Exact General-Purpose Solvers for Mixed-Integer Bilevel Linear Programs

Tutorial

I. Ljubić

ESSEC Business School of Paris, France

JFRO 2018, March 26, Paris

Based on the papers:

- **Part I:** M. Fischetti, I. Ljubić, M. Monaci, M. Sinnl: On the Use of Intersection Cuts for Bilevel Optimization, Mathematical Programming, to appear, 2018
- **Part II:** M. Fischetti, I. Ljubić, M. Monaci, M. Sinnl: A new general-purpose algorithm for mixed-integer bilevel linear programs, Operations Research 65(6): 1615-1637, 2017

Bilevel Optimization

General bilevel optimization problem

$$\min_{x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}} F(x, y) \tag{1}$$

$$G(x, y) \leq 0 \tag{2}$$

$$y \in \arg \min_{y' \in \mathbb{R}^{n_2}} \{ f(x, y') : g(x, y') \leq 0 \} \tag{3}$$

- Stackelberg game: two-person sequential game
- Leader takes follower's optimal reaction into account
- $N_x = \{1, \dots, n_1\}$, $N_y = \{1, \dots, n_2\}$
- $n = n_1 + n_2$: total number of decision variables

Bilevel Optimization

General bilevel optimization problem

$$\min_{x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}} F(x, y) \tag{1}$$

$$G(x, y) \leq 0 \tag{2}$$

$$y \in \arg\min_{y' \in \mathbb{R}^{n_2}} \{ f(x, y') : g(x, y') \leq 0 \} \tag{3}$$

Leader

- Stackelberg game: two-person sequential game
- Leader takes follower's optimal reaction into account
- $N_x = \{1, \ldots, n_1\}$, $N_y = \{1, \ldots, n_2\}$
- $n = n_1 + n_2$: total number of decision variables

Bilevel Optimization

General bilevel optimization problem

$$\min_{x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}} F(x, y) \quad (1)$$

$$G(x, y) \leq 0 \quad (2)$$

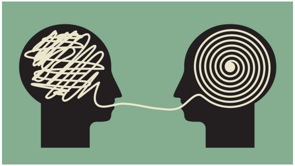
$$y \in \arg \min_{y' \in \mathbb{R}^{n_2}} \{ f(x, y') : g(x, y') \leq 0 \} \quad (3)$$

Leader

Follower

- Stackelberg game: two-person sequential game
- Leader takes follower's optimal reaction into account
- $N_x = \{1, \ldots, n_1\}$, $N_y = \{1, \ldots, n_2\}$
- $n = n_1 + n_2$: total number of decision variables

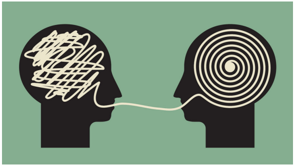
Optimistic vs Pessimistic Solution



The Stackelberg game under:

- **Perfect information:** both agents have perfect knowledge of each others strategy
- **Rationality:** agents act optimally, according to their respective goals
- **What if there are multiple optimal solutions for the follower?**
  - **Optimistic Solution:** among the follower's solution, the one leading to the best outcome for the leader is assumed
  - **Pessimistic Solution:** among the follower's solution, the one leading to the worst outcome for the leader is assumed

Optimistic vs Pessimistic Solution



The Stackelberg game under:

- **Perfect information:** both agents have perfect knowledge of each others strategy
- **Rationality:** agents act optimally, according to their respective goals
- **What if there are multiple optimal solutions for the follower?**
  - **Optimistic Solution:** among the follower's solution, the one leading to the best outcome for the leader is assumed
  - **Pessimistic Solution:** among the follower's solution, the one leading to the worst outcome for the leader is assumed

Our Focus: Mixed-Integer Bilevel Linear Programs (MIBLP)

$$\text{(MIBLP)} \quad \min c_x^T x + c_y^T y \tag{4}$$
$$G_x x + G_y y \leq 0 \tag{5}$$
$$y \in \arg\min\{d^T y : Ax + By \leq 0, \tag{6}$$
$$y_j \text{ integer}, \forall j \in J_y\} \tag{7}$$
$$x_j \text{ integer}, \forall j \in J_x \tag{8}$$
$$(x, y) \in \mathbb{R}^n \tag{9}$$

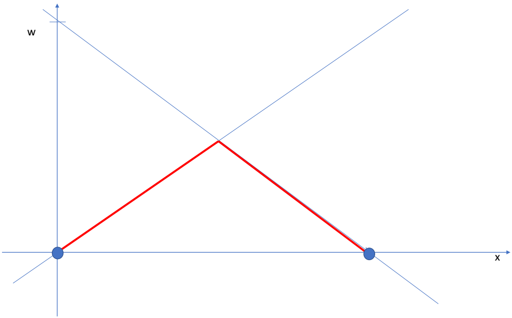
where c_x, c_y, G_x, G_y, A, B are given rational matrices/vectors of appropriate size.

Complexity

Bilevel Linear Programs

Bilevel LPs are strongly NP-hard (Audet et al. [1997], Hansen et al. [1992]).

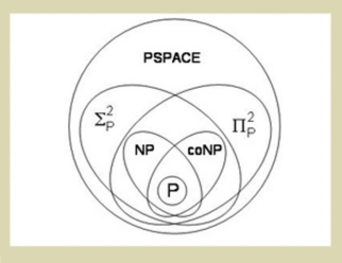
$$\min c^T x \quad Ax = b \quad x \in \{0,1\} \quad \Leftrightarrow \quad \min c^T x \quad Ax = b \quad v = 0 \quad v \in \arg\max\{w : w \leq x, w \leq 1 - x, w \geq 0\}$$



Complexity

Bilevel Mixed-Integer Linear Programs

MIBLP is Σ_2^P-hard (Lodi et al. [2014]): there is no way of formulating MIBLP as a MILP of polynomial size unless the polynomial hierarchy collapses.



Overview

Part I

- Develop a finitely convergent branch-and-bound approach (under certain conditions)
- Capable of dealing with unboundedness and infeasibility
- Introduce intersection cuts to speed-up convergence

Part II

- Introduce a fully-fledged branch-and-cut for MIBLPs

STEP 1: VALUE FUNCTION REFORMULATION

Our Focus: Mixed-Integer Bilevel Linear Programs (MIBLP)

Value Function Reformulation:

$$
\begin{aligned}
\text{(MIBLP)} \quad & \min c_x^T x + c_y^T y && (10)\\
& G_x x + G_y y \leq 0 && (11)\\
& Ax + By \leq 0 && (12)\\
& (x, y) \in \mathbb{R}^n && (13)\\
& d^T y \leq \Phi(x) && (14)\\
& x_j \text{ integer}, \quad \forall j \in J_x && (15)\\
& y_j \text{ integer}, \quad \forall j \in J_y && (16)
\end{aligned}
$$

where $\Phi(x)$ is non-convex, non-continuous:

$$\Phi(x) = \min\{d^T y : Ax + By \leq 0, \quad y_j \text{ integer}, \forall j \in J_y\}$$

- dropping $d^T y \leq \Phi(x)$ $\rightarrow$ **High Point Relaxation** (HPR) which is a MILP $\rightarrow$ we can use MILP solvers with all their tricks
- let $\overline{\text{HPR}}$ be LP-relaxation of HPR

Our Focus: Mixed-Integer Bilevel Linear Programs (MIBLP)

Value Function Reformulation:

I am a Mixed-Integer Linear Program (MILP) 😄

$$
\begin{aligned}
\text{(HPR)} \quad & \min c_x^T x + c_y^T y && (10)\\
& G_x x + G_y y \le 0 && (11)\\
& Ax + By \le 0 && (12)\\
& (x, y) \in \mathbb{R}^n && (13)\\
& && (14)\\
& x_j \text{ integer}, \quad \forall j \in J_x && (15)\\
& y_j \text{ integer}, \quad \forall j \in J_y && (16)
\end{aligned}
$$

where $\Phi(x)$ is non-convex, non-continuous:

$$\Phi(x) = \min\{d^T y : Ax + By \le 0, \quad y_j \text{ integer}, \forall j \in J_y\}$$

- dropping $d^T y \le \Phi(x) \rightarrow$ **High Point Relaxation** (HPR) which is a MILP $\rightarrow$ we can use MILP solvers with all their tricks
- let $\overline{\text{HPR}}$ be LP-relaxation of HPR

Our Focus: Mixed-Integer Bilevel Linear Programs (MIBLP)

Value Function Reformulation:

$$(\overline{\text{HPR}}) \quad \min c_x^T x + c_y^T y \tag{10}$$

$$G_x x + G_y y \leq 0 \tag{11}$$

$$Ax + By \leq 0 \tag{12}$$

$$(x, y) \in \mathbb{R}^n \tag{13}$$

(14)

(15)

(16)

where $\Phi(x)$ is non-convex, non-continuous:

