Branch-and-Cut Solvers for Mixed-Integer Bilevel Linear Programs

Tutorial

Ivana Ljubić

ESSEC Business School of Paris, France

ALOP Autumn School on Bilevel Optimization,
Oct 12-14, 2020
online edition

Bilevel Optimization

General bilevel optimization problem

$$\min_{x \in X, y \in Y} F(x, y) \tag{1}$$

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

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

- Stackelberg game: two-person sequential game
- Leader takes follower's optimal reaction into account
- $N_x = \{1, \ldots, n_1\}$, $N_y = \{1, \ldots, n_2\}$, $n = n_1 + n_2$: total number of decision variables
- Solution y of the follower: optimal follower's response
- Sets X and Y are $\mathbb{R}^{n_1}$ and $\mathbb{R}^{n_2}$, respectively, extended with constraints regarding possible discrete variables

Bilevel Optimization

General bilevel optimization problem

$$\boxed{\begin{aligned} &\min_{x \in X, y \in Y} F(x,y) \\ &G(x,y) \le 0 \end{aligned}} \tag{1, 2}$$

Leader

$$y \in \arg\min_{y' \in Y} \{ f(x,y') : g(x,y') \le 0 \} \tag{3}$$

- 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
- Solution y of the follower: optimal follower's response
- Sets X and Y are $\mathbb{R}^{n_1}$ and $\mathbb{R}^{n_2}$, respectively, extended with constraints regarding possible discrete variables

Bilevel Optimization

General bilevel optimization problem

$$\boxed{\min_{x \in X, y \in Y} F(x,y) \qquad G(x,y) \leq 0} \tag{1, 2}$$

$$\boxed{y \in \arg\min_{y' \in Y} \{ f(x,y') : g(x,y') \leq 0 \}} \tag{3}$$

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
- Solution y of the follower: optimal follower's response
- Sets X and Y are $\mathbb{R}^{n_1}$ and $\mathbb{R}^{n_2}$, respectively, extended with constraints regarding possible discrete variables

Bilevel Optimization

General bilevel optimization problem

$$\boxed{\begin{aligned} \min_{x \in X, y \in Y} & F(x, y) \\ & G(x, y) \leq 0 \end{aligned}} \tag{1, 2}$$

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

- Stackelberg game: two-person sequential game
- Leader takes follower's optimal reaction into account
- $N_x = \{1, \ldots, n_1\}$, $N_y = \{1, \ldots, n_2\}$, $n = n_1 + n_2$: total number of decision variables
- Solution y of the follower: optimal follower's response
- Sets X and Y are $\mathbb{R}^{n_1}$ and $\mathbb{R}^{n_2}$, respectively, extended with constraints regarding possible discrete variables

Bilevel Optimization

General bilevel optimization problem

$$\boxed{\begin{aligned} \min_{x \in X, y \in Y} & F(x, y) \\ & G(x, y) \leq 0 \end{aligned}} \tag{1, 2}$$

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

- 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
- Solution y of the follower: optimal follower's response
- Sets X and Y are $\mathbb{R}^{n_1}$ and $\mathbb{R}^{n_2}$, respectively, extended with constraints regarding possible discrete variables

Value Function Reformulation

Using the **Value Function**

$$\Phi(x) = \min_{y' \in Y} \{ f(x, y') : g(x, y') \leq 0 \},$$

we can reformulate the bilevel optimization problem as

$$\min_{x \in X, y \in Y} F(x, y) \tag{4a}$$
$$G(x, y) \leq 0 \tag{4b}$$
$$g(x, y) \leq 0 \tag{4c}$$
$$f(x, y) \leq \Phi(x) \tag{4d}$$

- (4c): solution y must be feasible for the follower
- (4d): guarantees that y is the optimal follower's response for a given x
- At optimality, equality is attained in (4d)

Some Notation

Follower's Feasible Region for a given x

$$\Omega(x) = \{y : g(x, y) \leq 0, y \in Y\}.$$

Follower's Rational Reaction Set for a given x

$$R(x) = \{y : y \in \arg\min\{f(x, y) : g(x, y) \leq 0, y \in Y\}\}.$$

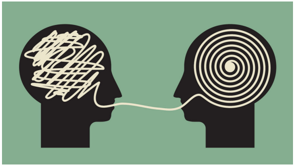
Bilevel Feasible Set (also called Inducible Region (IR))

$$IR = \{(x, y) : x \in X, G(x, y) \leq 0, y \in R(x)\}.$$

The problem is now reformulated as:

$$\min\{F(x, y) : (x, y) \in IR\}.$$

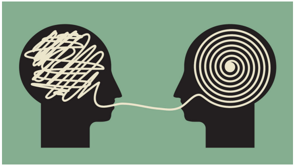
Optimistic vs Pessimistic Solution



The Stackelberg game under:

