Recent improvements on the integer Frank-Wolfe framework

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

IOL Lab at ZIB and TUB

Motivation

We consider constrained mixed-integer nonlinear programs (MINLPs) where nonlinearities are primarily in the objective function.

Considered problem form:

$$\begin{aligned} \min_{\mathbf{x}} \; & f(\mathbf{x}) \\ \text{s.t. } & \mathbf{x} \in \mathcal{X} \\ & x_j \in \mathbb{Z} \quad \forall j \in J. \end{aligned} \tag{P}$$

where f is convex and Lipschitz-smooth, and $\mathcal{X}$ is a compact convex set.

Oracle setting

We only assume access to:

- A **Boundable Linear Minimization Oracle** (B-LMO) for $\mathcal{X}$:
$$\mathbf{v} \to \underset{\mathbf{y} \in \mathcal{X} \cap B \cap \mathbb{Z}}{\arg\min} \langle \mathbf{y}, \mathbf{d} \rangle ,$$
where B is a set of bound constraints intersected with $\mathcal{X}$.
- A zero-th and first-order oracle for f: $\mathbf{x} \to (f(\mathbf{x}), \nabla f(\mathbf{x}))$.

Our approach: branch-and-bound with Frank-Wolfe methods

Our solution approach to Problem equation P consists in a **Branch-and-Bound** process over the **convex hull** of the feasible region with inexact node processing. A key feature is that at each node, Frank-Wolfe (FW) solves the nonlinear subproblem over the convex hull of integer-feasible solutions or *integer hull* and not over the continuous relaxation. This is allowed by solving either a MIP or a combinatorial problem as the LMO within the FW solving process, thus resulting in vertices of the integer hull, as summarized on Figure 1. Our approach is explained in detail in [4] and is implemented in the Julia package Boscia.jl.