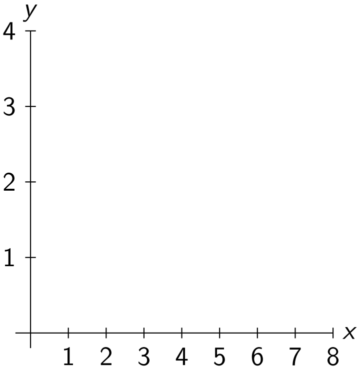
$$\Phi(x) = \min\{d^T y : Ax + By \leq 0, \quad y_j \text{ integer}, \forall j \in J_y\}$$

- dropping $d^T y \leq \Phi(x) \rightarrow$ **High Point Relaxation** (HPR) which is a MILP $\rightarrow$ we can use MILP solvers with all their tricks
- let $\overline{\text{HPR}}$ be LP-relaxation of HPR

Example

- notorious example from Moore and Bard [1990]
- HPR
- value-function reformulation

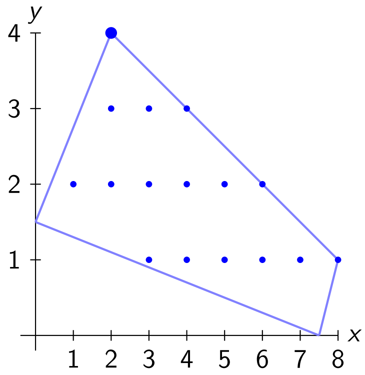
$$
\begin{aligned}
\min_{x \in \mathbb{Z}} \; & -x - 10y \\
y \in \arg\min_{y' \in \mathbb{Z}} \{ & y' : \\
-25x + 20y' &\leq 30 \\
x + 2y' &\leq 10 \\
2x - y' &\leq 15 \\
2x + 10y' &\geq 15 \}
\end{aligned}
$$



Example

- notorious example from Moore and Bard [1990]
- HPR
- value-function reformulation

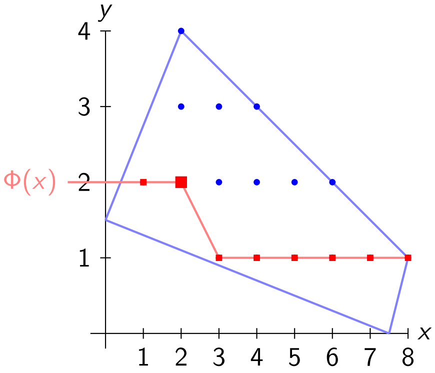
$$\min_{x,y\in\mathbb{Z}} -x - 10y$$
$$-25x + 20y \geq 30$$
$$x + 2y \leq 10$$
$$2x - y \leq 15$$
$$2x + 10y \geq 15$$



Example

- notorious example from Moore and Bard [1990]
- HPR
- value-function reformulation

$$\min_{x,y \in \mathbb{Z}} -x - 10y$$
$$-25x + 20y \geq 30$$
$$x + 2y \leq 10$$
$$2x - y \leq 15$$
$$2x + 10y \geq 15$$
$$y \leq \Phi(x)$$



General Idea

General Procedure

- Start with the HPR- (or $\overline{\text{HPR}}$-)relaxation
- Get rid of bilevel infeasible solutions on the fly
- Apply branch-and-bound or branch-and-cut algorithm

There are some unexpected difficulties along the way...

- Optimal solution can be unattainable
- HPR can be unbounded

(Un)expected Difficulties: **Unattainable Solutions**

Example from Köppe et al. [2010]

Continuous variables in the leader, integer variables in the follower ⇒ optimal solution may be **unattainable**

$$\inf_{x,y} \quad x - y$$
$$0 \leq x \leq 1$$
$$y \in \arg\min_{y'}\{y' : y' \geq x, 0 \leq y' \leq 1, y' \in \mathbb{Z}\}.$$

Equivalent to

$$\inf_{x}\{x - \lceil x \rceil : 0 \leq x \leq 1\}$$

Bilevel feasible set is neither convex nor closed.
Crucial assumption for us: follower subproblem depends only on integer leader variables $J_F \subseteq J_x$.

(Un)expected Difficulties: **Unattainable Solutions**

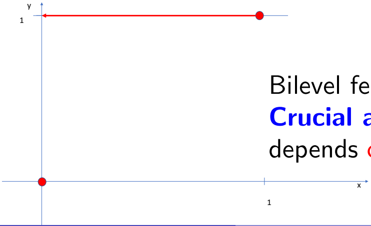
Example from Köppe et al. [2010]

Continuous variables in the leader, integer variables in the follower $\Rightarrow$ optimal solution may be **unattainable**

$$\inf_{x,y} \quad x - y$$
$$0 \le x \le 1$$
$$y \in \arg\min_{y'} \{y' : y' \ge x, 0 \le y' \le 1, y' \in \mathbb{Z}\}.$$

Equivalent to

$$\inf_{x} \{x - \lceil x \rceil : 0 \le x \le 1\}$$



Bilevel feasible set is neither convex nor closed.
Crucial assumption for us: follower subproblem depends only on integer leader variables $J_F \subseteq J_x$.

(Un)expected Difficulties: **Unbounded HPR-Relaxation**

Example from Xu and Wang [2014]

Unboundness of HPR-relaxation does not allow to draw conclusions on the optimal solution of MIBLP

- **unbounded**
- **infeasible**
- **admit an optimal solution**

$$\begin{aligned} \max_{x,y} \quad & x + y \\ & 0 \leq x \leq 2 \\ & x \in \mathbb{Z} \\ & y \in \arg\max_{y'}\{d \cdot y' : y' \geq x, y' \in \mathbb{Z}\}. \end{aligned}$$

$$\begin{aligned} \max_{x,y} \quad & x + y \\ & 0 \leq x \leq 2 \\ & y \geq x \\ & x, y \in \mathbb{Z} \end{aligned}$$

$d = 1 \quad \Rightarrow \Phi(x) = \infty$ (MIBLP infeasible)

$d = 0 \quad \Rightarrow \Phi(x)$ feasible for all $y \in \mathbb{Z}$ (MIBLP unbounded)

$d = -1 \quad \Rightarrow x^* = 2, y^* = 2$ (optimal MIBLP solution)

(Un)expected Difficulties: **Unbounded HPR-Relaxation**

Example from Xu and Wang [2014]

Unboundness of HPR-relaxation does not allow to draw conclusions on the optimal solution of MIBLP

- **unbounded**
- **infeasible**
- **admit an optimal solution**

$$\max_{x,y} \quad x + y$$
$$0 \leq x \leq 2$$
$$x \in \mathbb{Z}$$
$$y \in \arg\max_{y'} \{d \cdot y' : y' \geq x, y' \in \mathbb{Z}\}.$$

$$\max_{x,y} \quad x + y$$
$$0 \leq x \leq 2$$
$$y \geq x$$
$$x, y \in \mathbb{Z}$$

$d = 1 \quad \Rightarrow \Phi(x) = \infty$ (MIBLP infeasible)

$d = 0 \quad \Rightarrow \Phi(x)$ feasible for all $y \in \mathbb{Z}$ (MIBLP unbounded)

$d = -1 \quad \Rightarrow x^* = 2, y^* = 2$ (optimal MIBLP solution)

STEP 2: BRANCH-AND-CUT ALGORITHM

Assumption

All the integer-constrained variables x and y have finite lower and upper bounds both in HPR and in the follower MILP.

Assumption

Continuous leader variables x_j (if any) do not appear in the follower problem.

If for all HPR solutions, the follower MILP is unbounded $\Rightarrow$ MIBLP is infeasible. Preprocessing (solving a single LP) allows to check this. Hence:

Assumption

For an arbitrary HPR solution, the follower MILP is well defined.

Algorithm 1: A basic branch-and-bound scheme for MIBLP

Apply a standard LP-based B&B to HPR, inhibit incumbent update, and node-fathoming due to unboundedness of $\overline{\text{HPR}}$

for each unfathomed B&B node where standard branching cannot be performed do

  if $\overline{\text{HPR}}$ **is not unbounded then**

    Let (x^*, y^*) be the HPR solution at the current node;

    Compute $\Phi(x^*)$ by solving the follower MILP for $x = x^*$;

    if $d^T y^* \leq \Phi(x^*)$ **then**

      The current solution (x^*, y^*) is bilevel feasible: update the incumbent, fathom the current node, and **continue** with another node

    end

  end

  if all variables x_j **with** $j \in J_F$ **are fixed by branching** (x_F^*) **then**

    Refinement: Solve HPR with $x = x_F^*$, $d^T y \leq \Phi(x_F^*)$. If unbounded **return UNBOUNDED**;

    Possibly update the incumbent with the resulting solution $(\hat{x}, \hat{y})$, if any;

    Fathom the current node

  else

    Branch on any x_j $(j \in J_F)$ not fixed by branching yet (even if x_j^* is integer in the LP-solution at the node)

  end

end

Our Goal: Design MILP-based solver for MIBLP

For the rest of presentation: Assume HPR value is bounded.