- **Perfect information:** the leader has a perfect knowledge of the follower's 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:** the leader has a perfect knowledge of the follower's 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)

$$
\begin{aligned}
\text{(MIBLP)} \quad & \min c_x^T x + c_y^T y && \text{(5a)}\\
& G_x x + G_y y \leq q && \text{(5b)}\\
& x_j \text{ integer}, \forall j \in J_x && \text{(5c)}\\
& (x, y) \in \mathbb{R}^n && \text{(5d)}\\
& y \in \arg\min\{d^T y : Ax + By \leq b, && \text{(5e)}\\
& \qquad y_j \text{ integer}, \forall j \in J_y\} && \text{(5f)}
\end{aligned}
$$

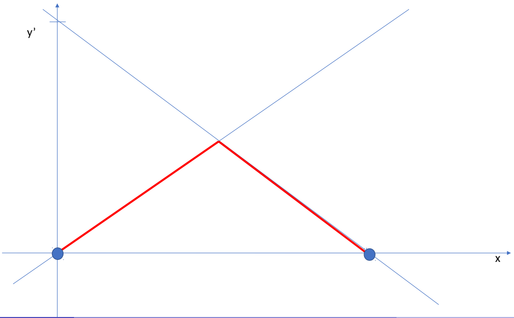
- $c_x, c_y, q, b, G_x, G_y, A, B$: given rational matrices/vectors of appropriate size
- $J_x \subseteq N_x$ and $J_y \subseteq N_y$: the index sets of variables that have to be integer
- Linking constraints: (5b)
- If $G_y = 0$: the problems can be tackled by a Benders-like decomposition (projecting out y variables)

Complexity

Bilevel Linear Programs

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

$$\begin{aligned} &\min c^T x \\ &Ax = b \\ &x \in \{0,1\} \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} &\min_{x,y} c^T x \\ &Ax = b \\ &y = 0 \\ &y \in \arg\max\{y' : y' \leq x, y' \leq 1 - x, y' \geq 0\} \end{aligned}$$



Complexity

Checking Feasibility

Given a vector (x^*, y^*), it is in general NP-complete to check whether this point is feasible for the MIBLP or not.

MIBLP

$$\begin{aligned} \min\ & c_x^T x + c_y^T y \\ & G_x x + G_y y \leq q \\ & x_j \text{ integer}, \forall j \in J_x \\ & (x, y) \in \mathbb{R}^n \\ & d^T y \leq \Phi(x) \end{aligned}$$

The Value Function constraint

Checking $d^T y^* \leq \Phi(x^*)$ requires finding the optimal solution of the MILP:

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

Complexity

Checking Feasibility

Given a vector (x^*, y^*), it is in general NP-complete to check whether this point is feasible for the MIBLP or not.

MIBLP

$$\begin{aligned} \min\ & c_x^T x + c_y^T y \\ & G_x x + G_y y \leq q \\ & x_j \text{ integer}, \forall j \in J_x \\ & (x, y) \in \mathbb{R}^n \\ & d^T y \leq \Phi(x) \end{aligned}$$

The Value Function constraint

Checking $d^T y^* \leq \Phi(x^*)$ requires finding the optimal solution of the MILP:

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

Complexity

Checking Feasibility

Given a vector (x^*, y^*), it is in general NP-complete to check whether this point is feasible for the MIBLP or not.

MIBLP

$$
\begin{aligned}
\min\ & c_x^T x + c_y^T y \\
& G_x x + G_y y \leq q \\
& x_j \text{ integer}, \forall j \in J_x \\
& (x, y) \in \mathbb{R}^n \\
& {\color{red} d^T y \leq \Phi(x)}
\end{aligned}
$$

The Value Function constraint

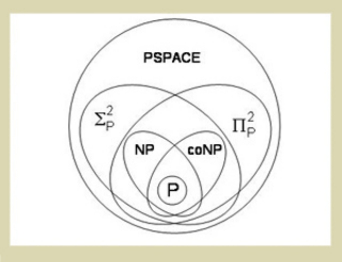
Checking $d^T y^* \leq \Phi(x^*)$ requires finding the optimal solution of the MILP:

$$
\begin{aligned}
\Phi(x^*) = \min\ \{ & d^T y \\
& By \leq b - Ax^*, \\
& y_j \text{ integer}, \forall j \in J_y \}
\end{aligned}
$$

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.