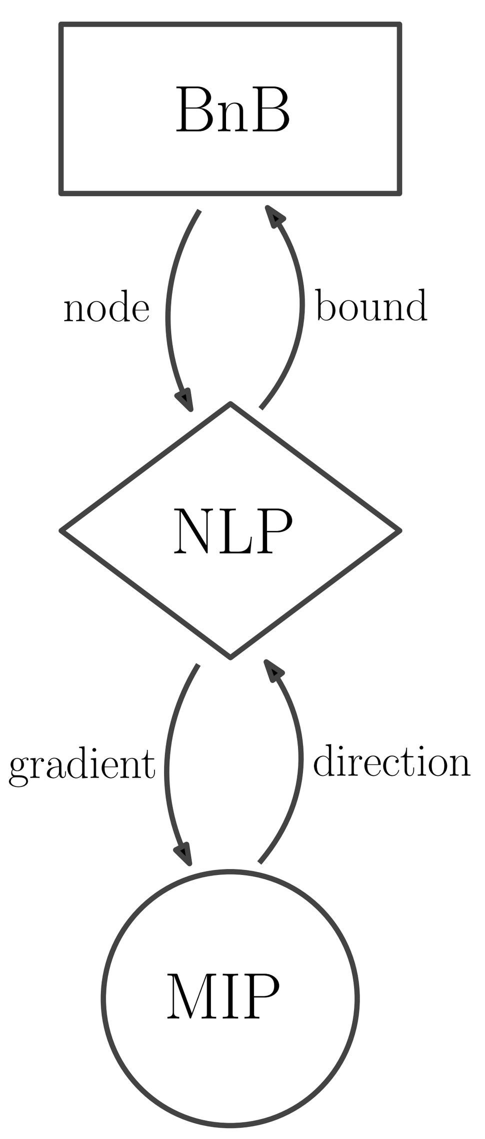

Figure 1. Summary of our approach

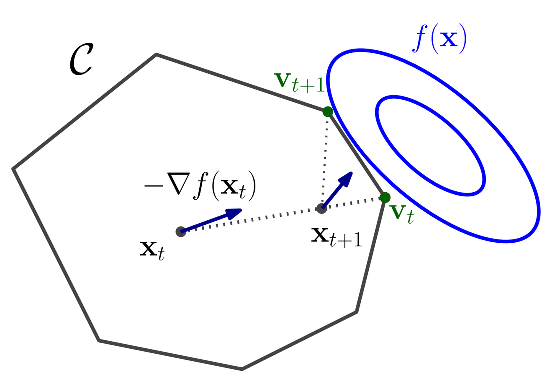

Figure 2. The Frank-Wolfe algorithm on polytopes

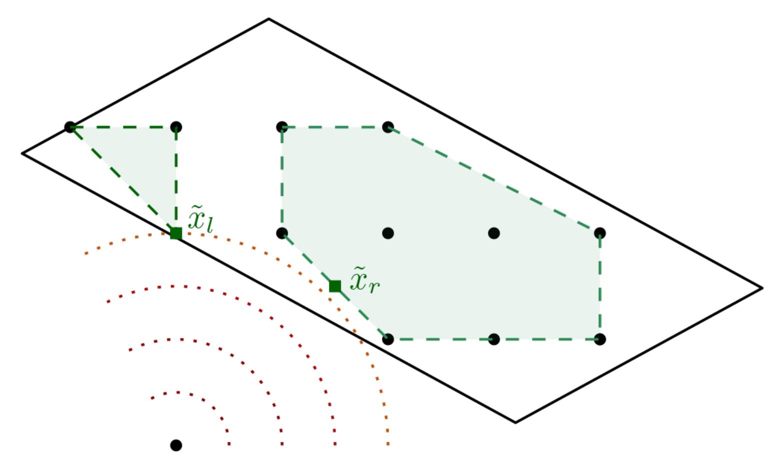

Figure 3. Our approach: branching over the convex hull

First-order methods for convex relaxations

We exploit first-order methods based on the Frank-Wolfe algorithm [1] illustrated on Figure 2, in particular for the following aspects:

- **Enhancing warm starts** via active-set decomposition of the last iterate.
- **Early termination** using the fact that Frank-Wolfe algorithms offer a safe dual bound at each iteration.
- **Dynamic determination of the best relaxation at each node**. Error adaptivity enables a dynamic decision on solution accuracy.
- **Lazification** utilizing lazification techniques for Frank-Wolfe algorithms [2] and additional lazification over the B&B tree.
- **Combinatorial solvers** to exploit combinatorial structure in the constraints; see `https://github.com/ZIB-IOL/CombinatorialLinearOracles.jl`.

The Secant line search (SLS)

In [3], we propose a new step-size strategy based on the secant method for solving the line search problem for Frank-Wolfe algorithms. The key insight is to reformulate line search as a root-finding problem:

Line search problem:

$$\gamma^* = \underset{\gamma \in [0, \gamma_{\max}]}{\arg\min} f(\mathbf{x}_t - \gamma \mathbf{d}_t) \tag{1}$$

Optimality condition:

$$\varphi(\gamma) = \langle \nabla f(\mathbf{x}_t - \gamma \mathbf{d}_t), \mathbf{d}_t \rangle = 0 \tag{2}$$

Secant method recursion:

$$\gamma_{n+1} \leftarrow \gamma_n - \langle \nabla f(\mathbf{x}_t - \gamma_n \mathbf{d}_t), \mathbf{d}_t \rangle \cdot \frac{\gamma_n - \gamma_{n-1}}{\langle \nabla f(\mathbf{x}_t - \gamma_n \mathbf{d}_t), \mathbf{d}_t \rangle - \langle \nabla f(\mathbf{x}_t - \gamma_{n-1} \mathbf{d}_t), \mathbf{d}_t \rangle} \tag{3}$$

Theoretical guarantees of SLS

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 $[0, \gamma_{\max}]$.
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 adaptive.
Implementation safeguards: When SLS fails, we can fall back to backtracking line search for robustness.

