Latest Developments in the MibS solver for Mixed Integer Bilevel Optimization

Ted Ralphs[1]
Primary Collaborators: Sahar Tahernejad[1], Scott DeNegre [1]

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

EURO, Copenhagen, June, 2024

Attributions

Many Ph.D students and postdocs contributed to development of this work over time.

Current/Former Students/Postdocs

- Federico Battista
- Suresh Bolusani
- Scott DeNegre
- Samira Fallah
- Menal Gúzelsoy
- Anahita Hassanzadeh
- Ashutosh Mahajan
- Sahar Tahernajad
- Yu Xie

Thanks!

Outline

Setting

- *First-level variables*: $x \in X$ where $X = \mathbb{Z}_+^{r_1} \times \mathbb{R}_+^{n_1 - r_1}$
- *Second-level variables*: $y \in Y$ where $Y = \mathbb{Z}_+^{r_2} \times \mathbb{R}_+^{n_2 - r_2}$

MIBLP

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

where

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

Later, we'll need to refer to

$$\mathcal{P} = \{(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \mid y \in \mathcal{P}_1(x) \cap \mathcal{P}_2(x)\}$$

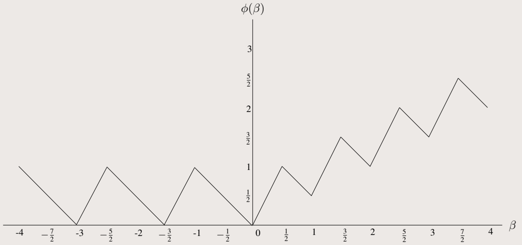
The Second-level Value Function

- The second-level *value function* is

MILP Value Function

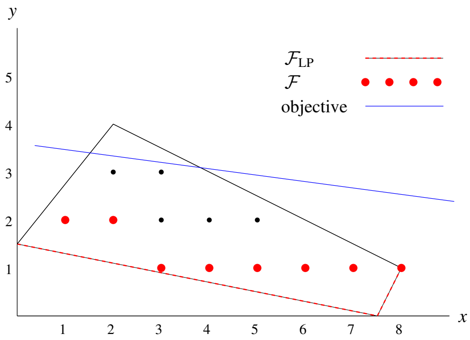
$$\phi(\beta) = \min \left\{ d^2 y \mid G^2 y \geq \beta, y \in Y \right\} \qquad \text{(VF)}$$

We let $\phi(\beta) = \infty$ if $\{y \in Y \mid G^2 y \geq \beta\} = \emptyset$.



The Standard Running Example

Example 1 Moore and Bard [1990]



$$
\begin{aligned}
\min_{x\in\mathbb{Z}_+} \quad & -x - 10y \\
\text{s.t.} \quad & y \in \operatorname{argmin}\big\{y : \\
& \quad -25x + 20y \le 30 \\
& \quad x + 2y \le 10 \\
& \quad 2x - y \le 15 \\
& \quad 2x + 10y \ge 15 \\
& \quad y \in \mathbb{Z}_+ \ \big\}
\end{aligned}
$$

□

Value Function Reformulation

- *First-level variables*: $x \in X$ where $X = \mathbb{Z}_+^{r_1} \times \mathbb{R}_+^{n_1 - r_1}$
- *Second-level variables*: $y \in Y$ where $Y = \mathbb{Z}_+^{r_2} \times \mathbb{R}_+^{n_2 - r_2}$

MIBLP

$$\min_{x,y} \left\{ cx + d^1 y \;\middle|\; x \in X, y \in \mathcal{P}_1(x) \cap \mathcal{P}_2(x) \cap Y, d^2 y \leq \phi(b^2 - A^2 x) \right\} \tag{MIBLP-VF}$$

Bilevel Feasible Region

$$\mathcal{F} = \left\{ (x, y) \in \mathcal{S} \;\middle|\; d^2 y \leq \phi(b^2 - A^2 x) \right\},$$

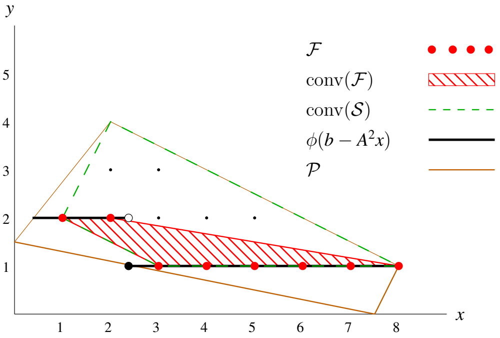
where

$$\mathcal{S} = \{ (x, y) \in X \times Y \mid y \in \mathcal{P}_1(x) \cap \mathcal{P}_2(x) \}$$

- This reformulation seems to suggest a Benders-type algorithm in which we approximate the second-level value function.
- Convexification helps avoid approximating the entire function.