Σ_2^P-hard: the class of problems that can be solved in non-deterministic polynomial time, given an NP oracle.

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

Part III

- Branch-and-cut for MIBLPs with Interdiction Structure

PART I: VALUE FUNCTION REFORMULATION

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

Value Function Reformulation:

$$
\begin{aligned}
\text{(MIBLP)} \quad & \min c_x^T x + c_y^T y && (6)\\
& G_x x + G_y y \leq q && (7)\\
& Ax + By \leq b && (8)\\
& (x, y) \in \mathbb{R}^n && (9)\\
& d^T y \leq \Phi(x) && (10)\\
& x_j \text{ integer}, \quad \forall j \in J_x && (11)\\
& y_j \text{ integer}, \quad \forall j \in J_y && (12)
\end{aligned}
$$

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

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

- 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) 😄

$$
\begin{aligned}
\text{(HPR)} \quad \min\ & c_x^T x + c_y^T y && (6)\\
& G_x x + G_y y \le q && (7)\\
& Ax + By \le b && (8)\\
& (x, y) \in \mathbb{R}^n && (9)\\
& && (10)\\
& x_j \text{ integer}, \quad \forall j \in J_x && (11)\\
& y_j \text{ integer}, \quad \forall j \in J_y && (12)
\end{aligned}
$$

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

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

- 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:

$$(\overline{\text{HPR}}) \quad \min c_x^T x + c_y^T y \tag{6}$$

$$G_x x + G_y y \leq q \tag{7}$$

$$Ax + By \leq b \tag{8}$$

$$(x, y) \in \mathbb{R}^n \tag{9}$$

(10)

(11)

(12)

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

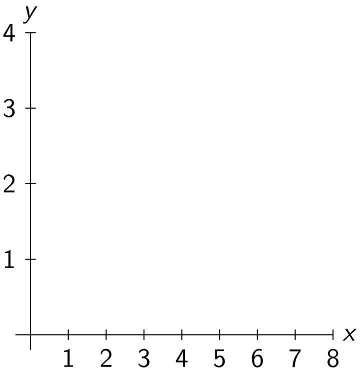
$$\Phi(x) = \min\{d^T y : By \leq b - Ax, \quad y_j \text{ integer}, \forall j \in J_y\}$$

- 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 from Moore and Bard [1990]

- HPR, $\overline{\text{HPR}}$, IR
- value-function reformulation
- Recall: for bilevel LPs, optimal solution is a vertex of the $\overline{\text{HPR}}$=HPR polytope. However, for MIBLPs, optimal solution can be in the interior of the $\mathrm{conv}(HPR)$.

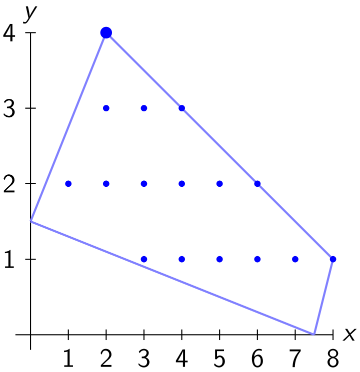
$$
\begin{aligned}
&\min_{x \in \mathbb{Z}} -x - 10y \\
&y \in \arg\min_{y' \in \mathbb{Z}} \{y' : \\
&\quad -25x + 20y' \leq 30 \\
&\quad x + 2y' \leq 10 \\
&\quad 2x - y' \leq 15 \\
&\quad 2x + 10y' \geq 15\}
\end{aligned}
$$



Example from Moore and Bard [1990]

- HPR, $\overline{\text{HPR}}$, IR
- value-function reformulation
- Recall: for bilevel LPs, optimal solution is a vertex of the $\overline{\text{HPR}}$=HPR polytope. However, for MIBLPs, optimal solution can be in the interior of the $\mathrm{conv}(HPR)$.