Our Goal

solve MIBLP by using a standard **simplex-based branch-and-cut** algorithm; enforce $d^T y \leq \Phi(x)$ on the fly, by adding cutting planes

- given **optimal vertex** (x^*, y^*) of $\overline{\text{HPR}}$
  - (x^*, y^*) infeasible for HPR (i.e., fractional) $\rightarrow$ branch as usual
  - (x^*, y^*) feasible for HPR and $f(x^*, y^*) \leq \Phi(x^*)$ $\rightarrow$ update the incumbent as usual
  - (x^*, y^*) feasible for HPR and $f(x^*, y^*) > \Phi(x^*)$, i.e., **bilevel-infeasible** $\rightarrow$ we need to do something!

Our Goal: Design MILP-based solver for MIBLP

For the rest of presentation: Assume HPR value is bounded.

Our Goal

solve MIBLP by using a standard **simplex-based branch-and-cut** algorithm; enforce $d^T y \leq \Phi(x)$ on the fly, by adding cutting planes

- given **optimal vertex** (x^*, y^*) of $\overline{\text{HPR}}$
  - (x^*, y^*) infeasible for HPR (i.e., fractional) $\rightarrow$ branch as usual
  - (x^*, y^*) feasible for HPR and $f(x^*, y^*) \leq \Phi(x^*)$ $\rightarrow$ update the incumbent as usual
  - (x^*, y^*) feasible for HPR and $f(x^*, y^*) > \Phi(x^*)$, i.e., **bilevel-infeasible** $\rightarrow$ we need to do something!

Our Goal: Design MILP-based solver for MIBLP

For the rest of presentation: Assume HPR value is bounded.

Our Goal

solve MIBLP by using a standard **simplex-based branch-and-cut** algorithm; enforce $d^T y \leq \Phi(x)$ on the fly, by adding cutting planes

- given **optimal vertex** (x^*, y^*) of $\overline{\text{HPR}}$
  - (x^*, y^*) infeasible for HPR (i.e., fractional) $\rightarrow$ branch as usual
  - (x^*, y^*) feasible for HPR and $f(x^*, y^*) \leq \Phi(x^*)$ $\rightarrow$ update the incumbent as usual
  - (x^*, y^*) feasible for HPR and $f(x^*, y^*) > \Phi(x^*)$, i.e., **bilevel-infeasible** $\rightarrow$ we need to do something!

- Moore and Bard [1990] (Branch-and-Bound)
  - branching to cut-off bilevel infeasible solutions
  - no y-variables in leader-constraints
  - either all x-variables integer or all y-variables continuous

Our Goal: Design MILP-based solver for MIBLP

For the rest of presentation: Assume HPR value is bounded.

Our Goal

solve MIBLP by using a standard **simplex-based branch-and-cut** algorithm; enforce $d^T y \leq \Phi(x)$ on the fly, by adding cutting planes

- given **optimal vertex** (x^*, y^*) of $\overline{\text{HPR}}$
  - (x^*, y^*) infeasible for HPR (i.e., fractional) $\rightarrow$ branch as usual
  - (x^*, y^*) feasible for HPR and $f(x^*, y^*) \leq \Phi(x^*)$ $\rightarrow$ update the incumbent as usual
  - (x^*, y^*) feasible for HPR and $f(x^*, y^*) > \Phi(x^*)$, i.e., **bilevel-infeasible** $\rightarrow$ we need to do something!

- DeNegre [2011], DeNegre & Ralphs (Branch-and-Cut)
  - cuts based on slack
  - needs all variables and coefficients to be integer
  - open-source solver `MibS`

Our Goal: Design MILP-based solver for MIBLP

For the rest of presentation: Assume HPR value is bounded.

Our Goal

solve MIBLP by using a standard **simplex-based branch-and-cut** algorithm; enforce $d^T y \leq \Phi(x)$ on the fly, by adding cutting planes

- given **optimal vertex** (x^*, y^*) of $\overline{\text{HPR}}$
  - (x^*, y^*) infeasible for HPR (i.e., fractional) $\rightarrow$ branch as usual
  - (x^*, y^*) feasible for HPR and $f(x^*, y^*) \leq \Phi(x^*)$ $\rightarrow$ update the incumbent as usual
  - (x^*, y^*) feasible for HPR and $f(x^*, y^*) > \Phi(x^*)$, i.e., **bilevel-infeasible** $\rightarrow$ we need to do something!

- Xu and Wang [2014], Wang and Xu [2017] (Branch-and-Bound)
  - multiway branching to cut-off bilevel infeasible solutions
  - all x-variables integer and bounded, follower coefficients of x-variables must be integer

Our Goal: Design MILP-based solver for MIBLP

For the rest of presentation: Assume HPR value is bounded.

Our Goal

solve MIBLP by using a standard **simplex-based branch-and-cut** algorithm;
enforce $d^T y \leq \Phi(x)$ on the fly, by adding cutting planes

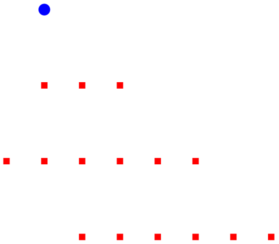
- given **optimal vertex** (x^*, y^*) of $\overline{\text{HPR}}$
  - (x^*, y^*) infeasible for HPR (i.e., fractional) $\rightarrow$ branch as usual
  - (x^*, y^*) feasible for HPR and $f(x^*, y^*) \leq \Phi(x^*)$ $\rightarrow$ update the incumbent as usual
  - (x^*, y^*) feasible for HPR and $f(x^*, y^*) > \Phi(x^*)$, i.e., **bilevel-infeasible** $\rightarrow$ we need to do something!

- **Our Approach** (Branch-and-Cut)
  - Use **Intersection Cuts** (Balas [1971]) to cut off bilevel infeasible solutions

STEP 3: INTERSECTION CUTS

Intersection Cuts (ICs)

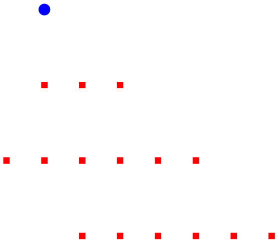
- powerful tool to separate a bilevel infeasible point (x^*, y^*) from a set of bilevel feasible points (X, Y) by a linear cut



- what we need to derive ICs
  - a cone pointed at (x^*, y^*) containing all (X, Y) (if (x^*, y^*) is a vertex of $\overline{HPR}$-**relaxation**, a possible cone comes from **LP-basis**)
  - a convex set S with (x^*, y^*) but no bilevel feasible points ($(x, y) \in (X, Y)$) in its **interior**
  - important: (x^*, y^*) should not be on the frontier of S.

Intersection Cuts (ICs)

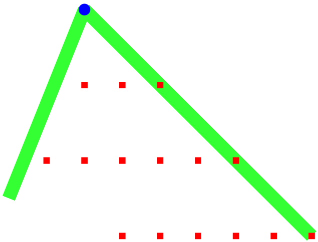
- powerful tool to separate a bilevel infeasible point (x^*, y^*) from a set of bilevel feasible points (X, Y) by a linear cut



- what we need to derive ICs
  - a cone pointed at (x^*, y^*) containing all (X, Y) (if (x^*, y^*) is a vertex of $\overline{HPR}$-**relaxation**, a possible cone comes from **LP-basis**)
  - a convex set S with (x^*, y^*) but no bilevel feasible points $((x, y) \in (X, Y))$ in its **interior**
  - important: (x^*, y^*) should not be on the frontier of S.

Intersection Cuts (ICs)

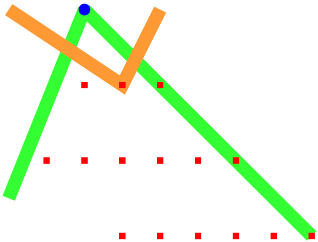
- powerful tool to separate a bilevel infeasible point (x^*, y^*) from a set of bilevel feasible points (X, Y) by a linear cut



- what we need to derive ICs
  - a cone pointed at (x^*, y^*) containing all (X, Y) (if (x^*, y^*) is a vertex of $\overline{HPR}$-**relaxation**, a possible cone comes from **LP-basis**)
  - a convex set S with (x^*, y^*) but no bilevel feasible points $((x, y) \in (X, Y))$ in its **interior**
  - important: (x^*, y^*) should not be on the frontier of S.

Intersection Cuts (ICs)

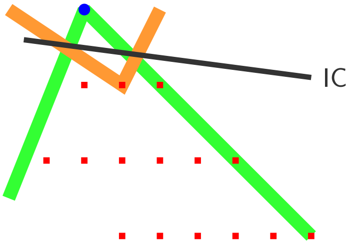
- powerful tool to separate a bilevel infeasible point (x^*, y^*) from a set of bilevel feasible points (X, Y) by a linear cut



- what we need to derive ICs
  - a cone pointed at (x^*, y^*) containing all (X, Y) (if (x^*, y^*) is a vertex of $\overline{HPR}$-**relaxation**, a possible cone comes from **LP-basis**)
  - a convex set S with (x^*, y^*) but no bilevel feasible points $((x, y) \in (X, Y))$ in its **interior**
  - important: (x^*, y^*) should not be on the frontier of S.

