A Branch-and-Cut Algorithm for Mixed-Integer Bilevel Linear Optimization Based on Improving Directions

F. Battista[1] and Ted K. Ralphs[1]

[1]COR@L Lab, Department of Industrial and System Engineering,
Lehigh University

24th of July, 2024

The Setting

- *First-level (aka Leader) variables*: $x \in X \subseteq \mathbb{Z}^{r_1} \times \mathbb{R}^{n_1 - r_1}$
- *Second-level (aka Follower) variables*: $y \in Y \subseteq \mathbb{Z}^{r_2} \times \mathbb{R}^{n_2 - r_2}$

Mixed-Integer Bilevel Linear Problem

$$\min_{x,y} \left\{ cx + d^1 y \mid x \in X, y \in \mathcal{P}_1(x), y \in \operatorname{argmin}\{d^2 z \mid z \in \mathcal{P}_2(x) \cap Y\} \right\}, \qquad \text{(MIBLP)}$$

where

$$\mathcal{P}_1(x) = \left\{ y \in \mathbb{R}^{n_2}_{+} \mid G^1 y \geq b^1 - A^1 x \right\},$$
$$\mathcal{P}_2(x) = \left\{ y \in \mathbb{R}^{n_2}_{+} \mid G^2 y \geq b^2 - A^2 x \right\}.$$

Assumptions

1. All input data are integer
2. All first-level variables are integer and appear in second-level constraints
3. The feasible regions are all bounded

The Notation

Let us denote

Rational reaction set

$$\mathcal{R}(x) = \left\{ y \in \mathcal{S}(x) \mid d^2 y \le d^2 \bar{y},\ \forall \bar{y} \in \mathcal{S}(x) \right\}$$

$\Rightarrow$ follower will respond **optimally** to leader's decision.

Bilevel Feasible Points

$$\mathcal{F} = \{(x, y) \in \mathcal{S} \mid y \in \mathcal{R}(x)\},$$

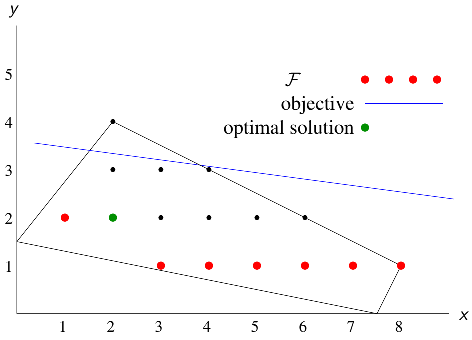
where

- $\mathcal{P} = \{(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \mid y \in \mathcal{P}_1(x) \cap \mathcal{P}_2(x)\}$,
- $\mathcal{S} = \{(x, y) \in X \times Y \mid (x, y) \in \mathcal{P}\}$,
- $\mathcal{S}(x) = \{y \in Y \mid (x, y) \in \mathcal{S}\} \Rightarrow$ follower's feasible points for a given leader's decision.

Optimistic setup

Whenever $|\mathcal{R}(x)| > 1$, follower selects the response most favorable for the leader.

Let us consider the following example from Moore and Bard [1990]:

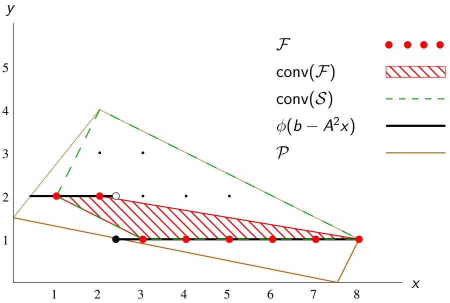


$$
\begin{aligned}
\min_{x \in \mathbb{Z}_+} \quad & -x - 10y \\
\text{s.t.} \quad & y \in \operatorname{argmin} \{ y : \\
& \quad -5x + 4y \leq 6 \\
& \quad x + 2y \leq 10 \\
& \quad 2x - y \leq 15 \\
& \quad 2x + 10y \geq 15 \\
& \quad y \in \mathbb{Z}_+ \}
\end{aligned}
$$

Theorem (Tahernejad and Ralphs [2020])

Under the stated assumptions, we have that

$$\min_{(x,y)\in\mathcal{F}} cx + d^1 y = \min_{(x,y)\in\text{conv}(\mathcal{F})} cx + d^1 y. \quad (1)$$



$$\begin{aligned} \min \quad & -x - 10y \\ \text{s.t.} \quad & (x, y) \in \text{conv}(\mathcal{F}) \end{aligned}$$

- Reformulation (1) suggests a Branch-and-Cut algorithm, similar to that for MILPs [DeNegre and Ralphs, 2009]
- Dual bounds can be obtained by optimizing over a relaxed feasible region
- The goal is to approximate conv($\mathcal{F}$) with linear inequalities

Branch-and-Cut Algorithm

- The basic framework presents many similarities to that used for MILPs, but with at least as many subtle differences.

