A Branch-and-Cut Algorithm for Mixed Integer Bilevel Linear Optimization Problems and Its Implementation

Sahar Tahernejad[1]
Joint work with Ted Ralphs[1], Scott T. DeNegre[2]

[1]COR@L Lab, Department of Industrial and Systems Engineering, Lehigh University
[2]Hospital for Special Surgery, New York, New York

INFORMS, Houston, Texas,October 25, 2017

ISE Industrial and Systems Engineering

COR@L COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

Mixed Integer Bilevel Linear Optimization Problems

- *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

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

where

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

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

Mixed Integer Bilevel Linear Optimization Problems

Value Function

$$\phi(\beta) = \min \left\{ d^2 y \;\middle|\; G^2 y \geq \beta, y \in Y \right\} \quad \forall \beta \in \mathbb{R}^{m_2}. \tag{VF}$$

Value function returns the optimal value of the second-level problem with respect to its right-hand-side.

Risk Function

$$\Xi(x) = \min \left\{ d^1 y \;\middle|\; y \in \mathcal{P}_1(x), y \in \operatorname{argmin}\{ d^2 y \mid y \in \mathcal{P}_2(x) \cap Y \} \right\}. \tag{RF}$$

Risk function encodes the part of the first-level objective value that depends on the response to $x \in X$ in the second level.

Software Framework

MibS software

- is an open-source solver for MIBLPs.
- works based on our branch-and-cut algorithm for MIBLPs.
- is implemented in C++.
- is built on top of the BLIS solver [Xu et al., 2009].
- employs different 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
  - *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

Branch-and-Cut Algorithm

- The algorithm is based on the basic algorithmic framework originally described by DeNegre and Ralphs [2009], but with many additional enhancements.
- The algorithm is comprised of the following steps
  - Bounding
    - Lower bound
    - Upper bound
  - Pruning
  - Cutting
  - Branching
    - Branching on linking variables
    - Branching on fractional variables

Lower Bound

Bilevel Feasible Region

$$\mathcal{F} = \left\{ (x,y) \in \mathbb{R}_+^{n_1 \times n_2} \;\middle|\; x \in X, y \in \mathcal{P}_1(x) \cap \mathcal{P}_2(x) \cap Y, d^2 y \leq \phi(b^2 - A^2 x) \right\}$$

Removing the *optimality constraint of the second-level problem* and the *integrality constraints*

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

$$\Downarrow$$

$$L^t = \min_{(x,y) \in \mathcal{P}^t} cx + d^1 y$$

Upper Bound

- The upper bound is derived by exhibiting a bilevel feasible solution.
- **Question:** How to find bilevel feasible solutions?
- Let (x^t, y^t) be the optimal solution of the relaxation problem at node t.
- (x^t, y^t) can be exploited to generate bilevel feasible solutions.

Upper Bound

◇(x^t, y^t) *may* be bilevel feasible ⇒ Feasibility check

- (x^t, y^t) does not satisfy integrality requirements ⇒ *infeasible*
- (x^t, y^t) satisfies integrality requirements
  - Solve the second-level problem with $\beta = b^2 - A^2 x^t$

$$\phi(b^2 - A^2 x^t) = \min \left\{ d^2 y \;\middle|\; G^2 y \geq b^2 - A^2 x^t, y \in Y \right\}$$

  - Let $\hat{y}^t$ be the optimal solution.
    - $d^2 \hat{y}^t = d^2 y^t$ ⇒ bilevel feasible
    - $d^2 \hat{y}^t < d^2 y^t$ ⇒ *infeasible*

◇ If $x^t \in X$ and $A^1 x^t + G^1 \hat{y}^t \geq b^1 \Rightarrow (x^t, \hat{y}^t)$ is bilevel feasible

Linking Variables

- Linking variables : The set of of first-level variables with non-zero coefficients in the second-level problem $\Rightarrow x_L$
- Assumption: All linking variables are discrete ones $\Rightarrow$ The optimal solution of MIBLP is attainable.

For the vectors x^1 and $x^2 \in \mathcal{X}$ with $x_L^1 = x_L^2$, we have

$$\phi(b^2 - A^2x^1) = \phi(b^2 - A^2x^2) \quad \text{and} \quad \Xi(x^1) = \Xi(x^2).$$

Linking Variables

For the vectors x^1 and $x^2 \in \mathcal{X}$ with $x^1_L = x^2_L$, we have

$$\phi(b^2 - A^2x^1) = \phi(b^2 - A^2x^2) \quad \text{and} \quad \Xi(x^1) = \Xi(x^2).$$

$$\Downarrow$$

- The best bilevel feasible solution (x,y) with $x_L = \gamma \in \mathbb{Z}^L$ can be obtained by solving just one MILP.

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

Upper Bound

$\diamond$ If $x_L^t \in \mathbb{Z}^L \Rightarrow (\hat{x}, \hat{y})$ is bilevel feasible where it is obtained by solving the problem (UB) with $x_L = x_L^t$.