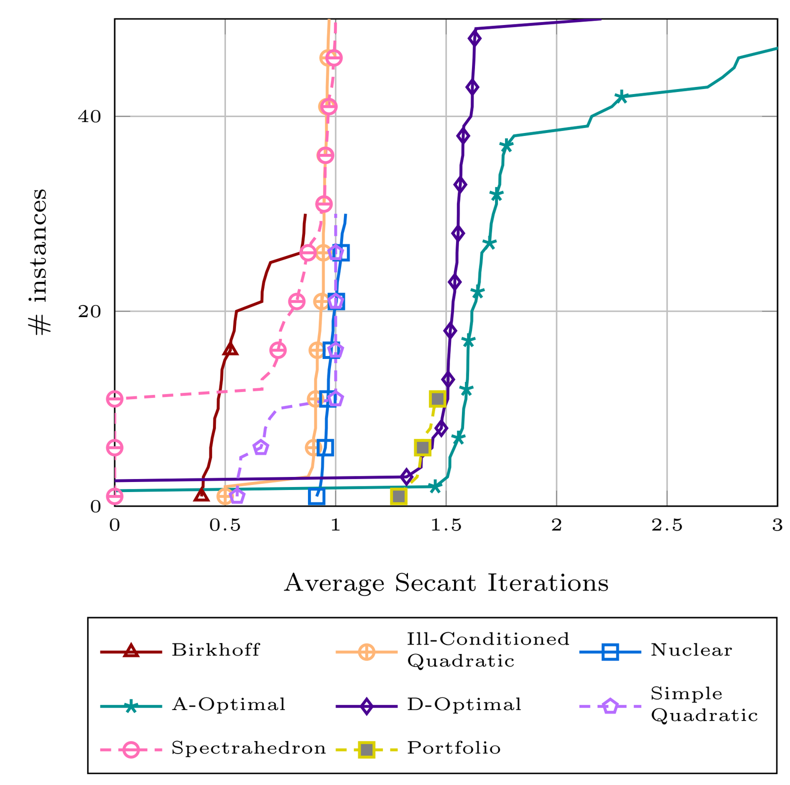

Figure 4. Average secant iterations per line search by problem class.

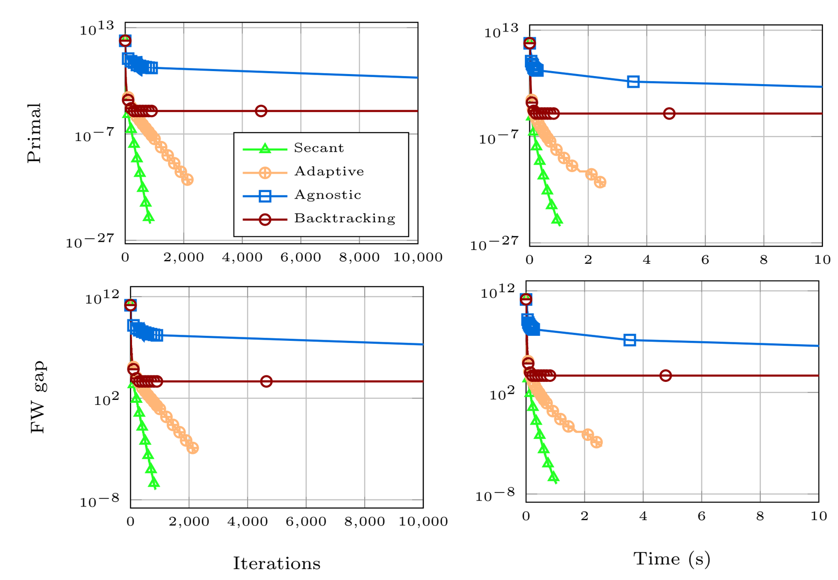

Figure 5. Convergence on nuclear norm problem.

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 $\left(\frac{2}{t+2}\right)$
- Golden ratio search

Heuristic for quadratic mixed-integer optimization

We won first place in the Land-Doig competition centered around primal heuristics for quadratic mixed-integer optimization. Our approach [5] is based on the Boscia framework with customizations for quadratic problems.

