Berlin Mathematics Research Center MATH+

Exploiting Combinatorial Algorithms within Convex Mixed-Integer Optimization

Deborah Hendrych (ZIB), Mathieu Besançon (Inria, UGA, ZIB), Sebastian Pokutta (ZIB,TUB)

Motivation

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

Considered problem form:

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

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

We only assume access to the following oracles:

- A **Boundable Linear Minimization Oracle** (B-LMO) for $\mathcal{X}$:

$$d \to \operatorname*{argmin}_{v \in \mathcal{X} \cap B} \langle v, d \rangle,$$

  where B is a set of bound constraints intersected with $\mathcal{X}$,
- A zero-th and first-order oracle for f: $x \to (f(x), \nabla f(x))$.

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

Our solution approach to Problem (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.

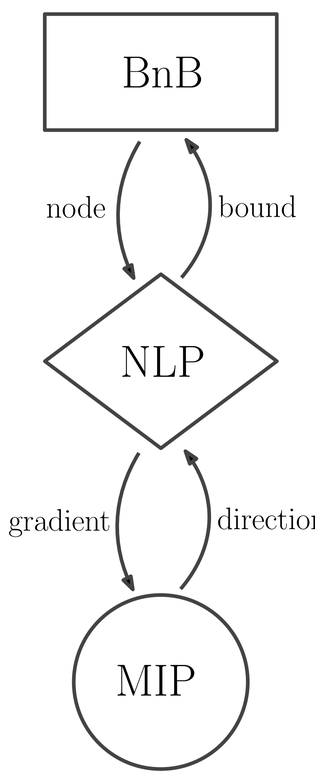

Figure 1. Summary of our approach

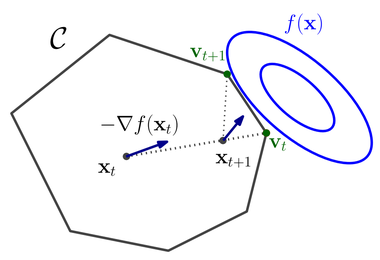

Figure 2. The Frank-Wolfe algorithm on polytopes

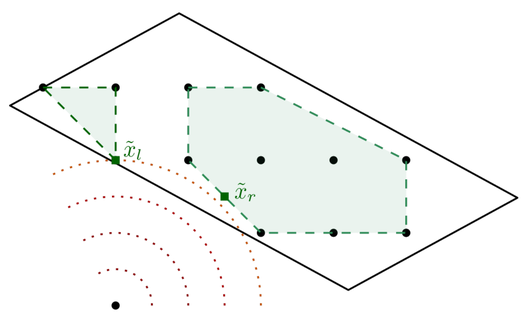

Figure 3. Our approach: branching over the convex hull

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. Thanks to the simple state first-order methods maintain, node warm starts will only require the primal and dual solutions or in the case of FW methods, the active set decomposition of the last iterate.

Early termination. Frank-Wolfe algorithms offer a safe dual bound at each iteration of the algorithm, letting us cut off the node if the bound reaches or overshoots the primal bound provided by the current incumbent solution.

Dynamic determination of the best relaxation at each node. Error adaptivity enables a dynamic decision that is aware of the context of the rest of the tree: to what accuracy should each subproblem be solved? The two extreme cases are:

- Continue solving the node only until it is not the lowest dual bound anymore.
- Solving the node to optimality.

Additionally, a node can also be terminated if specified number of nodes have a better lower bound and thus, are more promising to investigate.

Lazification of Frank-Wolfe and the bnb tree. Solving a MIP in each iteration of Frank-Wolfe can be prohibitive if the MIP is expensive. Thus, we exploit the lazification techniques available for Frank-Wolfe algorithms [2]. Additionally, we can also lazify over the bnb tree by collecting vertices discarded by FW (away-steps), resulting in every vertex being computed at most once.

Combinatorial solvers. On the other hand, many problems admit a combinatorial solution for the BLMO, e.g. Birkhoff polytope, simplex etc. Some of them can be found at `https://github.com/ZIB-IOL/CombinatorialLinearOracles.jl`. These can drastically speed up the Frank-Wolfe solution process and thereby the individual node evaluation since they are computationally less expensive than the corresponding MIP formulation. The lazification techniques from above can of course also be applied here.

Optimizing a quadratic over the Birkhoff polytope

To showcase the improved performance with combinatorial solvers within our approach, we look at a simple example: The aim is to minimize the distance to given double stochastic matrix $\hat{X}$.

$$\begin{aligned} \min_X \ & \frac{1}{2}\|X - \hat{X}\|_F^2 \\ \text{s.t. } & X \in P_n \end{aligned}$$

Solving the linear problem over the Birkhoff polytope comes down to the Assignment Problem. This can be efficiently solved with the Hungarian algorithm.

