line search problem:

$$\gamma^* = \operatorname*{arg\,min}_{\gamma\in[0,\gamma_{\max}]} f(\boldsymbol{x}_t - \gamma \boldsymbol{d}_t) \tag{2}$$

Optimality condition:

$$\varphi(\gamma) = \langle \nabla f(\boldsymbol{x}_t - \gamma \boldsymbol{d}_t), \boldsymbol{d}_t\rangle = 0 \tag{3}$$

Secant method recursion:

$$\gamma_{n+1} \leftarrow \gamma_n - \langle \nabla f(\boldsymbol{x}_t - \gamma_n \boldsymbol{d}_t), \boldsymbol{d}_t\rangle \cdot \frac{\gamma_n - \gamma_{n-1}}{\langle \nabla f(\boldsymbol{x}_t - \gamma_n \boldsymbol{d}_t), \boldsymbol{d}_t\rangle - \langle \nabla f(\boldsymbol{x}_t - \gamma_{n-1} \boldsymbol{d}_t), \boldsymbol{d}_t\rangle} \tag{4}$$

Algorithm

Algorithm 1 Secant Line Search (SLS)

Input: Function f, point $\boldsymbol{x}_t$, direction $\boldsymbol{d}_t$, initial γ_0, γ_1, tolerance ϵ
Output: Step size γ^*
Compute $\varphi_{-1} \leftarrow \langle \nabla f(\boldsymbol{x}_t - \gamma_{-1}\boldsymbol{d}_t), \boldsymbol{d}_t\rangle$
Compute $\varphi_0 \leftarrow \langle \nabla f(\boldsymbol{x}_t - \gamma_0\boldsymbol{d}_t), \boldsymbol{d}_t\rangle$
repeat
  $\gamma_1 \leftarrow \gamma_0 - \varphi_0 \cdot \frac{\gamma_0 - \gamma_{-1}}{\varphi_0 - \varphi_{-1}}$
  $\gamma_1 \leftarrow \max\{0, \min\{\gamma_1, \gamma_{\max}\}\}$ {clip}
  Update $\gamma_{-1} \leftarrow \gamma_0$, $\gamma_0 \leftarrow \gamma_1$
  Update $\varphi_{-1} \leftarrow \varphi_0$
  Compute $\varphi_0 \leftarrow \langle \nabla f(\boldsymbol{x}_t - \gamma_0\boldsymbol{d}_t), \boldsymbol{d}_t\rangle$
until $|\varphi_0| < \epsilon$
return $\gamma^* \leftarrow \gamma_0$

Key features:

- Only one gradient evaluation per iteration
- Warm-starting from previous optimal step size
- Automatic clipping to feasible range
- Robust fallback mechanisms