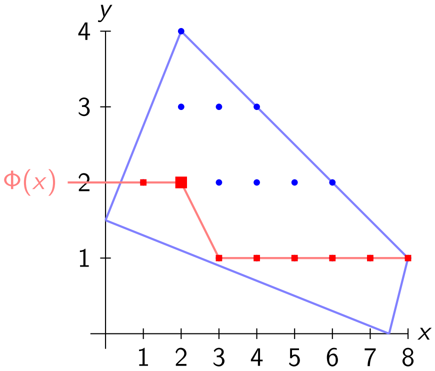
$$\begin{aligned} \min_{x,y\in\mathbb{Z}} & -x - 10y \\ -25x + 20y &\geq 30 \\ x + 2y &\leq 10 \\ 2x - y &\leq 15 \\ 2x + 10y &\geq 15 \end{aligned}$$



Example from Moore and Bard [1990]

- HPR, $\overline{\text{HPR}}$, IR
- value-function reformulation
- Recall: for bilevel LPs, optimal solution is a vertex of the $\overline{\text{HPR}}$=HPR polytope. However, for MIBLPs, optimal solution can be in the interior of the $\text{conv}(HPR)$.

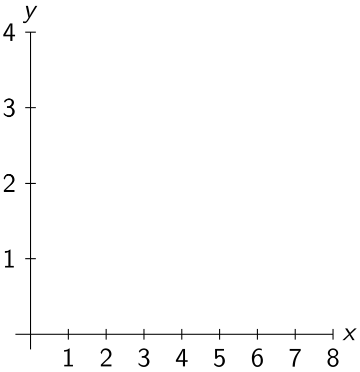
$$
\begin{aligned}
\min_{x,y\in\mathbb{Z}} & -x-10y \\
-25x+20y &\geq 30 \\
x+2y &\leq 10 \\
2x-y &\leq 15 \\
2x+10y &\geq 15 \\
y &\leq \Phi(x)
\end{aligned}
$$



(Un)expected Difficulties: **Relaxing integrality constraints**

- Relax both $x \in \mathbb{Z}$, $y' \in \mathbb{Z}$ into $(x, y') \in \mathbb{R}^2$

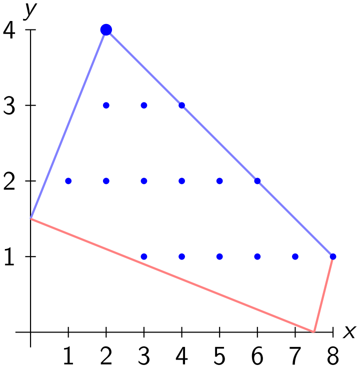
$$
\begin{aligned}
&\min_{x \in \mathbb{R}} -x - 10y \\
y \in &\arg\min_{y' \in \mathbb{R}} \{y' : \\
&\quad -25x + 20y' \le 30 \\
&\quad x + 2y' \le 10 \\
&\quad 2x - y' \le 15 \\
&\quad 2x + 10y' \ge 15\}
\end{aligned}
$$



(Un)expected Difficulties: **Relaxing integrality constraints**

- Relax both $x \in \mathbb{Z}$, $y' \in \mathbb{Z}$ into $(x, y') \in \mathbb{R}^2$

$$
\begin{aligned}
&\min_{x \in \mathbb{R}} -x - 10y \\
&y \in \arg\min_{y' \in \mathbb{R}} \{y' : \\
&\qquad -25x + 20y' \le 30 \\
&\qquad x + 2y' \le 10 \\
&\qquad 2x - y' \le 15 \\
&\qquad 2x + 10y' \ge 15\}
\end{aligned}
$$



(Un)expected Difficulties: **Relaxing integrality constraints**

- Relax both $x \in \mathbb{Z}$, $y' \in \mathbb{Z}$ into $(x, y') \in \mathbb{R}^2$

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

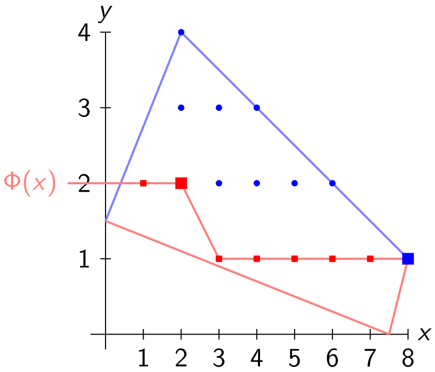
$$y \in \arg\min_{y' \in \mathbb{R}} \{y' :$$