Intersection Cuts (ICs)

- powerful tool to separate a bilevel infeasible point (x^*, y^*) from a set of bilevel feasible points (X, Y) by a linear cut



- what we need to derive ICs
  - a cone pointed at (x^*, y^*) containing all (X, Y) (if (x^*, y^*) is a vertex of $\overline{HPR}$-**relaxation**, a possible cone comes from **LP-basis**)
  - a convex set S with (x^*, y^*) but no bilevel feasible points $((x, y) \in (X, Y))$ in its **interior**
  - important: (x^*, y^*) should not be on the frontier of S.

Intersection Cuts for Bilevel Optimization

- we need a **bilevel-free set** S

Theorem

For any feasible solution of the follower $\hat{y} \in \mathbb{R}^{n_2}$, the set

$$S(\hat{y}) = \{(x, y) \in \mathbb{R}^n : d^T y > d^T \hat{y},\ Ax + B\hat{y} \le b\}$$

does not contain any bilevel-feasible point (not even on its frontier).

- note: $S(\hat{y})$ is a **polyhedron**
- problem: **bilevel-infeasible** (x^*, y^*) can be on the **frontier** of bilevel-free set $S \rightarrow$ IC based on $S(\hat{y})$ may not be able to cut off (x^*, y^*)

Intersection Cuts for Bilevel Optimization

- we need a **bilevel-free set** S

Theorem

For any feasible solution of the follower $\hat{y} \in \mathbb{R}^{n_2}$, the set

$$S(\hat{y}) = \{(x, y) \in \mathbb{R}^n : d^T y > d^T \hat{y},\ Ax + B\hat{y} \leq b\}$$

does not contain any bilevel-feasible point (not even on its frontier).

- note: $S(\hat{y})$ is a **polyhedron**
- problem: **bilevel-infeasible** (x^*, y^*) can be on the **frontier** of bilevel-free set $S \rightarrow$ IC based on $S(\hat{y})$ may not be able to cut off (x^*, y^*)

Intersection Cuts for Bilevel Optimization

- we need a **bilevel-free set** S

Theorem

For any feasible solution of the follower $\hat{y} \in \mathbb{R}^{n_2}$, the set

$$S(\hat{y}) = \{(x, y) \in \mathbb{R}^n : d^T y > d^T \hat{y},\ Ax + B\hat{y} \le b\}$$

does not contain any bilevel-feasible point (not even on its frontier).

- note: $S(\hat{y})$ is a **polyhedron**
- problem: **bilevel-infeasible** (x^*, y^*) can be on the **frontier** of bilevel-free set $S \rightarrow$ IC based on $S(\hat{y})$ may not be able to cut off (x^*, y^*)

Intersection Cuts for Bilevel Optimization

Assumption

$Ax + By - b$ is integer for all HPR solutions (x, y).

Theorem

Under the previous assumption, for any feasible solution of the follower $\hat{y} \in \mathbb{R}^{n_2}$, the extended polyhedron

$$S^+(\hat{y}) = \{(x, y) \in \mathbb{R}^n : d^T y \geq d^T \hat{y},\ Ax + B\hat{y} \leq b + \mathbf{1}\}, \tag{17}$$

where $\mathbf{1} = (1, \cdots, 1)$ denote a vector of all ones of suitable size, does not contain any bilevel feasible point in its interior.

Intersection Cuts for Bilevel Optimization

Assumption

$Ax + By - b$ is integer for all HPR solutions (x, y).

Theorem

Under the previous assumption, for any feasible solution of the follower $\hat{y} \in \mathbb{R}^{n_2}$, the extended polyhedron

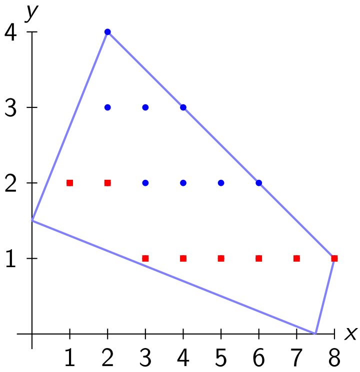
$$S^+(\hat{y}) = \{(x, y) \in \mathbb{R}^n : d^T y \geq d^T \hat{y},\ Ax + B\hat{y} \leq b + \mathbf{1}\}, \tag{17}$$

where $\mathbf{1} = (1, \cdots, 1)$ denote a vector of all ones of suitable size, does not contain any bilevel feasible point in its interior.

Intersection Cuts for Bilevel Optimization

- application sketch on the example from Moore and Bard [1990]
- solve $\overline{\text{HPR}} \rightarrow$ obtain $(x^*, y^*) = (2, 4)$ and LP-cone, take $\hat{y} = 2$
- solve $\overline{\text{HPR}}$ again $\rightarrow$ obtain $(x^*, y^*) = (6, 2)$ and LP-cone, take $\hat{y} = 1$

$$\min_{x \in \mathbb{Z}} -x - 10y$$
$$y \in \arg\min_{y' \in \mathbb{Z}} \{ y' :$$
$$-25x + 20y' \leq 30$$
$$x + 2y' \leq 10$$
$$2x - y' \leq 15$$
$$2x + 10y' \geq 15 \}$$



Intersection Cuts for Bilevel Optimization

- application sketch on the example from Moore and Bard [1990]
- solve $\overline{\text{HPR}} \rightarrow$ obtain $(x^*, y^*) = (2, 4)$ and LP-cone, take $\hat{y} = 2$
- solve $\overline{\text{HPR}}$ again $\rightarrow$ obtain $(x^*, y^*) = (6, 2)$ and LP-cone, take $\hat{y} = 1$

$$\min_{x \in \mathbb{Z}} -x - 10y$$

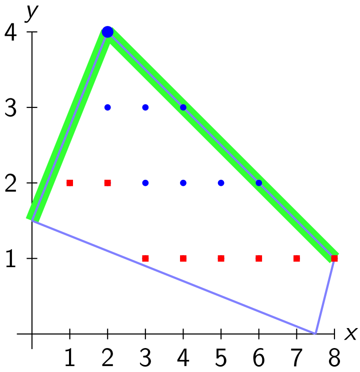
$$y \in \arg\min_{y' \in \mathbb{Z}} \{ y' :$$

$$-25x + 20y' \leq 30$$

$$x + 2y' \leq 10$$

$$2x - y' \leq 15$$

$$2x + 10y' \geq 15 \}$$



Intersection Cuts for Bilevel Optimization

- application sketch on the example from Moore and Bard [1990]
- solve $\overline{\text{HPR}} \rightarrow$ obtain $(x^*, y^*) = (2, 4)$ and LP-cone, take $\hat{y} = 2$
- solve $\overline{\text{HPR}}$ again $\rightarrow$ obtain $(x^*, y^*) = (6, 2)$ and LP-cone, take $\hat{y} = 1$

$$\min_{x \in \mathbb{Z}} -x - 10y$$

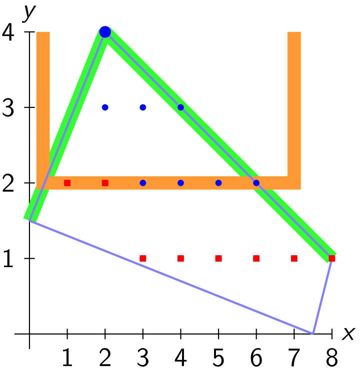
$$y \in \arg\min_{y' \in \mathbb{Z}} \{ y' :$$

$$-25x + 20y' \leq 30$$

$$x + 2y' \leq 10$$

$$2x - y' \leq 15$$

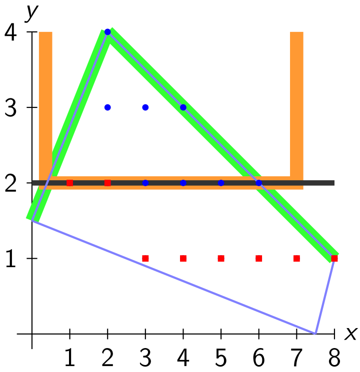
$$2x + 10y' \geq 15 \}$$



Intersection Cuts for Bilevel Optimization

- application sketch on the example from Moore and Bard [1990]
- solve $\overline{\text{HPR}} \rightarrow$ obtain $(x^*, y^*) = (2, 4)$ and LP-cone, take $\hat{y} = 2$
- solve $\overline{\text{HPR}}$ again $\rightarrow$ obtain $(x^*, y^*) = (6, 2)$ and LP-cone, take $\hat{y} = 1$