Note that $x_L^t \in \mathbb{Z}^L$ does not mean that a bilevel feasible solution can be found inevitably because

- $\phi(b^2 - A^2 x^t)$ may be $+\infty$.
- Problem (UB) may be *infeasible*.

Additional Enhancements

- In contrast with MILPs, we do *not* check the bilevel feasibility of (x^t, y^t) necessarily.
- The relevant parameters in MibS are
  - `solveSecondLevelWhenXYVarsInt`
  - `solveSecondLevelWhenXVarsInt`
  - `solveSecondLevelWhenLVarsInt`
  - `solveSecondLevelWhenLVarsFixed`
- MibS does *not* always allow solving problem (UB) when $x_L^t \in \mathbb{Z}^L$.
- The relevant parameters are
  - `computeBestUBWhenXVarsInt`
  - `computeBestUBWhenLVarsInt`
  - `computeBestUBWhenLVarsFixed`

Pruning

In a similar way as all branch-and-bound algorithms, pruning of node t occurs whenever

1. The relaxation problem is *infeasible*.
2. The optimal value of the relaxation problem is not better than the current upper bound.
3. The optimal solution of the relaxation problem is bilevel feasible.

   There is one additional case

4. All linking variables are fixed.

Although node t can be pruned after fixing all linking variables, MibS may *not* do it.

Cutting

- **Question:** When may MibS generate a cut?

  1. Removing $(x^t, y^t) \notin X \times Y$
  2. Removing $(x^t, y^t) \in X \times Y$, but $d^2 y^t > \phi(b^2 - A^2 x^t)$
  3. Removing all $x \in \mathcal{P}^x$ with $x_L = \gamma \in \mathbb{Z}^L$

  Note that P^x denotes the projection of the relaxation feasible region on the set of first-level variables, i.e., $\mathcal{P}^x = \text{proj}_x(\mathcal{P})$.

- With respect to the goals of generating valid inequalities, the set of valid inequalities for MIBLPs can be classified.

Cutting

With respect to the goals of generating valid inequalities, the set of valid inequalities for MIBLPs can be classified.

Removing $(x^t, y^t) \notin X \times Y$	$\Rightarrow$	*Feasibility cuts*: Valid inequalities which are violated by (x^t, y^t), but valid for $\text{conv}\left(\left\{(x, y) \in \mathcal{S} \mid cx + d^1 y < U\right\}\right)$.
Removing $(x^t, y^t) \in X \times Y$, but $d^2 y^t > \phi(b^2 - A^2 x^t)$	$\Rightarrow$	*Optimality cuts*: Valid inequalities which are violated by (x^t, y^t), but valid for $\text{conv}\left(\left\{(x, y) \in \mathcal{F} \mid cx + d^1 y < U\right\}\right)$.
Removing all $x \in \mathcal{P}^x$ with $x_L = \lambda \in \mathbb{Z}^L$	$\Rightarrow$	*Projected optimality cuts*: Valid inequalities which are violated by $x \in \mathcal{P}^x$ with $x_L = \gamma \in \mathbb{Z}^L$, but valid for $\text{conv}(\{(x, y) \in \mathcal{F} \mid cx + \xi(x) < U\})$.

Cutting

- Feasibility cuts: This set includes all valid inequalities work for the *MILPs*.
- Optimality cuts:
  - Integer no-good cut [DeNegre and Ralphs, 2009]
  - Increasing objective cut [DeNegre, 2011]
  - Benders cut
  - Intersection cut [Fischetti et al., 2016b]
  - Bound cut
- Projected optimality cuts:
  - Generalized no-good cut

Branching

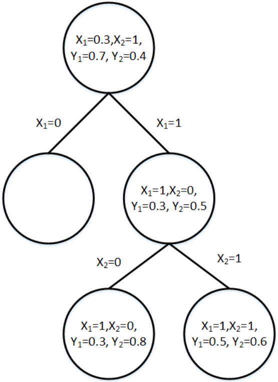
Question: When may MibS employ branching?

1. the solution $(x^t, y^t) \notin \mathcal{F}$ because
   - $(x^t, y^t) \notin X \times Y$
   - $d^2 y^t > \phi(b^2 - A^2 x^t)$
2. $(x^t, y^t) \in X \times Y$ and we are *not* sure of its feasibility status.

Fractional Branching Scheme

- In a similar way as the *traditional* branching scheme for *MILPs*.
- **Main idea:** We *only* branch on discrete variables with *fractional values*.
- The branching object can be *either* a first- or second-level one.

- **Motivation:** A node can be pruned after fixing the linking variables.
- **Main idea:** We *only* consider branching on *linking variables* while any such variables *remain unfixed*.
- **Challenge:** $x_L^t \in \mathbb{Z}^L$ and x_L is not fixed.