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

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

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

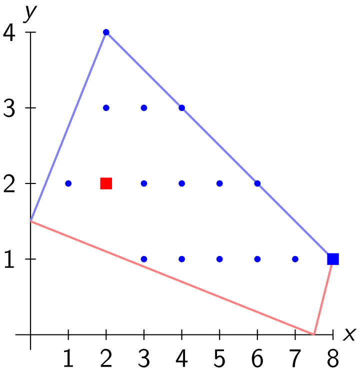
$$2x + 10y' \geq 15\}$$



(Un)expected Difficulties: **Relaxing integrality constraints**

- Relax both $x \in \mathbb{Z}$, $y' \in \mathbb{Z}$ into $(x, y') \in \mathbb{R}^2$

$$\min_{x \in \mathbb{R}} -x - 10y$$
$$y \leq \Phi_r(x)$$



- Constraints $x \in \mathbb{Z}$ are relaxed into $x \in \mathbb{R}$
- Constraints $d^T y \leq \Phi(x)$ are over-restricted into $d^T y \leq \Phi_r(x)$
- Neither valid LB nor UB can be obtained!
- IR of the LP-relaxation does not even contain the IR!
- HPR is the right way to go

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...

- HPR can be unbounded
- Optimal solution can be unattainable

(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**

Bilevel problem:

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

HPR:

$$\max_{x,y} \quad x + y$$
$$0 \le x \le 2$$
$$y \ge x$$
$$x, y \in \mathbb{Z}$$

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

(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**

Bilevel problem:

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

HPR:

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

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

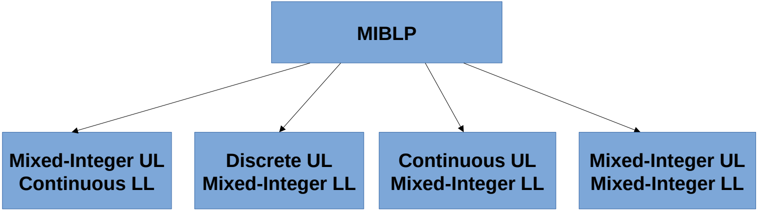
$d = 0 \quad \Rightarrow \Phi(x)$ feasible for all $y \in \mathbb{Z}$ (MIBLP unbounded)

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

Classification

Assume that variables are bounded and that HPR is feasible

Can the optimal solution always be attained?



Use standard reformulation techniques from the linear bilevel optimization

Enumeration guarantees the existence of optimal solution

Optimal solution may not be attainable if a linking variable from the upper level is continuous!

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

Example from Köppe et al. [2010]

Continuous variables in the UL, integer variables in the LL $\Rightarrow$ optimal solution may be **unattainable**

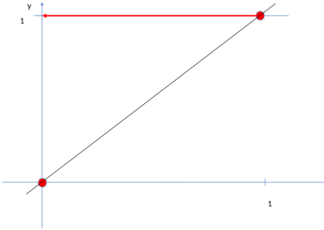
$$\inf_{x,y} \quad x - y$$

$$0 \le x \le 1$$

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

Equivalent to

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



- Bilevel feasible set is neither convex nor closed.
- See also Vicente et al. [1996].

To guarantee finite termination:

The follower subproblem depends **only on integer leader variables** $J_F \subseteq J_x$ (linking variables).

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

Example from Köppe et al. [2010]

Continuous variables in the UL, integer variables in the LL $\Rightarrow$ optimal solution may be **unattainable**

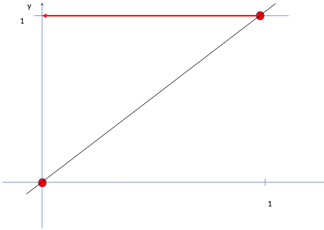
$$\inf_{x,y} \quad x - y$$

$$0 \leq x \leq 1$$

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

Equivalent to

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



- Bilevel feasible set is neither convex nor closed.
- See also Vicente et al. [1996].

To guarantee finite termination:

The follower subproblem depends **only on integer leader variables** $J_F \subseteq J_x$ (linking variables).

BRANCH-AND-BOUND

Some Basic Assumptions

Assumption 1

The $\overline{\text{HPR}}$ feasible set is a bounded polyhedron.

Assumption 2

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

Assumption 3

For any HPR solution $(x, .)$, the follower MILP is well defined and has a finite optimal solution.

If for all HPR solutions, the follower MILP is unbounded $\Rightarrow$ MIBLP is infeasible. Preprocessing (solving a single LP) allows to check this (see Theorem 1 in Fischetti et al. [2018]).

A Brute Force Approach

Recall

Continuous leader variables x_j (if any) do not appear in the follower problem. The linking variables from $J_F \subseteq J_x$ are bounded and can take only integer values.