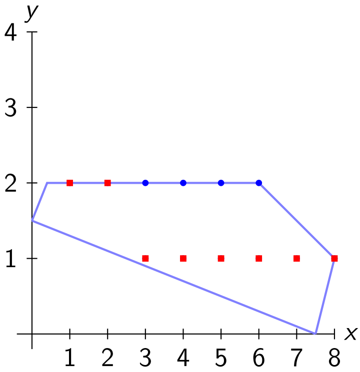
$$\min_{x \in \mathbb{Z}} -x - 10y$$
$$y \in \arg\min_{y' \in \mathbb{Z}} \{ y' :$$
$$-25x + 20y' \leq 30$$
$$x + 2y' \leq 10$$
$$2x - y' \leq 15$$
$$2x + 10y' \geq 15 \}$$



Intersection Cuts for Bilevel Optimization

- application sketch on the example from Moore and Bard [1990]
- solve $\overline{\text{HPR}} \rightarrow$ obtain $(x^*, y^*) = (2, 4)$ and LP-cone, take $\hat{y} = 2$
- solve $\overline{\text{HPR}}$ again $\rightarrow$ obtain $(x^*, y^*) = (6, 2)$ and LP-cone, take $\hat{y} = 1$

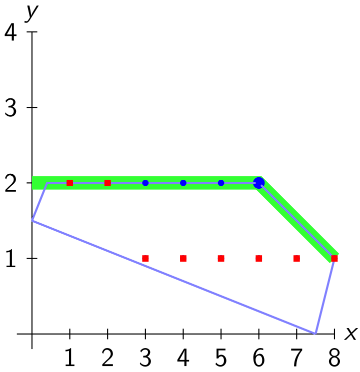
$$\min_{x \in \mathbb{Z}} -x - 10y$$
$$y \in \arg\min_{y' \in \mathbb{Z}} \{ y' :$$
$$-25x + 20y' \leq 30$$
$$x + 2y' \leq 10$$
$$2x - y' \leq 15$$
$$2x + 10y' \geq 15 \}$$



Intersection Cuts for Bilevel Optimization

- application sketch on the example from Moore and Bard [1990]
- solve $\overline{\text{HPR}} \rightarrow$ obtain $(x^*, y^*) = (2, 4)$ and LP-cone, take $\hat{y} = 2$
- solve $\overline{\text{HPR}}$ again $\rightarrow$ obtain $(x^*, y^*) = (6, 2)$ and LP-cone, take $\hat{y} = 1$

$$\min_{x \in \mathbb{Z}} -x - 10y$$
$$y \in \arg\min_{y' \in \mathbb{Z}} \{y' :$$
$$-25x + 20y' \leq 30$$
$$x + 2y' \leq 10$$
$$2x - y' \leq 15$$
$$2x + 10y' \geq 15\}$$



Intersection Cuts for Bilevel Optimization

- application sketch on the example from Moore and Bard [1990]
- solve $\overline{\text{HPR}} \rightarrow$ obtain $(x^*, y^*) = (2, 4)$ and LP-cone, take $\hat{y} = 2$
- solve $\overline{\text{HPR}}$ again $\rightarrow$ obtain $(x^*, y^*) = (6, 2)$ and LP-cone, take $\hat{y} = 1$

$$\min_{x \in \mathbb{Z}} -x - 10y$$

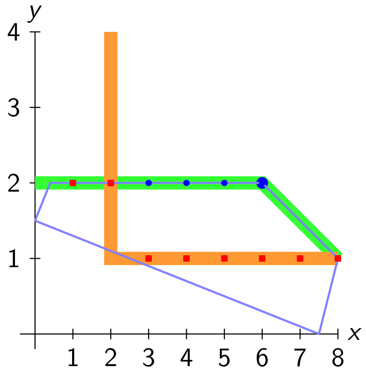
$$y \in \arg\min_{y' \in \mathbb{Z}} \{y' :$$

$$-25x + 20y' \leq 30$$

$$x + 2y' \leq 10$$

$$2x - y' \leq 15$$

$$2x + 10y' \geq 15\}$$



Intersection Cuts for Bilevel Optimization

- application sketch on the example from Moore and Bard [1990]
- solve $\overline{\text{HPR}} \rightarrow$ obtain $(x^*, y^*) = (2, 4)$ and LP-cone, take $\hat{y} = 2$
- solve $\overline{\text{HPR}}$ again $\rightarrow$ obtain $(x^*, y^*) = (6, 2)$ and LP-cone, take $\hat{y} = 1$

$$\min_{x \in \mathbb{Z}} -x - 10y$$

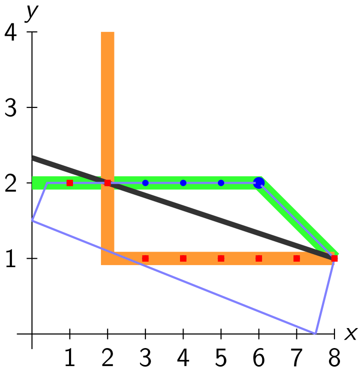
$$y \in \arg\min_{y' \in \mathbb{Z}} \{y' :$$

$$-25x + 20y' \leq 30$$

$$x + 2y' \leq 10$$

$$2x - y' \leq 15$$

$$2x + 10y' \geq 15\}$$



Separating Intersection Cuts

- given bilevel infeasible (x^*, y^*), how do we determine convex bilevel-free set $S^+(\hat{y})$?
- a natural option: use the **optimal solution** $\hat{y}$ **of the follower subproblem** for $x = x^*$
  - needs to be solved in any case to check bilevel-feasibility of (x^*, y^*)
- separation procedure is a MILP:

$$\mathbf{SEP-1}: \quad \hat{y} \in \arg\min\{d^T y \quad Ax^* + By \leq b \quad y_j \text{ integer} \quad \forall j \in J_y\}$$

SEP-1 maximizes distance of (x^*, y^*) to $d^T y \geq d^T \hat{y}$.

Separating Intersection Cuts

- given bilevel infeasible (x^*, y^*), how do we determine convex bilevel-free set $S^+(\hat{y})$?
- a natural option: use the **optimal solution $\hat{y}$ of the follower subproblem** for $x = x^*$
  - needs to be solved in any case to check bilevel-feasibility of (x^*, y^*)
- separation procedure is a MILP:

$$\textbf{SEP}-\textbf{1}: \quad \hat{y} \in \arg\min\{d^T y \quad Ax^* + By \leq b \quad y_j \text{ integer} \quad \forall j \in J_y\}$$

SEP−1 maximizes distance of (x^*, y^*) to $d^T y \geq d^T \hat{y}$.

Separating Intersection Cuts

- given bilevel infeasible (x^*, y^*), how do we determine convex bilevel-free set $S^+(\hat{y})$?
- a natural option: use the **optimal solution $\hat{y}$ of the follower subproblem** for $x = x^*$
  - needs to be solved in any case to check bilevel-feasibility of (x^*, y^*)
- separation procedure is a MILP:

$$\mathbf{SEP-1}: \quad \hat{y} \in \arg\min\{d^T y \\ Ax^* + By \leq b \\ y_j \text{ integer} \qquad \forall j \in J_y\}$$

`SEP-1` maximizes distance of (x^*, y^*) to $d^T y \geq d^T \hat{y}$.

COMPUTATIONAL RESULTS (First insights about usefulness of intersection cuts)

Computational Results

C, CPLEX 12.6.3, Intel Xeon E3-1220V2 3.1 GHz, four threads

Table: Our testbed. Column **#inst** reports the total number of instances in the class, while column **type** indicates whether the instances are binary (B) or integer (I).

Class	source	# inst	type	Notes
DENEGRE	DeNegre [2011]	50	I	randomly generated
INTERDICTION	DeNegre [2011]	125	B	interdiction inst.s
MIPLIB	Fischetti et al. [2016]	57	B	from MIPLIB 3.0

Table: Our tested settings.
$\#cuts_r/\#cuts_o$: maximum number of cuts added at root node/all other nodes

Name	Sep.	$\#cuts_r$	$\#cuts_o$
SEP-1a	SEP-1	20	20
SEP-1b	SEP-1	20	0
BENCHMARK	our benchmark code implementing cuts in DeNegre [2011]		

Computational Results

C, CPLEX 12.6.3, Intel Xeon E3-1220V2 3.1 GHz, four threads

Table: Our testbed. Column **#inst** reports the total number of instances in the class, while column **type** indicates whether the instances are binary (B) or integer (I).

Class	source	# inst	type	Notes
DENEGRE	DeNegre [2011]	50	I	randomly generated
INTERDICTION	DeNegre [2011]	125	B	interdiction inst.s
MIPLIB	Fischetti et al. [2016]	57	B	from MIPLIB 3.0

Table: Our tested settings.
$\#cuts_r/\#cuts_o$: maximum number of cuts added at root node/all other nodes

Name	Sep.	$\#cuts_r$	$\#cuts_o$
SEP-1a	SEP-1	20	20
SEP-1b	SEP-1	20	0
BENCHMARK	our benchmark code implementing cuts in DeNegre [2011]		

Computational Results

C, CPLEX 12.6.3, Intel Xeon E3-1220V2 3.1 GHz, four threads

Table: Our testbed. Column **#inst** reports the total number of instances in the class, while column **type** indicates whether the instances are binary (B) or integer (I).