Components

- **Bounding**
  - **Dual bound** ⇒ A "tractable" relaxation strengthened with valid inequalities
  - **Primal bound** ⇒ **Feasible solutions**
- Branching ⇒ Valid disjunctions
- **Cut generation** ⇒ Valid inequalities for conv($\mathcal{F}$)
- Search strategies
- Preprocessing methods
- Primal heuristics
- Control mechanisms

- In this talk we focus on the highlighted components.

Dual bounds

Standard relaxations

1. $\mathcal{S} \Rightarrow$ Relaxing the *optimality constraint of the second-level problem* (MILP relaxation).
2. $\mathcal{P} \Rightarrow$ Relaxing the *optimality constraint of the second-level problem* and *integrality constraints* (LP relaxation).

A different rationale may be to focus on relaxing the definition of $\mathcal{R}(x)$

$$\tilde{\mathcal{R}}(x) = \left\{ y \in \mathcal{S}(x) \mid d^2 y \leq d^2 \bar{y}, \;\; \forall \bar{y} \in \tilde{\mathcal{S}}(x) \right\},$$

for some $\tilde{\mathcal{S}}(x) \subseteq \mathcal{S}(x)$, for all $x \in X$.

For $Y = \{0,1\}^{n_2}$, Shi et al. [2023] proposed the **k-neighborhood set**

$$\mathcal{N}_k(y) = \left\{ \bar{y} \in \{0,1\}^{n_2} \mid \|\bar{y} - y\|_1 \leq k \right\}, \text{for } k \in \mathbb{Z}_+, k \leq n_2,$$

and the **k-optimal reaction set**

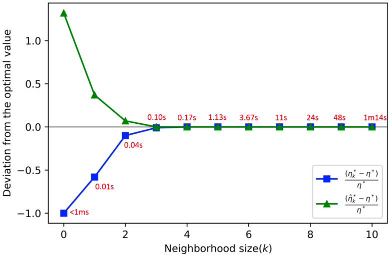
$$\mathcal{R}_k(x) = \left\{ y \in \mathcal{S}(x) \mid d^2 y \leq d^2 \bar{y}, \;\; \forall \bar{y} \in \mathcal{N}_k(y) \cap \mathcal{S}(x) \right\}.$$

Dual bounds: a locally optimal follower

k-optimal follower relaxation

$$
\begin{aligned}
\min \quad & cx + d^1 y \\
\text{s.t.} \quad & (x, y) \in \mathcal{S} \\
& y \in \mathcal{R}_k(x)
\end{aligned}
\tag{k-BP}
$$

Let us look at an example of Knapsack Interdiction Problem from Shi et al. [2023]



At a glance

- (k-BP) can be formulated as a MILP;
- Dual bounds are tighter than $\mathcal{S}$ and $\mathcal{P}$, even for small k;
- (k-BP) converges to (MIBLP) rather fast;
- (k-BP) bound comes at a reasonable computational cost.

Which relaxation?

- What relaxation is the "best" is ultimately an empirical question.
- It is tempting to think that a stronger relaxation (either $\mathcal{S}$ or (k-BP)) should be better than a "simple" LP.
- However, employing a B&C to solve MILP subproblems equals to delegate part of the **same branching process** that the "outer" B&C would undertake anyway.
- More importantly, re-optimization is **crucial** for cut generation.
- All in all, it only seems to make sense to use the "good old" LP relaxation for bounding.

Basic Idea: Identifying Infeasible Solutions

- When $(x, y) \in \mathcal{P} \setminus \mathcal{S}$, infeasibility is easy to verify.
- However for $(x, y) \in \mathcal{S} \setminus \mathcal{F}$, this might be a hard task.
- Usually, this is accomplished by solving the follower's problem (an MILP) to optimality.
- But maybe, looking at the "neighborhood" of y might give us information about the (sub-)optimality of y for the second-level problem.

Second-level Improving Directions

Let $(\hat{x}, \hat{y}) \in \mathcal{P}$. We say that $w \in \mathbb{Z}^{r_2} \times \mathbb{R}^{n_2 - r_2}$ is a **direction** (D). Moreover, we say that:

- $d^2 w < 0 \Rightarrow w$ is **improving** (I), and
- $\hat{y} + w \in \mathcal{P}_2(\hat{x}) \Rightarrow w$ is **feasible** (F).

