Berlin Mathematics Research Center MATH+

Berlin Mathematical School BMS

IOL Lab at ZIB and TUB

Exploiting Combinatorial Algorithms within Convex Mixed-Integer Optimization

Deborah Hendrych (ZIB,TUB), 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 $\mathcal{X}$:
$$d \rightarrow \underset{v \in \mathcal{X} \cap B \cap \mathbb{Z}}{\arg\min}\, \langle v, d\rangle,$$
where B is a set of bound constraints intersected with $\mathcal{X}$,
- A **zero-th and first-order oracle** for f: $x \rightarrow (f(x), \nabla f(x))$.

Existing methods

Nonlinear Branch-and-Bound:

- solves nonlinear continuous relaxations at each node,
- cannot directly handle oracle-based $\mathcal{X}$
- potentially many subproblems $\rightarrow$ full branch-and-bound tree.

Polyhedral outer-approximation:

- Reformulates the problem:
$$\begin{aligned}\min_{x,z}\ & z \\ \text{s.t. } & f(x) \leq z \\ & x \in \mathcal{X} \\ & x_j \in \mathbb{Z} \quad \forall j \in J\end{aligned}$$
- uses linear cuts to approximate $f(x) \leq z$
- loses the original problem structure
- potentially unstable through the addition of many potentially dense cuts.

Our approach and contributions

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, [3]. A key feature is that at each node, Frank-Wolfe (FW) [1] 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 a MIP as the LMO within the FW solving process, thus resulting in vertices of the integer hull, as summarized on Figure 1.

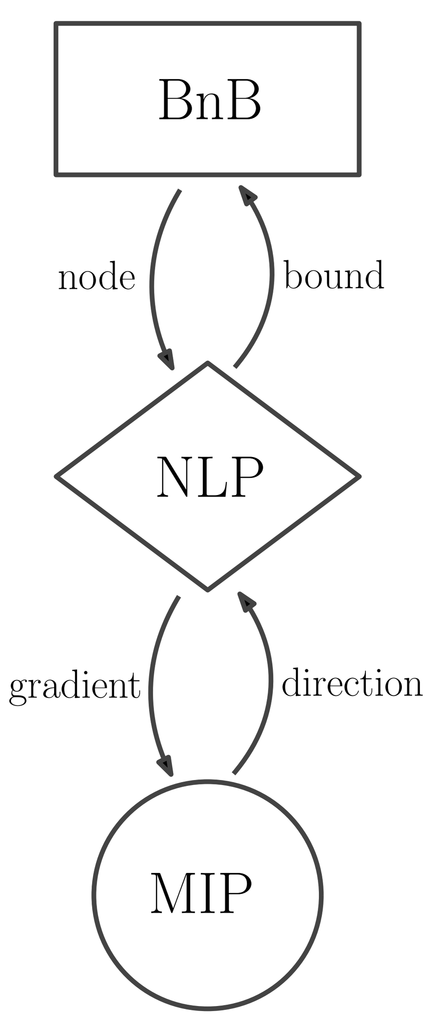

Figure 1. Summary of our approach

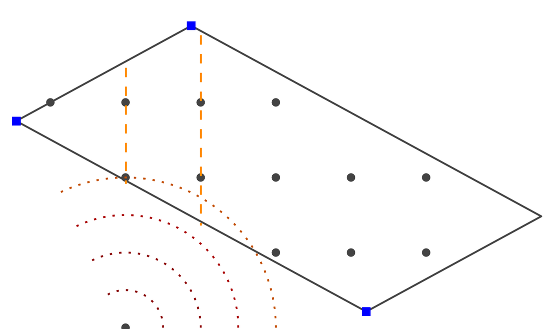
Figure 2. Branching over the continuous relaxation