Class	source	# inst	type	Notes
DENEGRE	DeNegre [2011]	50	I	randomly generated
INTERDICTION	DeNegre [2011]	125	B	interdiction inst.s
MIPLIB	Fischetti et al. [2016]	57	B	from MIPLIB 3.0

Table: Our tested settings.
$\#cuts_r/\#cuts_o$: maximum number of cuts added at root node/all other nodes

Name	Sep.	$\#cuts_r$	$\#cuts_o$
SEP-1a	SEP-1	20	20
SEP-1b	SEP-1	20	0
BENCHMARK	our benchmark code implementing cuts in DeNegre [2011]		

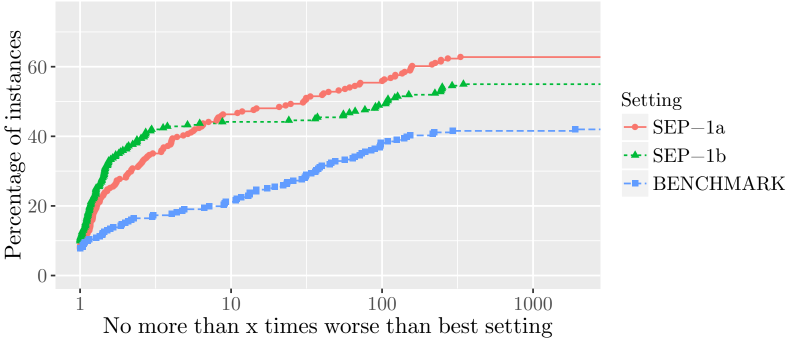
Computational Results

Table: Summary of obtained results. We report the number of solved instances (#), the shifted geometric mean for computing time ($t[s]$) and for number of nodes (*nodes*), and the average gaps ($g[\%]$).

	MIPLIB (57 inst.s)				INTERDICTION (125 inst.s)				DENEGRE (50 inst.s)			
setting	#	$t[s]$	*nodes*	$g[\%]$	#	$t[s]$	*nodes*	$g[\%]$	#	$t[s]$	*nodes*	$g[\%]$
SEP-1a	20	599	9655.9	27.65	83	148	36769.3	33.06	42	40	574.0	4.61
SEP-1b	18	660	100475.8	27.85	64	245	240859.4	48.39	45	35	12452.1	3.89
BENCHMARK	15	954	234670.7	31.78	44	496	1310639.5	63.45	38	58	27918.5	9.20

Computational Results

Figure: Performance profile plot over all instances (classes DENEGRE, INTERDICTION and MIPLIB).



The leftmost point of the graph for a setting *s* shows the percentage of instances for which *s* is the fastest setting.

The rightmost point shows the percentage of instances solved to optimality by *s*.

PART II:
MILP-BASED SOLVER for MIBLP

MILP-based solver for MIBLP

Basic Solution Scheme

- standard **simplex-based branch-and-cut** algorithm ...
- ... that enforces $d^T y \leq \Phi(x)$, on the fly, by adding cutting planes.

New features:

- Follower preprocessing.
- Follower Upper-Bound cuts.
- Intersection Cuts (ICs):
  - New families of ICs;
  - Separation of ICs.

Follower Preprocessing

$$\hat{y} \in \arg\min\{d^T y$$
$$Ax + By \leq b$$
$$l \leq y \leq u$$
$$y_j \text{ integer} \qquad \forall j \in J_y\}$$

Theorem

Let y_j be a follower variable and let l_j be its lower bound in the follower.

If *$d_j > 0$ and $B_j \geq 0$* **then** *$y_j = l_j$ in any optimal solution.*

- Idea: for any $x^* \in \mathbb{R}^{n_1}$, fixing variable y_j to the lower bound decreases the follower cost and does not reduce the associated feasible set.
- **Fix $y_j = l_j$ in the HPR as well.**
- Large impact in the performance of the algorithm.
- Observation: to preserve equivalent optimal solutions for the follower, we require d_j be *strictly* positive.

Follower Preprocessing

$$\begin{aligned} \hat{y} \in \arg\min\{d^T y & \\ Ax + By \leq b & \\ l \leq y \leq u & \\ y_j \text{ integer} & \qquad \forall j \in J_y\} \end{aligned}$$

Theorem

Let y_j be a follower variable and let u_j be its upper bound in the follower.

If *$d_j < 0$ and $B_j \leq 0$* **then** *$y_j = u_j$ in any optimal solution.*

- Idea: for any $x^* \in \mathbb{R}^{n_1}$, fixing variable y_j to the upper bound decreases the follower cost and does not reduce the associated feasible set.
- **Fix $y_j = u_j$ in the HPR as well.**
- Large impact in the performance of the algorithm.
- Observation: to preserve equivalent optimal solutions for the follower, we require d_j be *strictly* negative.

Follower Upper-Bound (FUB) cuts

Observation:

Let FUB be an upper bound for the value of the follower's solution, independently on the choice of x. Then:

$$d^T y \leq FUB$$

is a valid cut for HPR.

Tighter Bounds

Tighter FUB values could be obtained inside the B&B tree, but these cuts are only locally valid.

Overrestricting the Follower

By replacing original constraints $Ax + By \leq b$ by more restricting ones (independent on the choice of x), a FUB can be obtained.

Follower Upper-Bound cuts

Theorem

Let (x^-, x^+) denote the bounds for the x variables at the current B&B node. The following inequality

$$d^T y \leq FUB(x^-, x^+)$$

is locally valid for the current node, where

$$
\begin{aligned}
FUB(x^-, x^+) := \min\{d^T y & \\
\sum_{j \in N_x} \max\{A_{ij} x_j^-, A_{ij} x_j^+\} + \sum_{j \in N_y} B_{ij} y_j \leq b_i, & \qquad i = 1, \ldots, m \\
y_j \ \textit{integer}, & \qquad \forall j \in J_y\}.
\end{aligned}
$$

- $FUB(x^-, x^+)$ is an overestimator of the follower objective at the current node (all x's are set to their worst value).

Follower Upper-Bound cuts

Theorem

Let (x^-, x^+) denote the bounds for the x variables at the current B&B node. The following inequality

$$d^T y \leq FUB(x^-, x^+)$$

is locally valid for the current node, where

$$\begin{aligned} FUB(x^-, x^+) := \min\{d^T y & \\ \sum_{j \in N_x} \max\{A_{ij} x_j^-, A_{ij} x_j^+\} + \sum_{j \in N_y} B_{ij} y_j \leq b_i, & \qquad i = 1, \dots, m \\ y_j \textit{ integer}, & \qquad \forall j \in J_y\}. \end{aligned}$$

- $FUB(x^-, x^+)$ is an overestimator of the follower objective at the current node (all x's are set to their worst value).

MORE ON INTERSECTION CUTS

Intersection Cuts (ICs)

- Main ingredient of our basic branch-and-cut algorithm.
- Given an infeasible x^* and the associated simplex cone, the definition of an IC asks for the definition of a *convex set* S with x^* but no bilevel-feasible $x \in X$ in its *interior*.
- **The choice of bilevel-free polyhedra is not unique.**
- The larger the bilevel-free set, the better the IC.

Theorem (Fischetti et al. [2018])

Given $\hat{y} \in \mathbb{R}^n_2$ such that $\hat{y}_j$ integer $\forall j \in J_y$, the following set

$$S^+(\hat{y}) = \{(x, y) \in \mathbb{R}^n : d^T y \geq d^T \hat{y},\ Ax + B\hat{y} \leq b + \mathbf{1}\}$$

is bilevel-feasible free.

Other Bilevel-Free Sets can be defined

Motivated by the results Xu [2012], Wang and Xu [2017]:
Assumption: $Ax + By - b$ is integer for all HPR solutions (x, y).

Theorem (Fischetti et al. [2017])

Given $\Delta\hat{y} \in \mathbb{R}^n_2$ such that $d^T \Delta\hat{y} < 0$ and $\Delta\hat{y}_j$ integer $\forall j \in J_y$, the following set

$$X^+(\Delta\hat{y}) = \{(x, y) \in \mathbb{R}^n : Ax + By + B\Delta\hat{y} \leq b + \mathbf{1}\}$$

has no bilevel-feasible points in its interior.

Proof: by contradiction. Assume $(\tilde{x}, \tilde{y}) \in X^+(\Delta\hat{y})$ is bilevel-feasible. But then, $d^T\tilde{y} > d^T(\tilde{y} + \Delta\hat{y})$ and $(\tilde{x}, \tilde{y} + \Delta\hat{y})$ is feasible for the follower, hence contradiction.