Polyhedral Reformulation

Convexification considers the following conceptual reformulation.



$$\begin{aligned} \min \quad & cx + d^1y \\ \text{s.t.} \quad & (x, y) \in \text{conv}(\mathcal{F}) \end{aligned}$$

- This reformulation suggests a branch-and-cut algorithm similar to that used for solving MILPs DeNegre and Ralphs [2009].
- To get dual bounds, we optimize over a relaxed feasible region.
- We iteratively approximate conv($\mathcal{F}$) with linear inequalities.

Basic Principle: Disjunction

Definition 1 (Valid Disjunction)**.** A collection of disjoint sets $X_i \subseteq \mathbb{R}^{n_1+n_2}$ for $i = 1, ..., k$ represents a *valid disjunction* for $\mathcal{F}$ if

$$\mathcal{F} \subseteq \bigcup_{i=1}^{k} X_i.$$

Two classes of disjunction

- $(\bar{x}, \bar{y}) \in \mathcal{P} \setminus \mathcal{S} \Leftarrow$ must violate a variable disjunction.
- $(\bar{x}, \bar{y}) \in \mathcal{S} \setminus \mathcal{F} \Leftarrow$ must violate this valid disjunction (points in $\mathcal{P} \setminus \mathcal{S}$ may also).

$$\begin{pmatrix} A^1 x \geq b^1 - G^1 y^* \\ A^2 x \geq b^2 - G^2 y^* \\ d^2 y \leq d^2 y^* \end{pmatrix} \quad \text{OR} \quad \begin{pmatrix} A^1 x \not\geq b^1 - G^1 y^* \\ \text{OR} \\ A^2 x \not\geq b^2 - G^2 y^*, \end{pmatrix} \qquad \text{(OPT-DISJ)}$$

where $y^* \in \mathcal{P}_2(\bar{x}) \cap Y$ and $d^2 \bar{y} > d^2 y^*$.

- Note that such a $y^* \neq \bar{y}$ must exist when $\bar{y} \in \mathcal{S}$.

Basic Principle: Identifying Infeasible Solutions

- Just as in MILP, an important key to solving MIBLPs is identifying large (convex) subsets of $\mathcal{P}$ that contain no member of $\mathcal{F}$.
- This should be done by carefully exploiting available information and keeping computational overhead low.
- Two methods for proving a solution infeasible underlie much of the methodology for doing this.

Second-level Improving Solutions

Let $(x, y) \in \mathcal{P}$ and $y^* \in \mathcal{P}_2(x) \cap Y$. Then $d^2 y > d^2 y^* \Rightarrow (x, y) \notin \mathcal{F}$.

Second-level Improving Directions

Let $(x, y) \in \mathcal{P}$ and $\Delta y \in \mathbb{Z}^{n_2}$ such that $d^2 \Delta y < 0$. Then $y + \Delta y \in \mathcal{P}_2(x) \Rightarrow (x, y) \notin \mathcal{F}$.

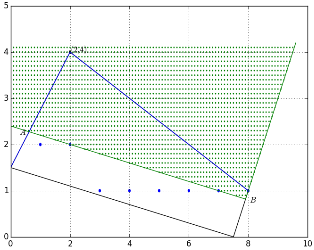
Basic Principle: Bilevel Free Sets [Fischetti et al., 2018]

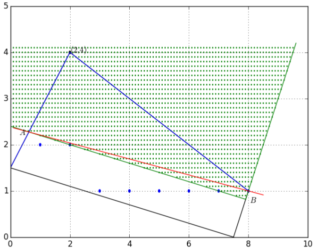
Bilevel Free Set

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

General Recipe for Valid Inequalities

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





Outline

Branch-and-Cut Algorithm

- The basic framework is very similar to that used for solving MILPs, but with many subtle differences.

Components

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

- This talk will focus on the highlighted areas.

Challenges

- On the surface, branch-and-cut for MIBLPs looks similar to that for MILPs.
- Digging deeper, they are *very* different and there is a lot we still don't know.
- We have to tear down the solver and re-examine every aspect of its performance. Some challenges that remain.
  - In contrast with MILP, it can be difficult to move the bound in the root node.
  - Thus, we don't have a very good approximation of $\text{conv}(\mathcal{F})$ in the early stages.
  - This (probably) makes it difficult to predict the impact of branching.
  - Because the disjunctions used for cutting are much stronger than those used for branching, it seems more important to emphasize cuts.
  - On the other hand, cuts are expensive to generate.
  - We don't really know how to integrate MILP cuts and MIBLP cuts.
  - In general, the interaction of cutting and branching is much more intricate, which makes good control mechanisms vitally important.
  - Specific properties of instances (e.g., degree of alignment of objectives) can affect performance dramatically and this needs to be understood better.