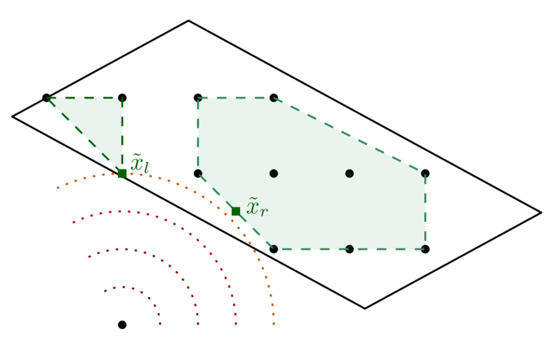
Figure 3. Our approach: branching over the convex hull

First-order methods for convex relaxations

We exploit first-order methods based on the Frank-Wolfe algorithm illustrated on Figure 4, 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.

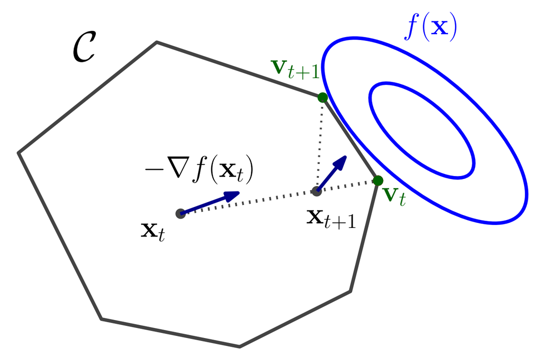

Figure 4. Frank-Wolfe algorithms on polytopes

Application to optimal design of experiments

Given a set of m experiments with n parameters decoded in the matrix $A \in \mathrm{R}^{m\times n}$, the **Optimal Design of Experiment Problem (OEDP)** aims to find the subset of experiments of size $N \geq n$ providing the most information about the system.
To measure information, we use the so-called information matrix

$$X(x) = A^{\intercal}\mathrm{diag}(x)A$$

where the variable x_i denotes how many times the experiment i is to be run. We also require an information function $\phi : \mathbb{S}_{++} \rightarrow \mathbb{R}$ which has to be concave, non-constant, non-negative, semi-upper continuous, and should respect the Loewner ordering. Different choices of ϕ lead to different criteria. The most common criteria are the D-criterion, $\log\det(X(x))$, and the A-criterion, $-\operatorname{Tr}(X(x)^{-1})$. The **A-Optimal Experiment Design Problem** and **D-Optimal Experiment Design Problem** can be formulated as:

$$\begin{aligned}\min_{\boldsymbol{x}}\ & \operatorname{Tr}\left((X(\boldsymbol{x}))^{-1}\right) \\ \text{s.t. } & \sum_{i=1}^{m} x_i = N \\ & \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u} \\ & \boldsymbol{x} \in \mathbb{Z}^m\end{aligned} \tag{A-Opt}$$

$$\begin{aligned}\min_{\boldsymbol{x}}\ & -\log(\det(X(\boldsymbol{x}))) \\ \text{s.t. } & \sum_{i=1}^{m} x_i = N \\ & \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u} \\ & \boldsymbol{x} \in \mathbb{Z}^m\end{aligned} \tag{D-Opt}$$

We compared the performance of our algorithm, implemented in the `Julia` package `Boscia.jl`, on the (A-Opt) and (D-Opt) Problems to two Outer Approximation (OA) Approaches based on Conic Optimization and another Branch-and-Bound method specifically designed for OEDP, called *Co-BnB*. The first OA approach utilizes a natural formulation of cones via the conic solver `Hypatia.jl`, hence labelled *Direct Conic*. The second method formulates the problem via second order cones, labbelled *SOCP*. Both OA schemes were implemented using `Pajarito.jl`, a conic mixed-integer solver with `Hypatia.jl` as conic solver and `HiGHS.jl` as MIP solver.