SEPARATION of INTERSECTION CUTS

Separation of ICs associated to $S^+(\hat{y})$

Given $\hat{y} \in \mathbb{R}^n_2$ such that $\hat{y}_j$ integer $\forall j \in J_y$, the following set

$$S^+(\hat{y}) = \{(x, y) \in \mathbb{R}^n : d^T y \geq d^T \hat{y},\ Ax + B\hat{y} \leq b + \mathbf{1}\}$$

is bilevel-feasible free. **How to compute $\hat{y}$?**

- **SEP1**

$$\hat{y} \in \arg\min_{y \in \mathbb{R}^{n_2}} \{d^T y \ : \ By \leq b - Ax^*, \quad y_j \text{ integer } \forall j \in J_y\}.$$

  - $\hat{y}$ is the optimal solution of the follower when $x = x^*$.
  - Maximize the distance of (x^*, y^*) from the facet $d^T y \geq d^T \hat{y}$ of $S(\hat{y})$.
- **SEP2** Alternatively, try to find $\hat{y}$ such that some of the facets in $Ax + b\hat{y} \leq b$ can be removed (making thus $S(\hat{y})$ larger!)

Separation of ICs associated to $S^+(\hat{y})$

Given $\hat{y} \in \mathbb{R}_2^n$ such that $\hat{y}_j$ integer $\forall j \in J_y$, the following set

$$S^+(\hat{y}) = \{(x, y) \in \mathbb{R}^n : d^T y \geq d^T \hat{y},\ Ax + B\hat{y} \leq b + \mathbf{1}\}$$

is bilevel-feasible free. **How to compute $\hat{y}$?**

- **SEP2** (Fischetti et al. [2018])

$$\begin{aligned}
\hat{y} \in \arg\min \sum_{i=1}^{m} w_i & \\
d^T y \leq d^T y^* - 1 & \\
By + s = b & \\
s_i + (L_i^{max} - L_i^*)\, w_i \geq L_i^{max}, & \qquad \forall i = 1, \dots, m \\
y_j \text{ integer}, & \qquad \forall j \in J_y \\
s \text{ free },\ w \in \{0, 1\}^m &
\end{aligned}$$

where

$$L_i^* := \sum_{j \in N_x} A_{ij} x_j^* \ \leq\ L_i^{max} := \sum_{j \in N_x} \max\{A_{ij} x_j^-, A_{ij} x_j^+\}.$$

- $w_i = 0$ if i-th facet of $Ax + B\hat{y} \leq b$ can be removed
- the number of "removable facets" is maximized $\rightarrow$ larger $S^+(\hat{y})$.

Separation of ICs associated to $X^+(\Delta\hat{y})$

Given $\Delta\hat{y} \in \mathbb{R}_2^n$ such that $d^T\Delta\hat{y} < 0$ and $\Delta\hat{y}_j$ integer $\forall j \in J_y$, the following set

$$X^+(\Delta\hat{y}) = \{(x,y) \in \mathbb{R}^n : Ax + By + B\Delta\hat{y} \leq b + \mathbf{1}\}$$

has no bilevel-feasible points in its interior. **How to compute $\Delta\hat{y}$?**

- **XU** (Xu [2012])

$$\begin{aligned} \Delta\hat{y} \in \arg\min \sum_{i=1}^{\tilde{m}} t_i & \\ d^T\Delta y &\leq -1 \\ B\Delta y &\leq b - Ax^* - By^* \\ \Delta y_j &\text{ integer}, & \forall j \in J_y \\ B\Delta y &\leq t \text{ and } t \geq 0. \end{aligned}$$

  - variable t_i has value 0 in case $(\tilde{B}\Delta y)_i \leq 0$ ("removable facet");
  - "maximize the size" of the bilevel-feasible set associated with $\Delta\hat{y}$.

COMPUTATIONAL STUDY

Settings

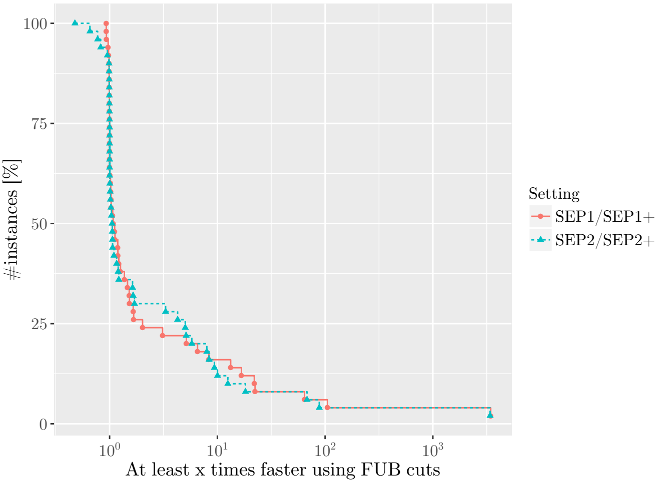
C, CPLEX 12.6.3, Intel Xeon E3-1220V2 3.1 GHz, four threads.

Class	Source	Type	#Inst	#OptB	#Opt
DENEGRE	DeNegre [2011],Ralphs and Adams [2016]	I	50	45	**50**
MIPLIB	Fischetti et al. [2016]	B	57	20	27
XUWANG	Xu and Wang [2014]	I,C	140	140	**140**
INTER-KP	DeNegre [2011],Ralphs and Adams [2016]	B	160	79	138
INTER-KP2	Tang et al. [2016]	B	150	53	**150**
INTER-ASSIG	DeNegre [2011],Ralphs and Adams [2016]	B	25	25	**25**
INTER-RANDOM	DeNegre [2011],Ralphs and Adams [2016]	B	80	-	**80**
INTER-CLIQUE	Tang et al. [2016]	B	80	10	**80**
INTER-FIRE	Baggio et al. [2016]	B	72	-	**72**
total			814	372	762

- #OptB = number of optimal solutions known before our work.
- #Opt = number of optimal solutions known after our work.

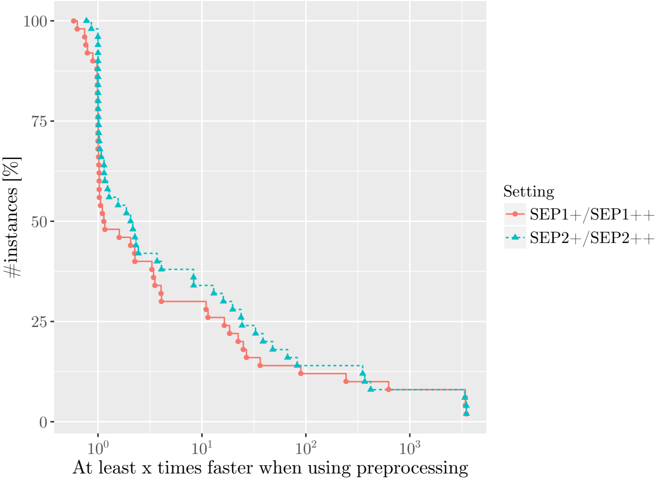
Effects of FUB cuts

- Speed-ups achieved by FUB cuts for the instance set DENEGRE.



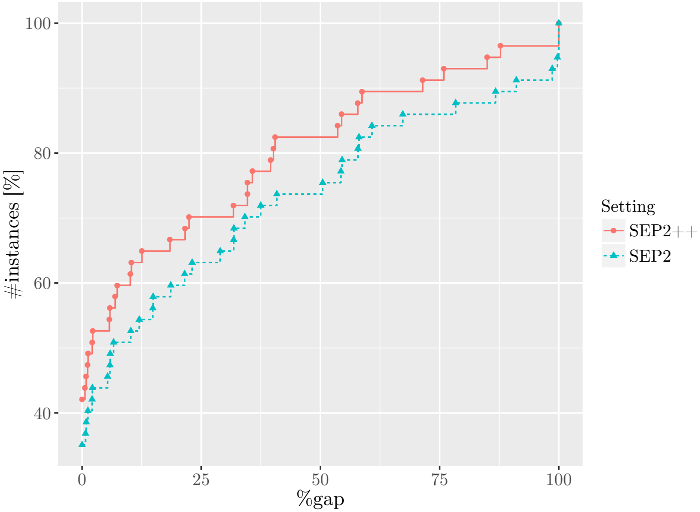
Effects of follower preprocessing

- Speed-ups achieved using follower preprocessing.



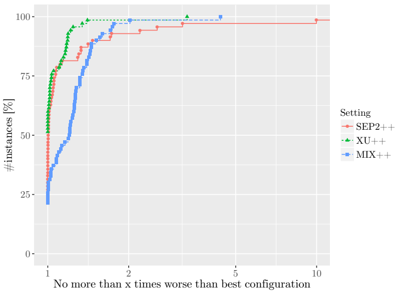
Combining FUB cuts and follower preprocessing

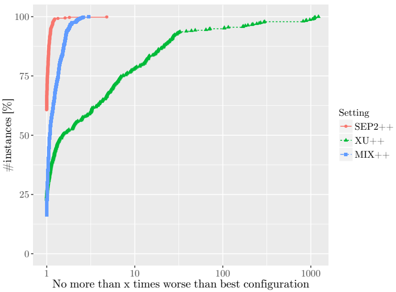
- Final gaps for settings SEP2 and SEP2++ for instance set MIPLIB, obtained when the time-limit of one hour is reached.