Dual Bound

Possible relaxations

1. Remove the *optimality constraint of the second-level problem* (MIP relaxation)

$$\mathcal{S} = \left\{(x, y) \in \mathbb{R}_+^{n_1 \times n_2} \;\middle|\; x \in X, y \in \mathcal{P}_1(x) \cap \mathcal{P}_2(x) \cap Y\right\}$$

2. Remove the *optimality constraint of the second-level problem* and the *integrality constraints* (LP relaxation)

$$\mathcal{P} = \left\{(x, y) \in \mathbb{R}_+^{n_1 \times n_2} \;\middle|\; y \in \mathcal{P}_1(x) \cap \mathcal{P}_2(x)\right\}$$

3. Something in between? (Neighborhood relaxation)

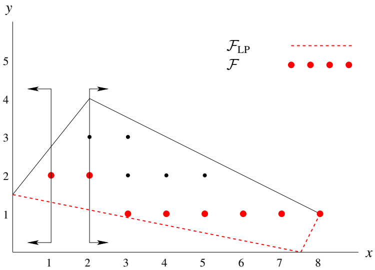
$$\mathcal{R}_{\mathcal{N}}(x) = \{y \in \mathrm{Proj}_y(\mathcal{S}) \mid d^2 y \le d^2 \bar{y} \quad \forall\, \bar{y} \in \mathcal{N}(y) \cap \mathrm{Proj}_y(\mathcal{S})\}$$

where $\mathcal{N}(y)$ is a neighborhood of y Xueyu et al. [2022].

Branching

- In general, there has been very little study of how to branch in solving MIBLPs.
- What we do today is use roughly the same rules for branching that are used in solving MILPs.

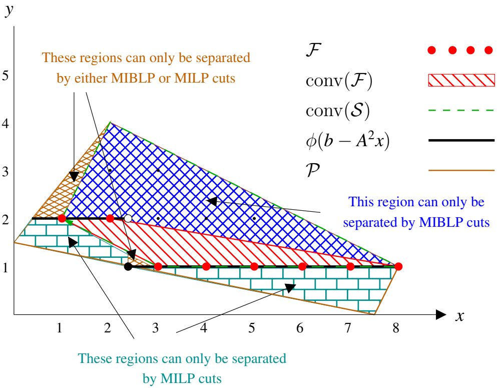
- Does this make sense? Not always...
- We may need to branch on variables that already have an integer value (more on this).
- MILP strategies predict the impact of branching using the dual bound as a proxy.
- In MIBLP, this is probably not a very good proxy.



- One of the open challenges is to figure out a better prediction function.
- Currently, `MibS` uses straightforward pseudo-cost branching.

Cut Generation

- Unlike in MILP, we have several distinct classes of infeasible solution.
- Each requires different handling.
- Which types arise is (somewhat) dictated by the objective alignment.



1. $(\bar{x}, \bar{y}) \in \mathbb{R}^{n_1+n_2}$ for which $d^2\bar{y} \leq \phi(b^2 - A^2\bar{x}) \Leftarrow (\bar{x}, \bar{y}) \notin \mathcal{S}$
   - Need MILP cuts, but it's not easy to recognize this case!
2. $(\bar{x}, \bar{y}) \in \mathbb{R}^{n_1+n_2}$ for which $d^2\bar{y} > \phi(b^2 - A^2\bar{x}) \Leftarrow (\bar{x}, \bar{y})$ may or may not be in $\mathcal{S}$.
   - $\bar{x} \in X \Leftarrow$ Can evaluate $\phi(b^2 - A^2\bar{x})$ or $\Xi(\bar{x})$ to separate.
   - $\bar{y} \in Y \Leftarrow$ Relatively easier to separate with MIBLP cuts
   - $\bar{x} \notin X, \bar{y} \notin Y \Leftarrow$ Important, but tricky case!

Classes of Inequalities Valid for MIBLPs

Generalized Chvátal Cuts

- Let $C = \{(x, y) \in \mathcal{P} \mid \pi^x x + \pi^y y \leq \beta\}$ be a BFS, where $(\pi^x, \pi^y) \in X \times Y$, $\beta \in \mathbb{Z}$.
- Then $(\pi^x, \pi^y, \beta + 1)$ is valid for $\mathcal{F}$.