Enumeration

Try all possible combinations of discrete values for $x_j = x_j^*$, $j \in J_F$, and choose the best from the IR. Requires solving two MILPs:

- $\Phi(x^*)$
- refined HPR

$$\min_{x \in X, y \in Y} F(x,y) \tag{13a}$$

$$G(x,y) \leq 0 \tag{13b}$$

$$g(x,y) \leq 0 \tag{13c}$$

$$f(x,y) \leq \Phi(x^*) \tag{13d}$$

$$x_j = x_j^*, j \in J_F \tag{13e}$$

General Idea

Branch-and-Bound for the Value Function Reformulation

Solve HPR and enforce $d^T y \leq \Phi(x)$ on the fly, by branching and/or cutting.

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!
  - Generate a valid inequality violated by (x^*, y^*): DeNegre [2011], Tahernejad et al. [2020], Fischetti et al. [2018, 2017]
  - Improve the approximation of the value function so that (x^*, y^*) is no longer feasible: Lozano and Smith [2017], Kleniati and Adjiman [2015]
  - Branch on a disjunction violated by (x^*, y^*): Xu [2012], Wang and Xu [2017], Xu and Wang [2014], Moore and Bard [1990], Fischetti et al. [2018]

Algorithm 1: A branch-and-bound scheme for MIBLP, Fischetti et al. [2018]

Apply a standard LP-based B&B to HPR, inhibit incumbent update ;

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

  Let (x^*, y^*) be the integer 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 and fathom the current node

  else

    if not all variables x_j **with** $j \in J_F$ **are fixed by branching then**

      Branch on any x_j ($j \in J_F$) not fixed by branching yet, even if x_j^* is integer, so as to reduce its domain in both child nodes

    else

      let $(\hat{x}, \hat{y})$ be an optimal solution of the refined HPR at the current node;

      Possibly update the incumbent with $(\hat{x}, \hat{y})$, and fathom the current node

    end

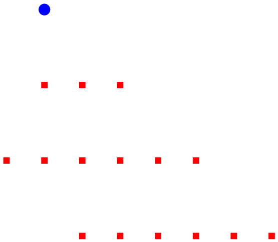
  end

end

INTERSECTION CUTS

Intersection Cuts (ICs), Balas [1971]

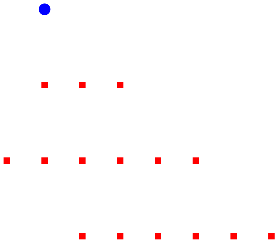
- powerful tool to separate a bilevel infeasible point (x^*, y^*) from the inducible region (IR) by a linear cut



- what we need to derive ICs
  - a cone pointed at (x^*, y^*) containing the IR (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 ($(x, y) \in IR$) in its **interior**
  - important: (x^*, y^*) should not be on the frontier of S.

Intersection Cuts (ICs), Balas [1971]

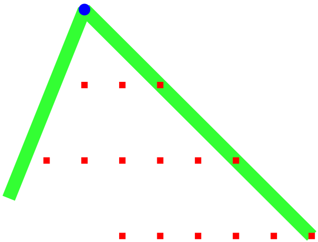
- powerful tool to separate a bilevel infeasible point (x^*, y^*) from the inducible region (IR) by a linear cut



- what we need to derive ICs
  - a cone pointed at (x^*, y^*) containing the IR (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 ($(x, y) \in IR$) in its **interior**
  - important: (x^*, y^*) should not be on the frontier of S.

Intersection Cuts (ICs), Balas [1971]

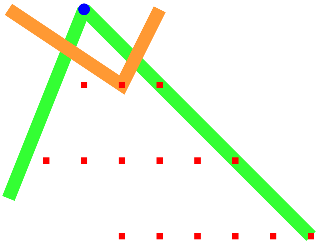
- powerful tool to separate a bilevel infeasible point (x^*, y^*) from the inducible region (IR) by a linear cut



- what we need to derive ICs
  - a cone pointed at (x^*, y^*) containing the IR (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 ($(x, y) \in IR$) in its **interior**
  - important: (x^*, y^*) should not be on the frontier of S.

Intersection Cuts (ICs), Balas [1971]

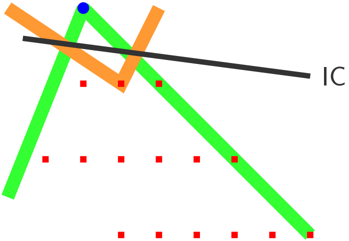
- powerful tool to separate a bilevel infeasible point (x^*, y^*) from the inducible region (IR) by a linear cut



- what we need to derive ICs
  - a cone pointed at (x^*, y^*) containing the IR (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 ($(x, y) \in IR$) in its **interior**
  - important: (x^*, y^*) should not be on the frontier of S.