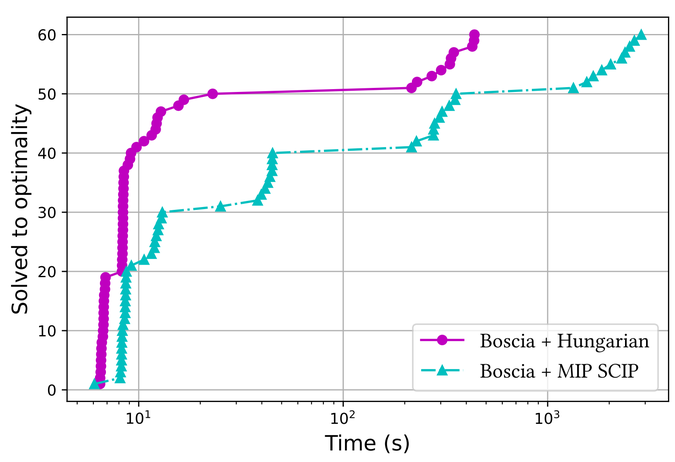

Figure 4. Comparing the solving time of Boscia with the MIP formulation of the Birkhoff polytope to the Hungarian algorithm as BLMO.

Evidently, using the **combinatorial solver as LMO** (pink) leads to a **faster solution process** than using the MIP formulation (cyan).

Application to the optimal experiment design problem

The *Optimal Experiment Design Problem (OEDP)* amounts to finding a subset of size N of a large set of experiments maximising the information gain. We assume we have a known experiment matrix $A \in \mathbb{R}^{m \times n}$. Then the information is encoded in the so-called information matrix, a linear map $X(\mathbf{x}) = A^\intercal \operatorname{diag}(\mathbf{x}) A$. An information measure is needed to compress the matrix X into a single number. There are different information measures, two popular measures are the A-Optimality criterion and D-Optimality criterion. The feasible region is just a scaled probability simplex intersected with a hypercube. The resulting BLMO can be computed by hand.

The A-optimal design problem

$$\begin{aligned} \max_{\mathbf{x}} \ & -\operatorname{Tr}(X(\mathbf{x})^{-1}) \\ \text{s.t. } & \sum_{i=1}^m x_i = N \\ & l_i \le x_i \le u_i \quad \forall i \in [m] \\ & \mathbf{x} \in \mathbb{Z}_+^m. \end{aligned}$$

The D-optimal design problem

$$\begin{aligned} \max_{\mathbf{x}} \ & \log\det(X(\mathbf{x})) \\ \text{s.t. } & \sum_{i=1}^m x_i = N \\ & l_i \le x_i \le u_i \quad \forall i \in [m] \\ & \mathbf{x} \in \mathbb{Z}_+^m. \end{aligned}$$

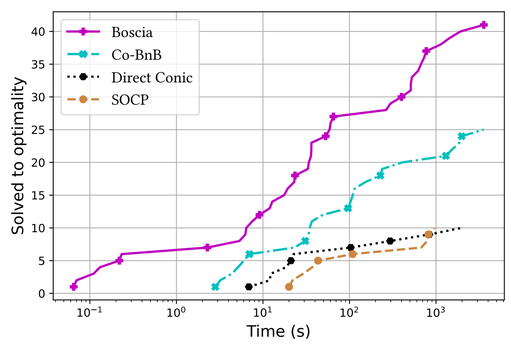

Figure 5. Comparing the number of optimally solved instances between the different solvers over time for the A-Optimal Problem

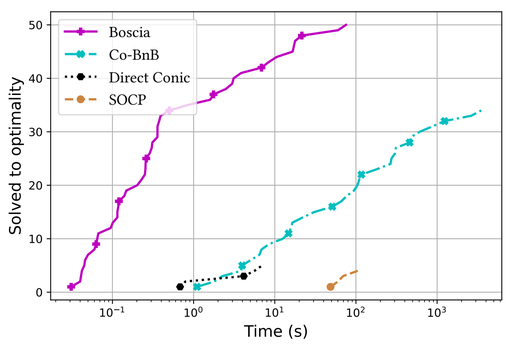

Figure 6. Comparing the number of optimally solved instances between the different solvers over time for the D-Optimal Problem

We compare the number of optimally solved instances of our framework (pink) to other approaches, another Branch-and-Bound approach (cyan) and two Conic Outer Approximation methods (yellow, black) [3].

Outlook

- Design a **precision tolerance schedule** for the precision at which we solve the subproblems with theoretical underpinning and guarantees.
- Define **valid inequalities** specific to our problem structure.
- Adapt the algorithm to problems with objective functions **changing across nodes**, e.g., Lagrangian approaches, smoothing-based methods.

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 *Proceedings of International Conference on Machine Learning*, volume 70, page 566–575, 2017.
URL: `https://proceedings.mlr.press/v70/braun17a`, `arXiv:1610.05120`.

[3] D. Hendrych, M. Besançon, and S. Pokutta.
Solving the optimal experiment design problem with mixed-integer convex methods.
In *Proceedings of Symposium on Experimental Algorithms*, 2024.
`arXiv:2312.11200`, `doi:10.4230/LIPIcs.SEA.2024.16`.

[4] D. Hendrych, H. Troppens, M. Besançon, and S. Pokutta.
Convex mixed-integer optimization with Frank-Wolfe methods.
arXiv preprint arXiv:2208.11010, 2022.