Intersection Cuts

- Let C be a convex set containing no improving solutions and let (x, y) be an extreme point of $\mathcal{P}$ in the interior of C.
- Then the intersection cut with respect to C and (x, y) is valid for $\mathcal{F}$.

Benders Cuts

- Let $\bar{\psi} : \mathbb{R}^{n_1} \to \mathbb{R}$ be such that $\bar{\psi}(x) \geq \phi(b^2 - A^2 x)$ (a *primal function*).
- Then $C = \{(x, y) \in \mathcal{P} \mid d^2 y \geq \bar{\psi}(x)$ is a BFS and $d^2 y \leq \bar{\psi}(x)$ for all $(x, y) \in \mathcal{F}$.

Classes Implemented in `MibS`

- MILP cuts.
- Generalized Chvátal (Integer no-good cut) [DeNegre and Ralphs, 2009]
- Benders Cuts
  - Benders Binary Cut [DeNegre, 2011]
  - Benders Interdiction Cut [Ralphs et al., 2015, Caprara et al., 2014]
  - Benders Bound Cut [Tahernejad, 2019]
- Intersection cuts [Fischetti et al., 2017, 2018]
  - Improving Solution (Types I and II)
  - Improving Direction
  - Hypercube
- Generalized no-good cut [DeNegre, 2011]

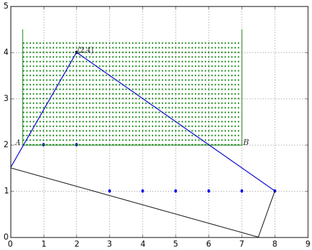
Improving Solution Intersection Cut (ISIC)

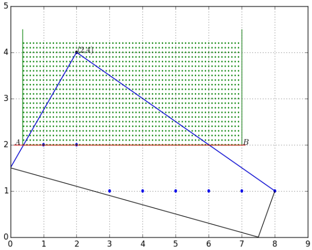
- For simplicity, assume all problem data are integral.
- Let $(\hat{x}, \hat{y})$ be an extreme point of $\mathcal{P}$ such that $d^2\hat{y} > d^2y^*$ for some $y^* \in \mathcal{P}_2(\hat{x}) \cap Y$ ($\Leftarrow$ the improving solution).

Bilevel Free Set

$$C = \left\{(x, y) \in \mathbb{R}^{n_1 \times n_2} \;\middle|\; d^2y \geq d^2y^*, A^2x \geq b^2 - G^2y^* - 1\right\}.$$

- The basic logic is very similar to the Benders cut.
- Crucially, note that we don't need $\hat{x} \in X$ or $\hat{y} \in Y$.





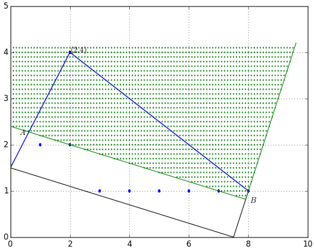
Improving Direction Intersection Cut (IDIC)

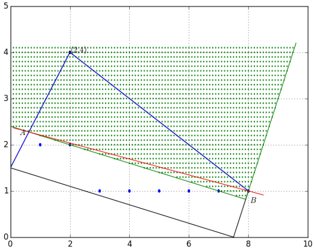
- Once again, assume all problem data are integral.
- Let $(\hat{x}, \hat{y})$ be an extreme point of $\mathcal{P}$ and let $\Delta y \in \mathbb{Z}^{n_2}$ ($\Leftarrow$ the improving direction) such that $\hat{y} + \Delta y \in \mathcal{P}_2(\hat{x})$ and $d^2 \Delta y < 0$

Bilevel Free Set

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

- Once again, note that we don't need $\hat{x} \in X$ or $\hat{y} \in Y$.





Comparing the Classes Analytically : Size of $\mathrm{int}(C)$

Generalized Chvátal cuts

Only a single point $(x, y) \in \mathcal{S} \setminus \mathcal{F}$

HICs and Generalized no-good cuts

All $(\hat{x}, y) \in \mathcal{S}$ (feasible or not) for some $\hat{x} \in X$ such that $\Xi(\hat{x})$ is known
$\Rightarrow$ All combinations of a **fixed** $\hat{x}$ with any y.

Benders cuts and ISICs

All $(x, y) \in \mathcal{P}$ such that $y^* \in \mathcal{P}_2(x)$ and $d^2 y > d^2 y^*$
$\Rightarrow$ All (x, y^*) for which a **fixed** y^* proves infeasibility.