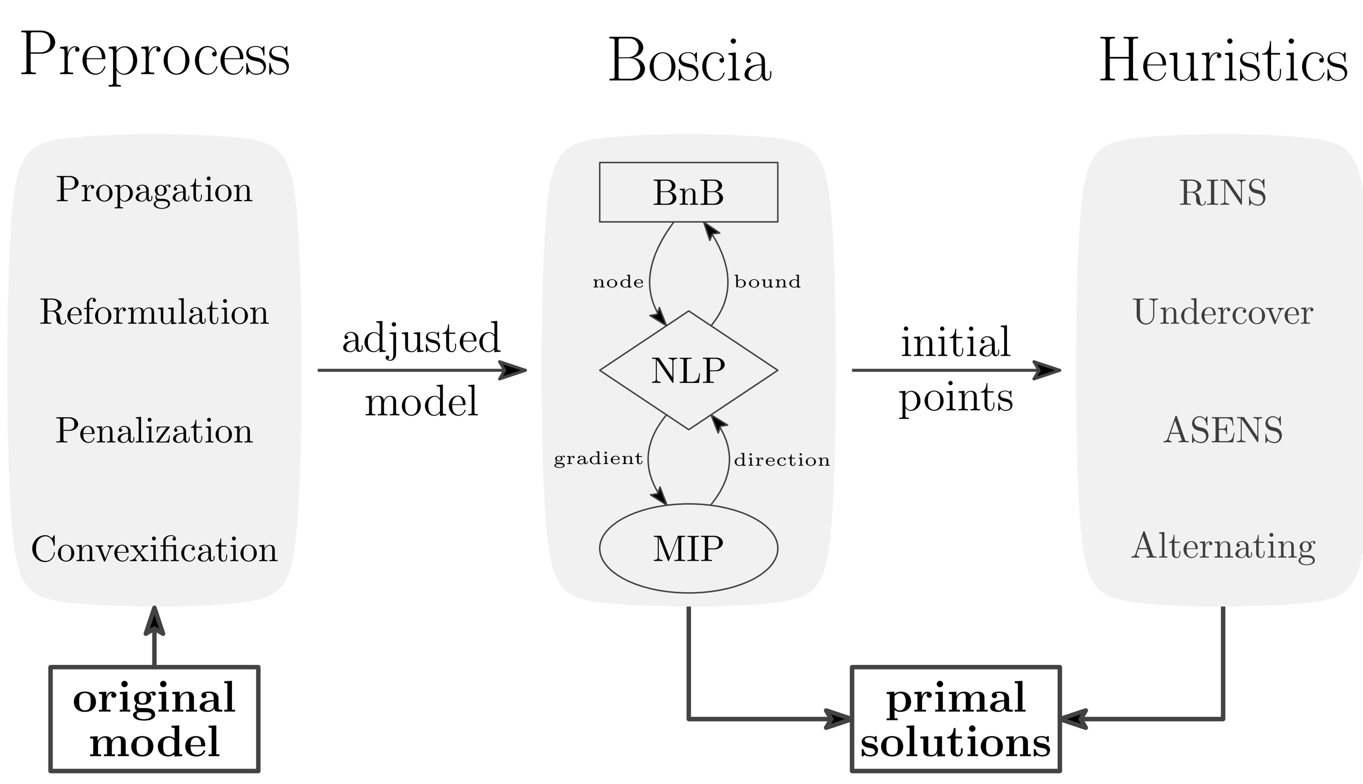

Figure 6. Overview of our approach. The model first undergoes a pre-processing whose main goal is to transfer nonlinearities into the objective function. Our workhorse is indeed the Frank-Wolfe-based solver Boscia which handles nonlinear problems (NLP) by suitably combining calls to a mixed-integer programming (MIP) solver in a branch-and-bound (BnB) framework. The points collected at each node in this process are then fed to various heuristics to reach better solutions.

Table 1. Improved solutions for QPLIB instances.

Instance	Obj. Sense	Previous Best	New Solution	Gap (%)
QPLIB_2169	max	29.0	30.0	3.45
QPLIB_2174	max	150.0	152.0	1.33
QPLIB_2205	max	88.0	90.0	2.27
QPLIB_3347	min	3,819,920	3,818,879	0.03
QPLIB_3584	min	-25,254	-25,386	0.52
QPLIB_3709	min	5,726,530	5,710,645	0.28
QPLIB_3860	min	-19,685	-20,161	2.42
QPLIB_10022	min	6,267,782	1,374,066	78.07

Table 2. Heuristic performance on the QPLIB instances.

Category	Found	Gap (%)	Primal Integral	Time to first
MIQP	163/166	4.06	5.23	1.76
MIQCQP	98/153	25.30	50.56	7.31
Total	261/319	11.57	12.77	3.18

References

[1] M. Besançon, A. Carderera, and S. Pokutta.
FrankWolfe.jl: A high-performance and flexible toolbox for Frank–Wolfe algorithms and conditional gradients.
INFORMS Journal on Computing, 2022.

[2] G. Braun, S. Pokutta, and D. Zink.
Lazifying conditional gradient algorithms.
In *International conference on machine learning*, pages 566–575. PMLR, 2017.

[3] D. Hendrych, M. Besançon, D. Martínez-Rubio, and S. Pokutta.
Secant line search for Frank-Wolfe algorithms.
PMLR, 2025.

[4] D. Hendrych, H. Troppens, M. Besançon, and S. Pokutta.
Convex mixed-integer optimization with Frank-Wolfe methods.
Mathematical Programming Computation, 2025.

[5] G. Mexi, D. Hendrych, S. Designolle, M. Besançon, and S. Pokutta.
A Frank-Wolfe-based primal heuristic for quadratic mixed-integer optimization.
2025.
`arXiv:2508.01299`.

ZUSE INSTITUTE BERLIN
UGA Université Grenoble Alpes
Inria