The set of all IFDs w.r.t. $(\hat{x}, \hat{y})$ is

$$\mathcal{W}(\hat{x}, \hat{y}) = \left\{ w \in \mathbb{Z}^{r_2} \times \mathbb{R}^{n_2 - r_2} \mid d^2 w < 0,\ \hat{y} + w \in \mathcal{P}_2(\hat{x}) \right\}.$$

- This leads to an alternative method to check bilevel feasibility.

Lemma (Bilevel Feasibility Oracle)

For $(\hat{x}, \hat{y}) \in \mathcal{S}$, we have $(\hat{x}, \hat{y}) \in \mathcal{F} \iff \mathcal{W}(\hat{x}, \hat{y}) = \emptyset$.

An Oracle based on Improving Directions

- We can describe $\mathcal{W}(\hat{x}, \hat{y})$ as the points satisfying

$$\begin{aligned} d^2 w &\leq -1 \\ G^2 w &\geq b^2 - A^2 \hat{x} - G^2 \hat{y} \\ w &\geq -\hat{y} \\ w &\in \mathbb{Z}^{r_2} \times \mathbb{R}^{n_2 - r_2}. \end{aligned} \tag{IFD}$$

- (IFD) is formally equivalent to solving the follower's problem.
- However, we can plug (IFD) with a variety of obj. functions to obtain directions with favorable properties, e.g.,
  - $\min 1 \Rightarrow$ Checking Bilevel Feasibility
  - $\min d^2 w \Rightarrow$ Find the best follower's solution for the given $\hat{x}$
- Or, we can solve (IFD) with some kind of local search and optimize nonlinear obj. functions such as
  - $\min \|w\|_2 \Rightarrow$ Find the "shortest" IFD.

Cut Generation: Basic Principles

Valid Inequality

The triple $(\alpha^x, \alpha^y, \beta) \in \mathbb{R}^{n_1+n_2+1}$ is a **valid inequality** for $\mathcal{F}$ if

$$\mathcal{F} \subseteq \left\{(x, y) \in \mathbb{R}^{n_1+n_2} \mid \alpha^x x + \alpha^y y \geq \beta\right\}.$$

We refer to a valid inequality for $\mathcal{F}$ that is violated by a given solution of the current relaxation as a **cutting plane** (cut).

Bilevel Free Set

A **bilevel free set** (BFS) is a set $C \subseteq \mathbb{R}^{n_1+n_2}$ such that $\text{int}(C) \cap \mathcal{F} = \emptyset$.

General recipe for cuts

- Find a BFS $C \subseteq \mathbb{R}^{n_1+n_2}$;
- Then inequalities valid for $\text{conv}(\overline{\text{int}(C)})$ are also valid for $\mathcal{F}$.

Improving Direction Intersection Cuts (IDICs)

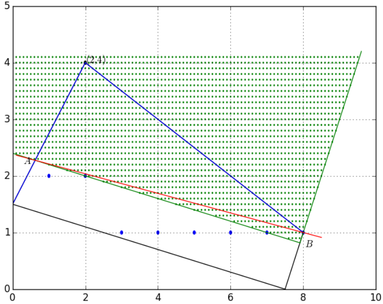
- Let $(\hat{x}, \hat{y})$ be an extreme point of $\text{conv}(\mathcal{S})$ (or $\mathcal{P}$) and let $w \in \mathcal{W}(\hat{x}, \hat{y})$ ($\Leftarrow$ IFD).

Bilevel Free Set [Fischetti et al., 2018]

$$C(w) = \left\{(x, y) \in \mathbb{R}^{n_1+n_2} \mid A^2x + G^2y \geq b^2 - G^2w - 1, y + w \geq -1\right\}.$$

$\Rightarrow$ the set of points (x, y) such that w is an IFD w.r.t. y.

- Let $\mathcal{V}(\hat{x}, \hat{y}) \supseteq \text{conv}(\mathcal{S})$ (or $\mathcal{P}$) be a radial cone with vertex $(\hat{x}, \hat{y})$.



- $\alpha^x x + \alpha^y y = \beta$ is the hyperplane passing through the intersection of $C(w)$ and $\mathcal{V}(\hat{x}, \hat{y})$.
- $(\alpha^x, \alpha^y, \beta)$ is valid for $\text{conv}(\mathcal{F})$.
- $\|w\|_2$ affects the "depth" of the cut
- separation is not guaranteed when $(\hat{x}, \hat{y}) \in \mathcal{P} \setminus \mathcal{S}$.