IDICs

$(x, y) \in \mathcal{P}$ such that Δy is an improving feasible direction for y, given x
$\Rightarrow$ All (x, y) for which **a fixed** Δy proves infeasibility.

ISICs versus IDICs

- For general IBLPs, it seems apparent that ISICs and IDICs provide the most "bang for the buck," but how do they compare to each other?
  - Both classes of inequalities can be used to separate arbitrary fractional solutions, which sets them apart.
  - Both also require solving an MILP subproblem.
  - The feasible regions of these subproblems are even (in a certain sense) equivalent.
  - Let
$$\mathcal{W}(\hat{x},\hat{y}) = \left\{ w \in \mathbb{Z}^{r_2} \times \mathbb{R}^{n_2 - r_2} \mid d^2 w < 0,\ \hat{y} + w \in \mathcal{P}_2(\hat{x}) \right\}.$$
   be the set of improving feasible directions with respect to $(\hat{x},\hat{y}) \in \mathcal{P}$.
  - Then for any $(x,y) \in \mathcal{S}$,

$$(x,y) \in \mathcal{F} \Leftrightarrow \mathcal{W}(\hat{x},\hat{y}) = \emptyset \Leftrightarrow \exists y^* \in \mathcal{P}_2(x) \cap Y \text{ with } d^2 y^* < d^2 y$$

- The crucial difference is that the construction of large bilevel free sets using the two different recipes requires much different solutions/directions.
  - To construct large bilevel free sets with IDICs, directions should be *short*
  - To construct large bilevel free sets with ISICs, solutions should be *high quality*.

Outline

Software Framework

`MibS` is an open-source solver for MIBLPs.

- Implements the branch-and-cut algorithm for MIBLPs described here.
- Implemented in C++.
- Built on top of the BLIS MILP solver [Xu et al., 2009].
- Employs software available from the *Computational Infrastructure for Operations Research (COIN-OR)* repository
  - *COIN High Performance Parallel Search (CHiPPS)*: To manage the global branch-and-bound
  - *SYMPHONY*: To solve the required MIPs (can also use Cbc or CPLEX)
  - *COIN LP Solver (CLP)*: To solve the LPs arising in the branch and cut.
  - *Cut Generation Library (CGL)*: To generate cutting planes within both SYMPHONY and `MibS`
  - *Open Solver Interface (OSI)*: To interface with other solvers

Data Sets

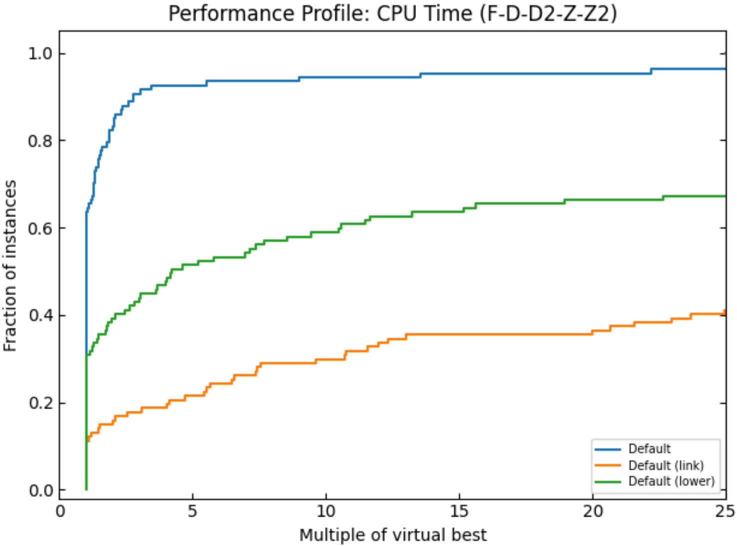
Table: The summary of data sets

Data Set	#	VT	V#	C#	Align	Notes
INT-DEN	300	B B	10-40 10-40	1 11-41	-1	Interdiction DeNegre [2011]
DEN	50	I I	5-15 5-15	0 20	Varies	DeNegre [2011]
DEN2	110	I I	5-10 5-20	0 5-15	Varies	DeNegre [2011]
ZHANG	30	B I	50-80 70-110	0 6-7	0.6-0.8	Zhang and Ozaltın [2017]
ZHANG2	30	I I	50-80 70-110	0 6-7	0.6-0.8	Zhang and Ozaltın [2017]
FIS	57	B B	Varies	Varies	-1	MIPLIB Fischetti et al. [2018]
XU	100	I IC	10-460 4-184	10-460 4-184	≈ 0	Mixed Xu and Wang [2014]

Computational Experiments

- Nearly 20K CPU hours with four different versions of `MibS` with both SYMPHONY and CPLEX as subsolvers (and filmosi for comparison).
- Run on the COR@L cluster: 14 nodes, dual 8-core .8 GHz CPUs, 32 Gb memory
- Instances that took less than 5 seconds to solve for all versions were filtered.
- Which data sets are included are indicated in the title (X = XU, F=FIS, etc.)