Intersection Cuts (ICs), Balas [1971]

- powerful tool to separate a bilevel infeasible point (x^*, y^*) from the inducible region (IR) by a linear cut



- what we need to derive ICs
  - a cone pointed at (x^*, y^*) containing the IR (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 ($(x, y) \in IR$) 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 (Fischetti et al. [2018])

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 (Fischetti et al. [2018])

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 (Fischetti et al. [2018])

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

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

$$S^+(\hat{y}) = \{(x, y) \in \mathbb{R}^n : d^T y \geq d^T \hat{y},\ Ax + B\hat{y} \leq b + \mathbf{1}\}, \tag{14}$$

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

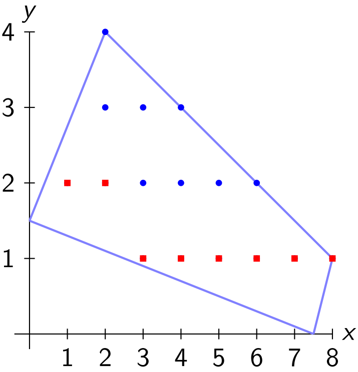
$$S^{+}(\hat{y}) = \{(x, y) \in \mathbb{R}^{n} : d^{T} y \geq d^{T} \hat{y},\ Ax + B\hat{y} \leq b + \mathbf{1}\}, \tag{14}$$

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 $\hat{y} = 1$

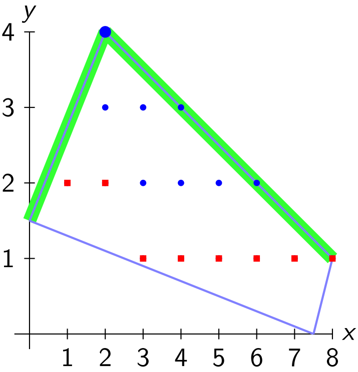
$$
\begin{aligned}
&\min_{x \in \mathbb{Z}} -x - 10y \\
&y \in \arg\min_{y' \in \mathbb{Z}} \{ y' : \\
&\qquad -25x + 20y' \leq 30 \\
&\qquad x + 2y' \leq 10 \\
&\qquad 2x - y' \leq 15 \\
&\qquad 2x + 10y' \geq 15 \}
\end{aligned}
$$



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 $\hat{y} = 1$

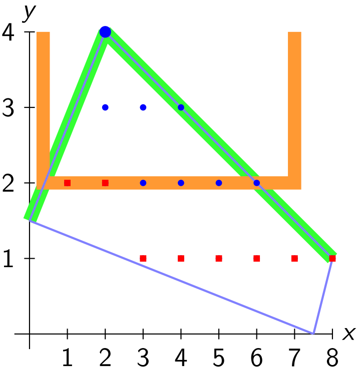
$$
\begin{aligned}
&\min_{x \in \mathbb{Z}} -x - 10y \\
&y \in \arg\min_{y' \in \mathbb{Z}} \{ y' : \\
&\qquad -25x + 20y' \leq 30 \\
&\qquad x + 2y' \leq 10 \\
&\qquad 2x - y' \leq 15 \\
&\qquad 2x + 10y' \geq 15 \}
\end{aligned}
$$



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 $\hat{y} = 1$

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

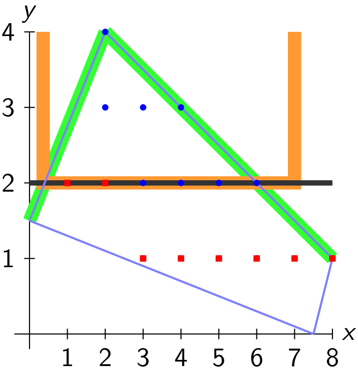
$$y \in \arg\min_{y' \in \mathbb{Z}} \{ y' :$$