IDIC and k-optimality

- Once again, assume $Y = \{0,1\}^{n_2}$.
- Given $w \in \mathcal{W}(x,y)$ with $\|w\|_1 = k$, we refer to the cut generated from $C(w)$ as k-IDIC.

k-IDIC closure

For $k = 0, \ldots, n_2$, we define

$$\mathcal{S}^k = \Big\{(x,y) \in \mathcal{S} \mid \alpha^x x + \alpha^y y \geq \beta, \\ \forall (\alpha^x, \alpha^y, \beta) \in \mathbb{R}^{n_1+n_2+1} \text{ s.t. } (\alpha^x, \alpha^y, \beta) \text{ is a } \bar{k}\text{-IDIC valid for } \mathcal{F}, \forall \bar{k} \leq k\Big\}.$$

Theorem

For all $0 \leq k \leq n_2, \ (x,y) \in \mathcal{S}^k \iff y \in \mathcal{R}_k(x)$.

Theorem

For all $0 \leq k \leq n_2, \ (x, y) \in \mathcal{S}^k \iff y \in \mathcal{R}_k(x).$

We need the following

Lemma (Local k-optimality, Shi et al. [2023])

Assume $Y = \{0,1\}^{n_2}$, *let* $0 \leq k \leq n_2$ *integer and* $(x, y) \in \mathcal{S}$. *Then*
$\nexists \ w \in \mathcal{W}(x, y)$ *with* $\|w\|_1 \leq k \iff (x, y)$ *is feasible for* (k-BP) *(i.e* $y \in \mathcal{R}_k(x)$*).*

Sketch of Proof. Assume $(x, y) \in \mathcal{S}$, then

$$y \in \mathcal{R}_k(x) \iff \nexists \ w \in \mathcal{W}(x, y) \text{ with } \|w\|_1 \leq k \cong$$

$$\cong \nexists \text{ a } k\text{-IDIC violated by } (x, y) \text{ with } \|w\|_1 \leq k \iff (x, y) \in \mathcal{S}^k.$$

Takeaways

- IDs unify the **bilevel feasibility check** and the **generation** of strong inequalities.
- The existence of an IFD for a given solution of the current relaxation is sufficient condition for bilevel **infeasibility**.
- By generating IDICs, we are iteratively restoring local/global-optimality condition of the follower.

Computational Results: The Setup

MibS [Tahernejad et al., 2020]

Branch&Cut open-source solver for MIBLPs (available at www.coin-or.org)

- **Bounding**
  - **Dual bound** $\Rightarrow$ Compute optimal $(\hat{x}, \hat{y}) \in \mathcal{P}$ (LP relaxation)
  - **Primal bound** $\Rightarrow$ **Feasible solutions** by solving follower's problem (a MILP)
- **Cut generation**
  - **IDICs** $\Rightarrow$ find w by solving (IFD) as a MILP
  - MILP cuts $\Rightarrow$ when $(\hat{x}, \hat{y}) \in \mathcal{P} \setminus \mathcal{S}$

All other cuts are turned off. Defaults are used unless stated otherwise.

idB&C

Implemented modifying MibS:

- **Bounding**
  - **Dual bound** $\Rightarrow$ Compute optimal $(\hat{x}, \hat{y}) \in \mathcal{P}$ (LP relaxation)
  - **Primal bound** $\Rightarrow$ **Feasible solutions** when $(\hat{x}, \hat{y}) \in \mathcal{S}$ and $\mathcal{W}(\hat{x}, \hat{y}) = \emptyset$
- **Cut generation**
  - **IDICs** $\Rightarrow$ find w with **either** a Local Search to enumerate solutions of (IFD), **or** $\Rightarrow$ solve (IFD) as a MILP (necessary when $(\hat{x}, \hat{y}) \in \mathcal{S}$ and Local Search fails)
  - MILP cuts $\Rightarrow$ when $(\hat{x}, \hat{y}) \in \mathcal{P} \setminus \mathcal{S}$

Computational Results: The Setup

MibS [Tahernejad et al., 2020]

Branch&Cut open-source solver for MIBLPs (available at `www.coin-or.org`)

- **Bounding**
  - **Dual bound** ⇒ Compute optimal $(\hat{x}, \hat{y}) \in \mathcal{P}$ (LP relaxation)
  - **Primal bound** ⇒ **Feasible solutions** by solving follower's problem (a MILP)
- **Cut generation**
  - **IDICs** ⇒ find w by solving (IFD) as a MILP
  - MILP cuts ⇒ when $(x, y) \in \mathcal{P} \setminus \mathcal{S}$

All other cuts are turned off. Defaults are used unless stated otherwise.

idB&C

Implemented modifying MibS:

- **Bounding**
  - **Dual bound** ⇒ Compute optimal $(\hat{x}, \hat{y}) \in \mathcal{P}$ (LP relaxation)
  - **Primal bound** ⇒ **Feasible solutions** when $(\hat{x}, \hat{y}) \in \mathcal{S}$ and $\mathcal{W}(\hat{x}, \hat{y}) = \emptyset$
- **Cut generation**
  - **IDICs** ⇒ find w with **either** a Local Search to enumerate solutions of (IFD), **or**
    ⇒ solve (IFD) as a MILP (necessary when $(\hat{x}, \hat{y}) \in \mathcal{S}$ and Local Search fails)
  - MILP cuts ⇒ when $(x, y) \in \mathcal{P} \setminus \mathcal{S}$

Computational Results: The Setup

Configurations

- Mibs: version 1.2 ($\Leftarrow$ the "baseline")
- idB&C-IDIC-LS: Using Local Search always (when we have the choice)
- idB&C-IDIC-LS-0-10: Using Local Search when $0 \leq$ tree depth ≤ 10
- idB&C-IDIC-LS-10-inf: Using Local Search when tree depth ≥ 10

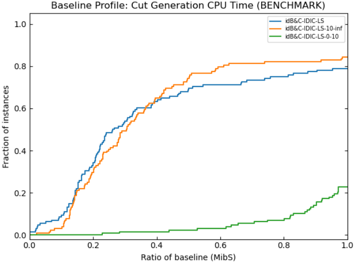
Dataset

The BENCHMARK (total of 179 instances) is made out of:

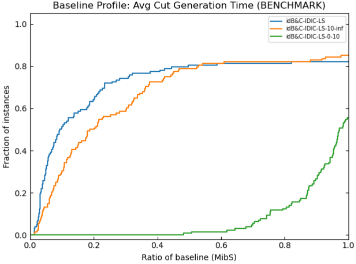
- Interdiction problems from Shi et al. [2023];
- Pure-integer from BOBILib (available at bobilib.org)

- Instances that required a Solution Time $\in \,]0, 1] \cup [3600, \infty[$ seconds for all configurations are excluded from the plots

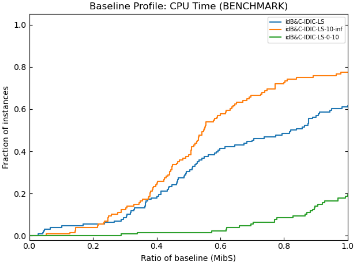
CG Time



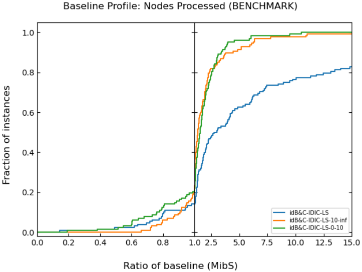
Average CG Time



Solution Time



Node Processed



Final Remarks

- IDs unify the **bilevel feasibility check** and the **generation** of strong inequalities.
- Our B&C based on an oracle for the existence of IFDs shows promising results.
- In particular, when the search for IFDs is combined with a Local Search it achieves both lower **Cut Generation** and **Solution Time**.

Future Works

- Implement more refined Control Mechanisms for the Local Search.
- Find good "off-the-shelf" defaults based on the problem structure.

Final Remarks

- IDs unify the **bilevel feasibility check** and the **generation** of strong inequalities.
- Our B&C based on an oracle for the existence of IFDs shows promising results.
- In particular, when the search for IFDs is combined with a Local Search it achieves both lower **Cut Generation** and **Solution Time**.

Future Works

- Implement more refined Control Mechanisms for the Local Search.
- Find good "off-the-shelf" defaults based on the problem structure.

Thanks for your attention!

Bibliography I

DeNegre and Ralphs. A branch-and-cut algorithm for integer bilevel linear programs. In Operations research and cyber-infrastructure, pages 65–78. Springer, 2009.

Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. On the use of intersection cuts for bilevel optimization. Mathematical Programming, 172(1-2): 77–103, 2018.

James T Moore and Jonathan F Bard. The mixed integer linear bilevel programming problem. Operations research, 38(5):911–921, 1990.

Xueyu Shi, Oleg A Prokopyev, and Ted K Ralphs. Mixed integer bilevel optimization with a k-optimal follower: a hierarchy of bounds. Mathematical Programming Computation, 15(1):1–51, 2023.

Tahernejad, Ralphs, and DeNegre. A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation. Mathematical Programming Computation, 12:529–568, 2020.

Sahar Tahernejad and Ted K Ralphs. Valid Inequalities for Mixed Integer Bilevel Linear Optimization Problems. Technical report, Lehigh University, 2020.