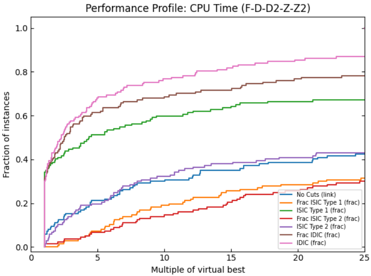
Comparing Branching Schemes

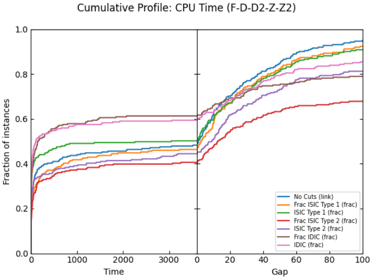


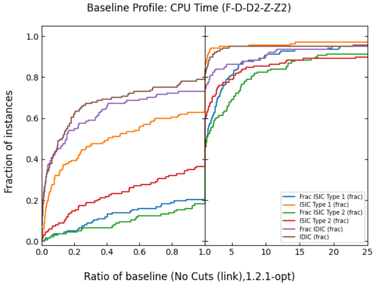
Comparing Cuts Empirically

- In the MILP context, it is typical to compare cuts using a closure bound or root gap to isolate the separate effects of branching and cutting.
- Results are displayed using a combination of
  - Performance profiles (CDF of the ratio
  - Cumulative profiles
  - Baseline profiles
- Performance measure
  - CPU time
  - Nodes evaluated
  - Root bound

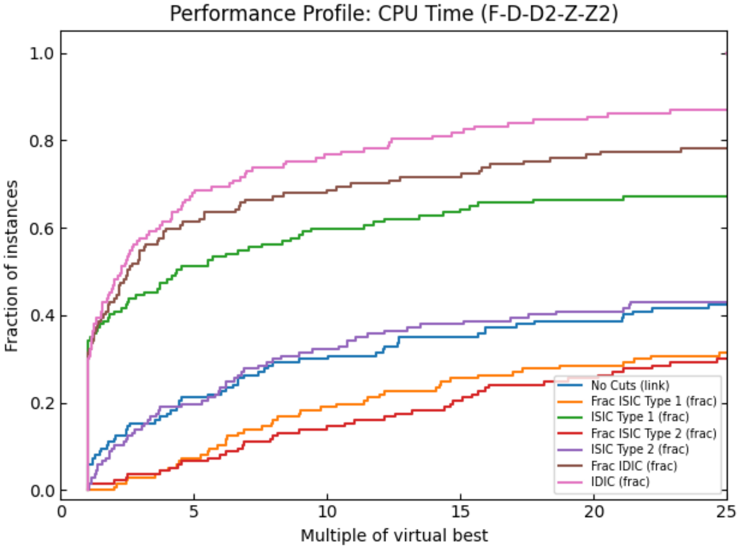
Summary Results (IDICs versus ISICs)



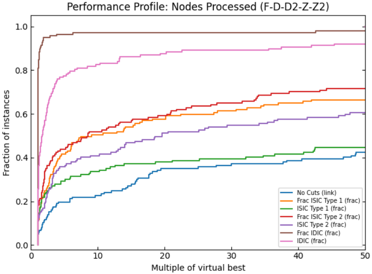


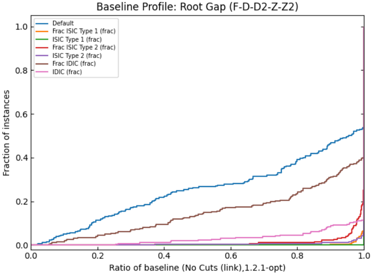


Summary Results (IDICs versus ISICs)

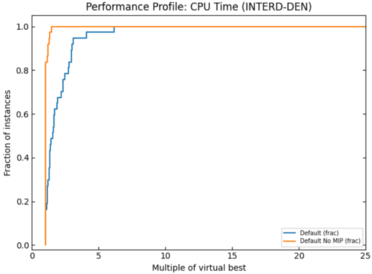


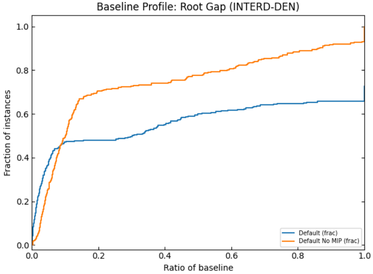
Summary Results (IDICs versus ISICs)