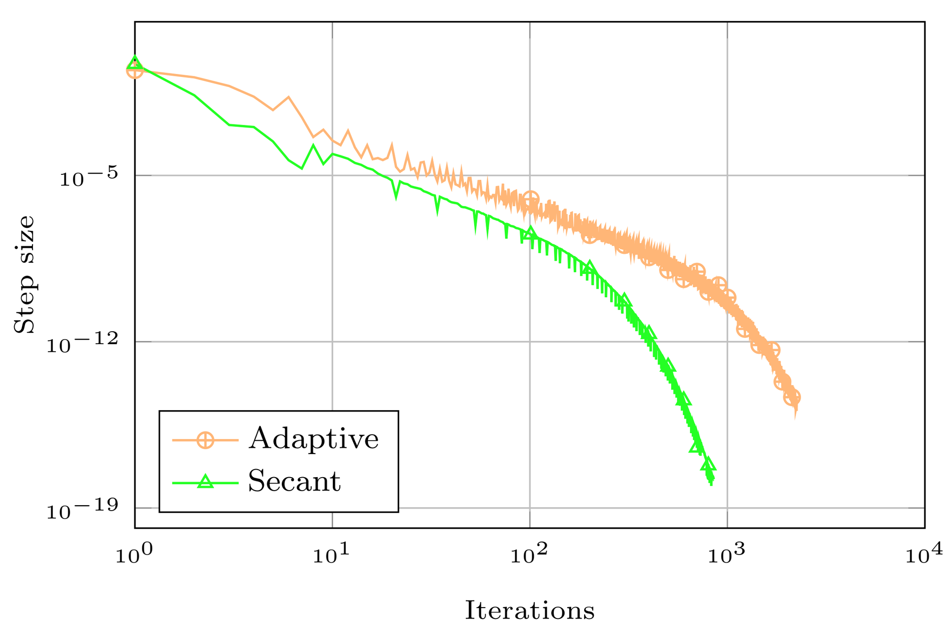

Figure 1. Step size evolution: SLS vs Adaptive line search on nuclear norm problem.

Theoretical guarantees

Global convergence: Under mild assumptions, SLS converges to the optimal step size. The key insight is that for FW algorithms, the optimal γ^* is confined to a bounded interval $[0, \gamma_{\max}]$, which allows us to establish convergence guarantees.
Superlinear convergence: For self-concordant functions, SLS achieves superlinear convergence with order ≈ 1.618 near the root.
Special case - Quadratics: For convex quadratics, SLS converges in exactly one iteration, making it as cheap as short-steps while being adaptive.
Implementation safeguards: When SLS fails, we can fall back to backtracking line search for robustness.

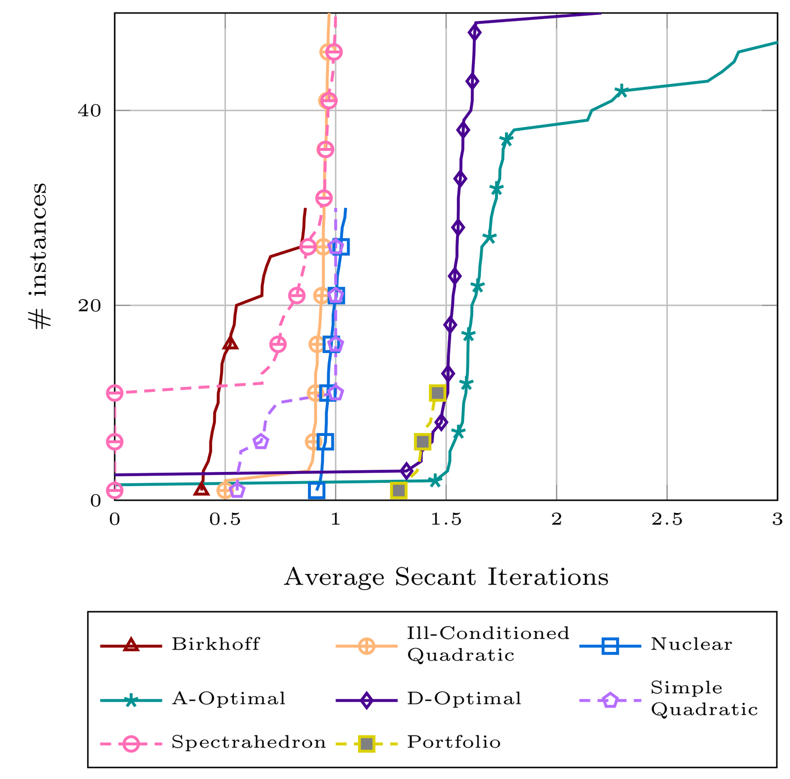

Figure 2. Average secant iterations per line search by problem class. Most quadratics converge in 1 iteration.