Effects of different ICs

- MIX++: combination of settings SEP2++ and XU++ (both ICs being separated at each separation call).
- Performance profile on the subsets of (bilevel and interdiction) instances that could be solved to optimality by all three settings within the given time-limit of one hour.





Comparison with the literature (1)

- Results for the instance set `XUWANG`

n_1	MIX++ $i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$	$i=7$	$i=8$	$i=9$	$i=10$	avg	Xu and Wang [2014] avg
10	3	3	3	3	2	3	2	3	2	3	2.6	1.4
60	2	0	0	1	1	1	1	1	2	2	0.9	45.6
110	2	1	2	2	1	2	1	2	2	12	2.8	111.9
160	2	2	3	2	3	1	4	1	1	3	2.1	177.9
210	2	3	1	1	3	3	3	2	5	3	2.6	1224.5
260	3	4	3	6	3	5	6	2	7	11	5.0	1006.7
310	5	10	11	14	7	16	15	8	5	3	9.4	4379.3
360	17	28	11	13	11	15	7	19	9	14	14.4	2972.4
410	19	10	29	8	21	10	9	15	23	42	18.7	4314.2
460	22	10	22	35	21	21	32	22	23	23	23.1	6581.4
B1-110	0	0	0	0	0	1	0	1	0	9	1.3	132.3
B1-160	1	1	3	1	2	1	3	0	0	2	1.3	184.4
B2-110	16	2	2	8	1	25	15	5	1	122	19.7	4379.8
B2-160	8	38	21	91	34	4	40	3	12	123	37.4	22999.7

Comparison with the literature (2)

- Results for the instance sets `INTER-KP2` (left) and `INTER-CLIQUE` (right)

n_1	k	MIX++ t[s]	Tang et al. [2016] t[s]	Tang et al. [2016] #unsol
20	5	5.4	721.4	0
20	10	1.7	2992.6	3
20	15	0.2	129.5	0
22	6	10.3	1281.2	6
22	11	2.3	3601.8	10
22	17	0.2	248.2	0
25	7	33.6	3601.4	10
25	13	8.0	3602.3	10
25	19	0.4	1174.6	0
28	7	97.9	3601.0	10
28	14	22.6	3602.5	10
28	21	0.5	3496.9	8
30	8	303.0	3601.0	10
30	15	31.8	3602.3	10
30	23	0.6	3604.5	10

ν	d	MIX++ t[s]	Tang et al. [2016] t[s]	Tang et al. [2016] #unsol
8	0.7	0.1	373.0	0
8	0.9	0.2	3600.0	10
10	0.7	0.3	3600.1	10
10	0.9	0.7	3600.2	10
12	0.7	0.8	3600.3	10
12	0.9	1.9	3600.4	10
15	0.7	2.2	3600.3	10
15	0.9	12.6	3600.2	10

Conclusions

- We presented an enhanced branch-and-cut algorithm, based on
  - follower preprocessing;
  - new locally-valid cuts;
  - new separation procedures for ICs.

- Detailed computational analysis (available on the paper) shows that
  - both preprocessing and FUB cuts can have a large impact on branch-and-cut performance;
  - the new algorithm outperforms previous methods from the literature (including our original branch-and-cut) by a large margin.

- Byproduct: the optimal solution for more than 300 instances previously unsolved instances from literature is now available.

Code is publicly available:

`https://msinnl.github.io/pages/bilevel.html`

Thanks for Your Attention! Questions?

Conclusions

- We presented an enhanced branch-and-cut algorithm, based on
  - follower preprocessing;
  - new locally-valid cuts;
  - new separation procedures for ICs.

- Detailed computational analysis (available on the paper) shows that
  - both preprocessing and FUB cuts can have a large impact on branch-and-cut performance;
  - the new algorithm outperforms previous methods from the literature (including our original branch-and-cut) by a large margin.

- Byproduct: the optimal solution for more than 300 instances previously unsolved instances from literature is now available.

Code is publicly available:

`https://msinnl.github.io/pages/bilevel.html`

Thanks for Your Attention! Questions?

Conclusions

- We presented an enhanced branch-and-cut algorithm, based on
  - follower preprocessing;
  - new locally-valid cuts;
  - new separation procedures for ICs.

- Detailed computational analysis (available on the paper) shows that
  - both preprocessing and FUB cuts can have a large impact on branch-and-cut performance;
  - the new algorithm outperforms previous methods from the literature (including our original branch-and-cut) by a large margin.

- Byproduct: the optimal solution for more than 300 instances previously unsolved instances from literature is now available.

Code is publicly available:

`https://msinnl.github.io/pages/bilevel.html`

Thanks for Your Attention! Questions?

Conclusions

- We presented an enhanced branch-and-cut algorithm, based on
  - follower preprocessing;
  - new locally-valid cuts;
  - new separation procedures for ICs.

- Detailed computational analysis (available on the paper) shows that
  - both preprocessing and FUB cuts can have a large impact on branch-and-cut performance;
  - the new algorithm outperforms previous methods from the literature (including our original branch-and-cut) by a large margin.

- Byproduct: the optimal solution for more than 300 instances previously unsolved instances from literature is now available.

Code is publicly available:

`https://msinnl.github.io/pages/bilevel.html`

Thanks for Your Attention! Questions?

Literature I

C. Audet, P. Hansen, B. Jaumard, and G. Savard. Links between linear bilevel and mixed 0–1 programming problems. **Journal of Optimization Theory and Applications**, 93(2):273–300, 1997.

A. Baggio, M. Carvalho, A. Lodi, and A. Tramontani. Multilevel approaches for the critical node problem. Working paper. **École Polytechnique de Montreal**, 2016.

E. Balas. Intersection cuts–a new type of cutting planes for integer programming. **Operations Research**, 19(1):19–39, 1971.

S. DeNegre. **Interdiction and Discrete Bilevel Linear Programming**. PhD thesis, Lehigh University, 2011.

M. Fischetti, I. Ljubic, M. Monaci, and M. Sinnl. Intersection cuts for bilevel optimization. In **IPCO Proceedings**, LNCS. Springer, 2016.

M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. A new general-purpose algorithm for mixed-integer bilevel linear programs. **Operations Research**, 65 (60):1615–1637, 2017.

Literature II

M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. On the use of intersection cuts for bilevel optimization. **Mathematical Programming**, 2018. doi: 10.1007/s10107-017-1189-5.

P. Hansen, B. Jaumard, and G. Savard. New branch-and-bound rules for linear bilevel programming. **SIAM Journal on Scientific and Statistical Computing**, 13(5):1194–1217, 1992.

M. Köppe, M. Queyranne, and C. T. Ryan. Parametric integer programming algorithm for bilevel mixed integer programs. **Journal of Optimization Theory and Applications**, 146(1):137–150, 2010.

A. Lodi, T. K. Ralphs, and G. J. Woeginger. Bilevel programming and the separation problem. **Math. Program.**, 146(1-2):437–458, 2014.

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

T. K. Ralphs and E. Adams. Bilevel instance library, 2016. `http://coral.ise.lehigh.edu/data-sets/bilevel-instances/`.

Y. Tang, J.-P. P. Richard, and J. C. Smith. A class of algorithms for mixed-integer bilevel min–max optimization. **Journal of Global Optimization**, 66(2): 225–262, 2016.

Literature III

L. Wang and P. Xu. The watermelon algorithm for the bilevel integer linear programming problem. **SIAM Journal on Optimization**, 27(3):1403–1430, 2017.

P. Xu. **Three essays on bilevel optimization algorithmsand applications**. PhD thesis, Iowa State University, 2012.

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.

Hypercube Intersection Cuts

- Simple polyhedron that can be used to generate IC even when $Ax + By - b$ is *NOT* integer.

Theorem

Assume $J_F := \{j \in N_x : A_j \neq 0\} \subseteq J_x$ *and let* $(\hat{x}, \hat{y})$ *an optimal bilevel-feasible solution with* $\hat{x}_j = x^*_j \; \forall j \in J_F$ *(if any). Then the following hypercube*

$$HC^+(x^*) = \{(x, y) \in \mathbb{R}^n : x^*_j - 1 \leq x_j \leq x^*_j + 1, \; \forall j \in J_F\}$$

does not contain any bilevel-feasible solution (or any bilevel-feasible solution strictly better than $(\hat{x}, \hat{y})$*, if the latter is defined) in its interior.*

- Idea: the interior of $HC^+(x^*)$ only contains bilevel-feasible solutions (x, y) with $x_j = \hat{x}_j = x^*_j \quad \forall j \in J_F$