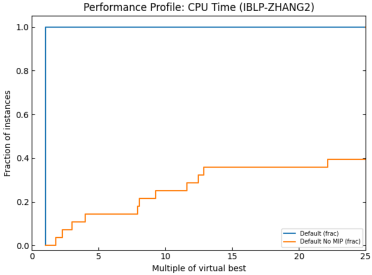


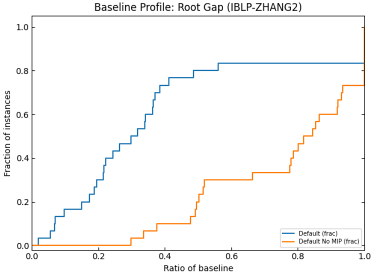


Do MILP Cuts Help?

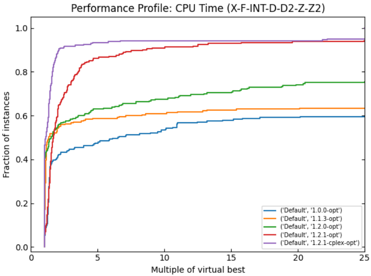


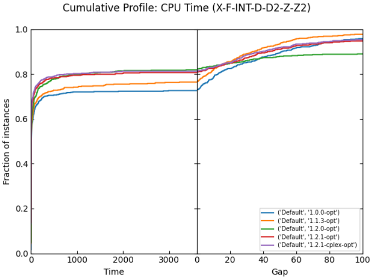


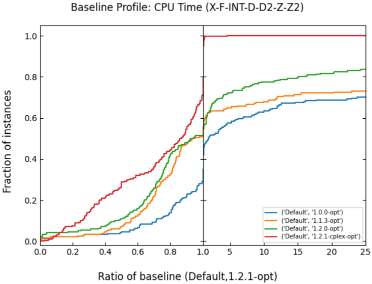




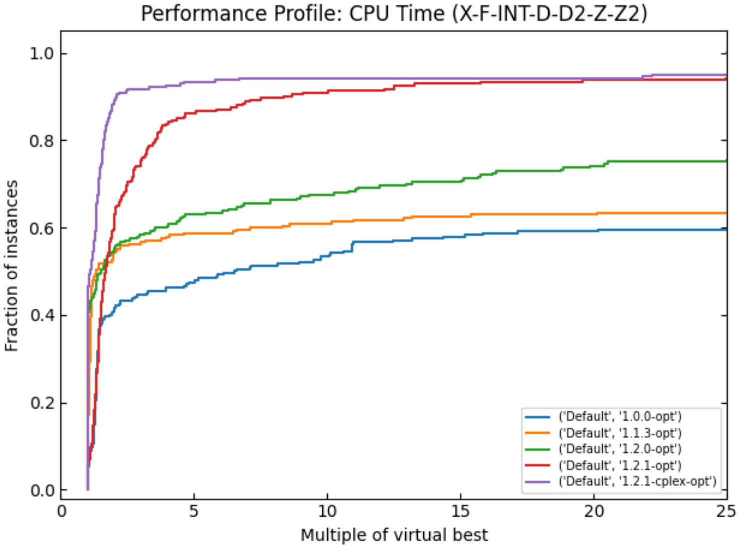
Overall Results: Different Versions of `MibS`



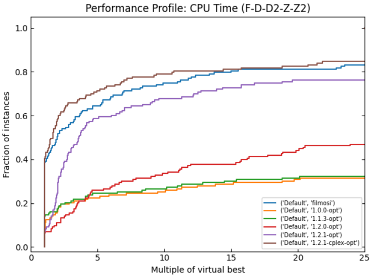


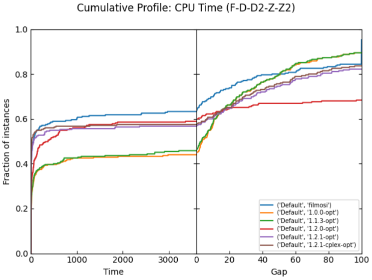


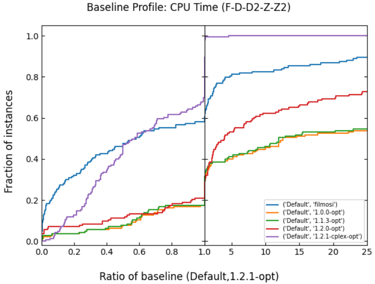
Overall Results: Different Versions of `MibS`