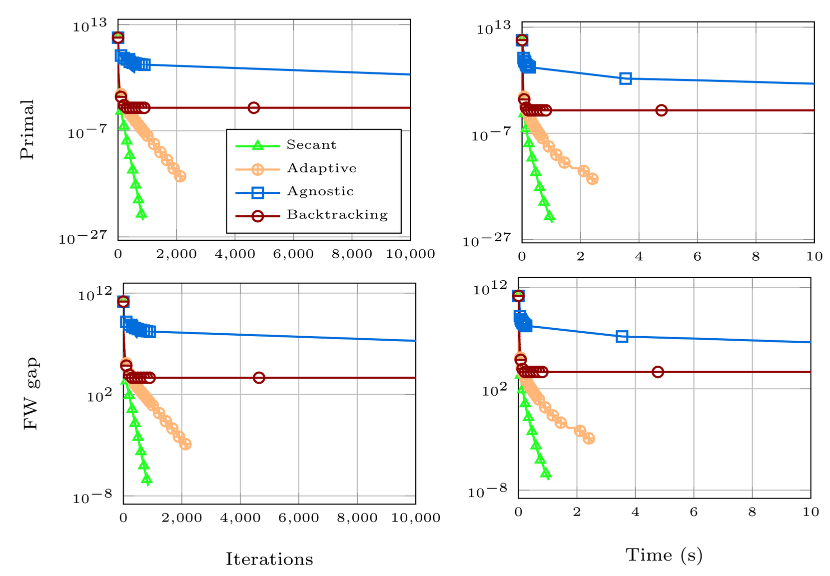

Figure 3. Convergence on nuclear norm problem. SLS significantly outperforms other methods.

Computational experiments

We evaluated SLS against state-of-the-art step-size strategies on 8 problem classes:

- **Quadratics:** Birkhoff polytope, simplex, spectraplex, nuclear norm ball
- **Self-concordant:** Portfolio optimization, optimal design of experiments
- **Ill-conditioned:** High condition number (quadratic) problems

Comparison methods:

- Adaptive line search (Pedregosa et al.)
- Backtracking line search
- Agnostic step size ($\frac{2}{t+2}$)
- Golden ratio search

Results summary

Performance highlights:

- **Low iteration count:** Average 1.5 secant iterations per line search
- **Superior convergence:** Often 2-10x faster than adaptive line search
- **Competitive timing:** Despite gradient evaluations, often fastest overall
- **Robust performance:** Best or close to best on most problem classes

Key advantages:

- Achieves all three requirements: effective, efficient, and adaptive
- Exploits local smoothness without requiring Lipschitz constants
- Works with any Frank-Wolfe variant
- Simple to implement and robust in practice

Outlook

Theoretical extensions:

- Broader convergence guarantees for non-convex objectives
- Analysis for other optimization algorithms beyond Frank-Wolfe
- Adaptive tolerance scheduling for line search accuracy

Practical improvements:

- Integration with more FW variants (away-steps, pairwise, etc.)
- Automatic parameter tuning for different problem classes
- GPU/parallel implementations for large-scale problems

Broader impact:

- Enables faster constrained optimization in machine learning
- Reduces computational costs in operations research
- Provides framework for adaptive step-size strategies

References

[1] F. Bach.
On the effectiveness of Richardson extrapolation in data science.
SIAM Journal on Mathematics of Data Science, 3(4):1251–1277, 2021.

[2] M. Frank and P. Wolfe.
An algorithm for quadratic programming.
Naval research logistics quarterly, 3(1-2):95–110, 1956.

[3] E. S. Levitin and B. T. Polyak.
Constrained minimization methods.
USSR Computational Mathematics and Mathematical Physics, 6(5):1–50, 1966.

[4] F. Pedregosa, G. Negiar, A. Askari, and M. Jaggi.
Linearly convergent Frank-Wolfe with backtracking line-search.
In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*.
PMLR, 2020.
arXiv:1806.05123v4.

Funding

This work was partially supported by:

- **Research Campus Modal** funded by the German Federal Ministry of Education and Research (fund numbers 05M14ZAM, 05M20ZBM)
- **Deutsche Forschungsgemeinschaft (DFG)** through the DFG Cluster of Excellence MATH+
- **IDEA-CM project** (TEC-2024/COM-89) - David Martínez-Rubio
- **MIAI at Université Grenoble Alpes** (ANR-19-P3IA-0003) - Mathieu Besançon