Linking Solution Pool

For the vectors x^1 and $x^2 \in \mathcal{X}$ with $x^1_L = x^2_L$, we have

$$\phi(b^2 - A^2x^1) = \phi(b^2 - A^2x^2) \quad \text{and} \quad \Xi(x^1) = \Xi(x^2).$$

$\Downarrow$

Avoid the duplication of effort in evaluating the functions ϕ and Ξ

$\Downarrow$

Track the seen sub-vectors of values for linking variables in a *pool*

Computational Results

The investigated parameters are

- Branching scheme
- Feasibility check and computing best feasible solution
- Linking solution pool

The employed test sets are (*171 instances in total*)

- IBLP-DEN [DeNegre, 2011]
- IBLP-FIS [Fischetti et al., 2016a]
- MIBLP-XU [Xu and Wang, 2014]

Branching Scheme

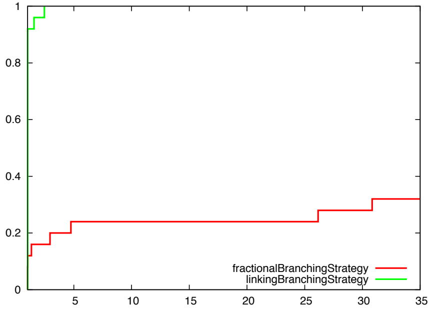


Figure: $r_1 \leq r_2$

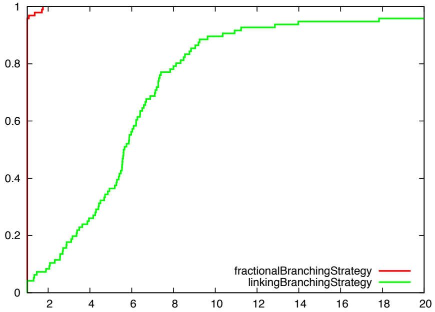


Figure: $r_1 > r_2$

Feasibility Check and Computing Best Feasible Solution

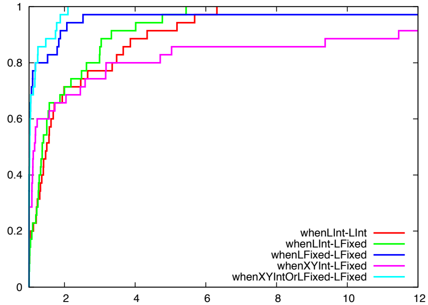


Figure: `IBLP-DEN` and `IBLP-FIS` sets

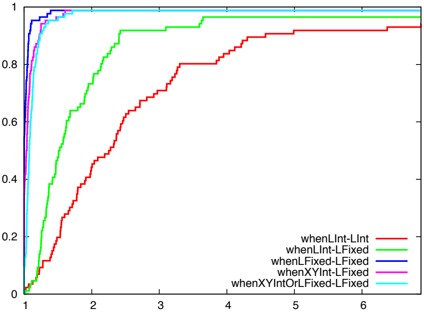


Figure: `MIBLP-XU` set

Linking Solution Pool

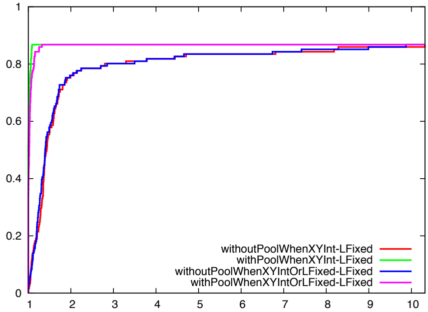


Figure: `fractional` branching strategy

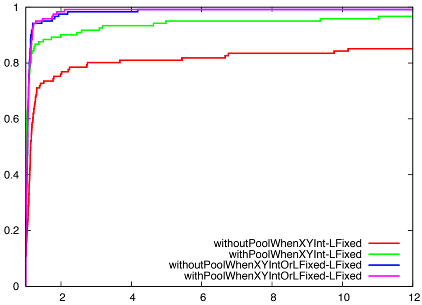


Figure: `linking` branching strategy

Conclusions

- A branch-and-cut algorithm for MIBLPs
  - Bounding
  - Pruning
  - Cutting
  - Branching
- An open-source solver for MIBLPs
- Generalizing the current algorithm to the stochastic MILPs

References I

Scott T DeNegre. *Interdiction and Discrete Bilevel Linear Programming*. Phd, Lehigh University, 2011. URL http://coral.ie.lehigh.edu/{~}ted/files/papers/ScottDeNegreDissertation11.pdf.

S.T. 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.

Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. Intersection cuts for bilevel optimization. Technical report, 2016a.

Matteo Fischetti, Ivana Ljubic, Michele Monaci, and Markus Sinnl. A new general-purpose algorithm for mixed-integer bilevel linear programs. Technical report, 2016b.

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

References II

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`.

Thank You!

sat214@lehigh.edu