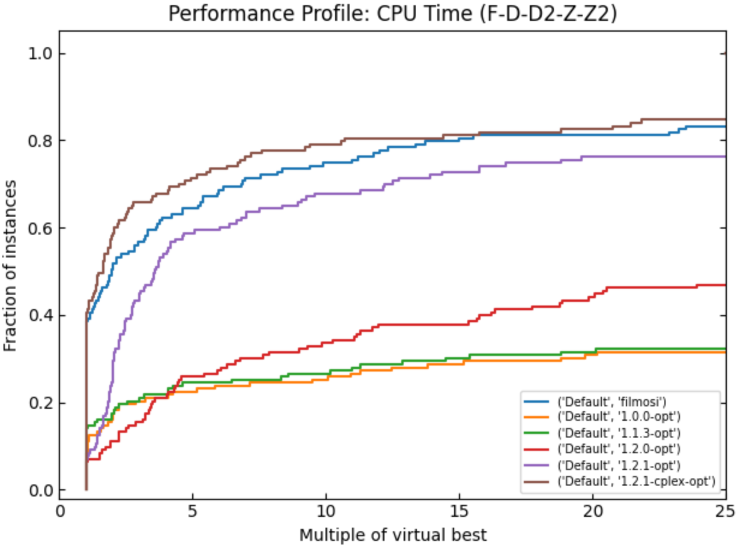
Overall Results: Comparing MibS with filmosi







Overall Results: Different Versions of MibS



Outline

Future Directions

- **There are still many avenues for improving performance and much low-hanging fruit.**
  - Improved branching
  - Better dynamic control mechanisms for cut generation (better integration of MIBLP and MILP cuts)
  - Warm-starting of subproblem solvers (SYMPHONY)
  - Pools of solutions/directions/cuts
  - ...
- **Existing capabilities that need further development**.
  - Stochastic bilevel solver
  - Pessimistic solver
  - Bounded rationality

The Dream

How would we design a solver if we could do it from the ground up?

- No explicit subsolvers, just one tightly integrated solver.
- Flexible reaction sets (bounded rationality).
- Flexible base relaxations.
- Solver based completely on improving directions?

Conclusions

- Solutions of MIBLPs is where solution of MILPs was 15 years ago.
- The basic theory is well-developed, but in practice, solvers are well-tuned bags of tricks.
- MILP solvers are still improving, thanks largely to commercial viability and fierce competition.
- It remains to be seen if MIBLP solvers will follow a similar trajectory.

References I

A. Caprara, M. Carvalho, A. Lodi, and G.J. Woeginger. Bilevel knapsack with interdiction constraints. Technical Report OR-14-4, University of Bologna, 2014.

S. DeNegre. *Interdiction and Discrete Bilevel Linear Programming*. PhD, Lehigh University, 2011. URL `http://coral.ie.lehigh.edu/~ted/files/papers/ScottDeNegreDissertation11.pdf`.

S. DeNegre and T.K. Ralphs. A Branch-and-Cut Algorithm for Bilevel Integer Programming. In *Proceedings of the Eleventh INFORMS Computing Society Meeting*, pages 65–78, 2009. doi: 10.1007/978-0-387-88843-9_4. URL `http://coral.ie.lehigh.edu/~ted/files/papers/BILEVEL08.pdf`.

M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research*, 65(6):1615–1637, 2017.

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.

References II

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

K. Ralphs, T, S. Tahernejad, S. DeNegre, M. Güzelsoy, and A. Hassanzadeh. Bilevel integer optimization: Theory and algorithms. International Symposium on Mathematical Programming 2015, 2015.

S. Tahernejad. *Two-stage Mixed Integer Stochastic Bilevel Linear Optimization*. PhD, Lehigh University, 2019.

P. Xu and L. Wang. An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions. *Computers & operations research*, 41:309–318, 2014.

Y. Xu, T.K. Ralphs, L. Ladányi, and M.J. Saltzman. Computational Experience with a Software Framework for Parallel Integer Programming. *The INFORMS Journal on Computing*, 21:383–397, 2009. doi: 10.1287/ijoc.1090.0347. URL `http://coral.ie.lehigh.edu/~ted/files/papers/CHiPPS-Rev.pdf`.

References III

Shi Xueyu, 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—-51, 2022. doi: 10.1007/s12532-022-00227-z. URL `http://coral.ie.lehigh.edu/~ted/files/papers/HierarchyOfBounds20.pdf`.

Junlong Zhang and Osman Y Ozaltın. A branch-and-cut algorithm for discrete bilevel linear programs. *Optim. Online*, 2017.