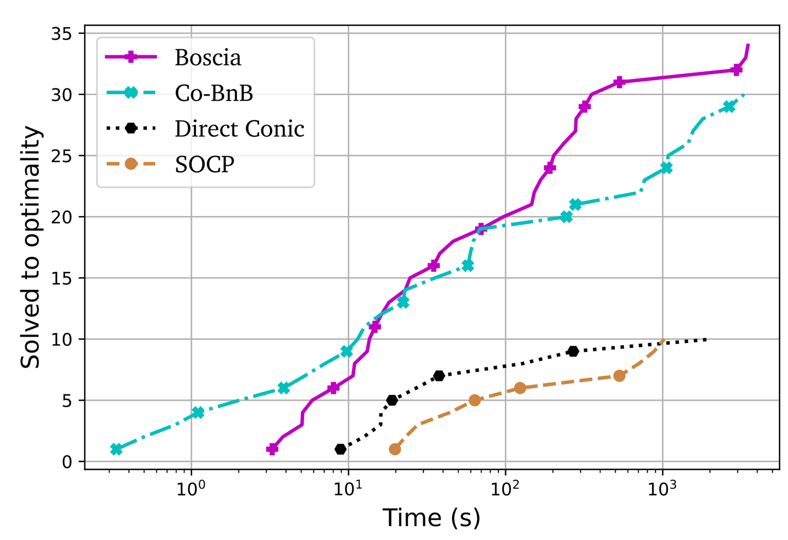

Figure 5. Termination over time for OEDP under the A-criterion.

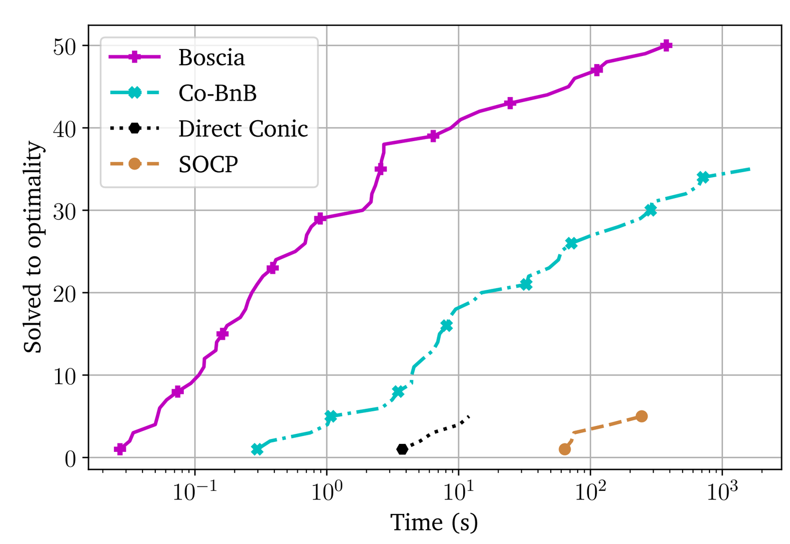

Figure 6. Termination over time for OEDP under the D-criterion.

For medium to large instances, $m \in [50, 120]$, our approach was superior to the other methods. It terminated more often in the given time limit of an hour and in the not-terminated cases, it was closer to the actual solution as could be seen at the dual gap between the current incumbent and lower bound of the tree. Extensive results can be found in [2].

Outlook

Theoretical:

- Determine **geometric criteria** for when to optimize over the convex hull or over the continuous relaxation.
- 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.
- Establish **linear convergence rates** for the continuous relaxations of design of experiment problems.

Practical:

- Adapt the algorithm to problems with objective functions **changing across nodes**, e.g., Lagrangian approaches, smoothing-based methods.
- Adapt the framework for **spatial branching** to handle nonconvex objectives.

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] D. Hendrych, M. Besançon, and S. Pokutta.
Solving the optimal experiment design problem with mixed-integer convex methods.
In *22nd International Symposium on Experimental Algorithms (SEA 2024)*, pages 16–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024.

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

ZIB ZUSE INSTITUTE BERLIN

IOL

TU berlin