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

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

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

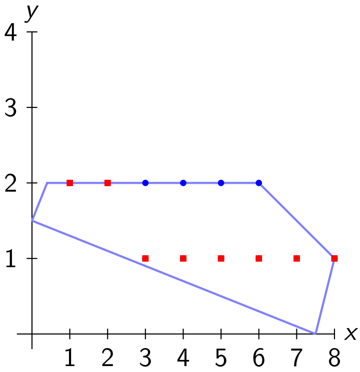
$$2x + 10y' \geq 15 \}$$



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 $\hat{y} = 1$

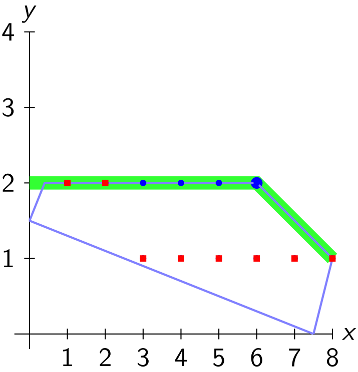
$$\min_{x \in \mathbb{Z}} -x - 10y$$
$$y \in \arg\min_{y' \in \mathbb{Z}} \{y' :$$
$$-25x + 20y' \leq 30$$
$$x + 2y' \leq 10$$
$$2x - y' \leq 15$$
$$2x + 10y' \geq 15\}$$



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 $\hat{y} = 1$

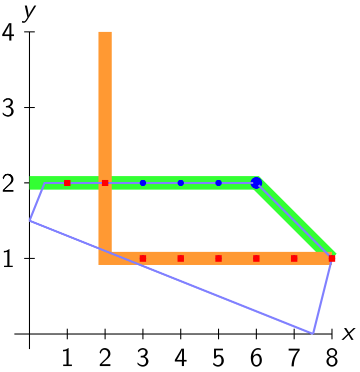
$$
\begin{aligned}
&\min_{x \in \mathbb{Z}} -x - 10y \\
&y \in \arg\min_{y' \in \mathbb{Z}} \{ y' : \\
&\qquad -25x + 20y' \leq 30 \\
&\qquad x + 2y' \leq 10 \\
&\qquad 2x - y' \leq 15 \\
&\qquad 2x + 10y' \geq 15 \}
\end{aligned}
$$



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 $\hat{y} = 1$

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

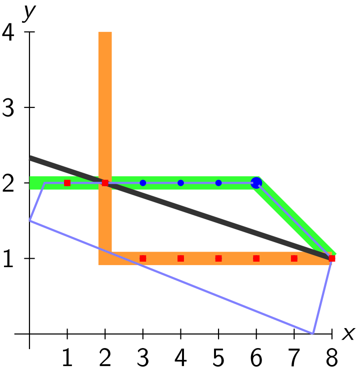
$$y \in \arg\min_{y' \in \mathbb{Z}} \{ y' :$$

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

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

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

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



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:

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

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:

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

`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]		

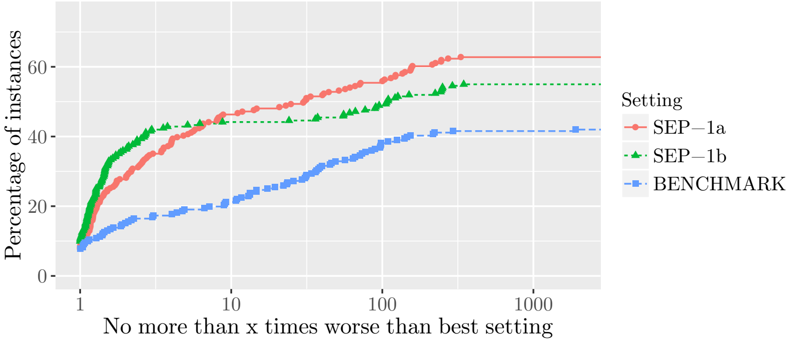
Computational Results

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.

Additional features:

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

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

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 (Xu [2012], Wang and Xu [2017], 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.

Hypercube Intersection Cuts

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

Theorem (Fischetti et al. [2017])

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 $x_j = \hat{x}_j = x_j^* \quad \forall j \in J_F$

SEPARATION of INTERSECTION CUTS

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 $\hat{y}$?**

- **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**

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

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 $X^+(\Delta\hat{y})$

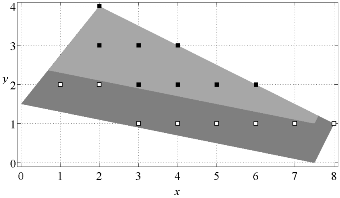
Given $\Delta\hat{y} \in \mathbb{R}^n_2$ such that $d^T\Delta\hat{y} < 0$ and $\Delta\hat{y}_j$ integer $\forall j \in J_y$, the following set

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

has no bilevel-feasible points in its interior.

To compute $\Delta\hat{y}$ (Xu [2012]):

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



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

Follower Preprocessing

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

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

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

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

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

is locally valid for the current node, where

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

- $FUB(x^-, x^+)$ 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).

COMPUTATIONAL STUDY

Settings

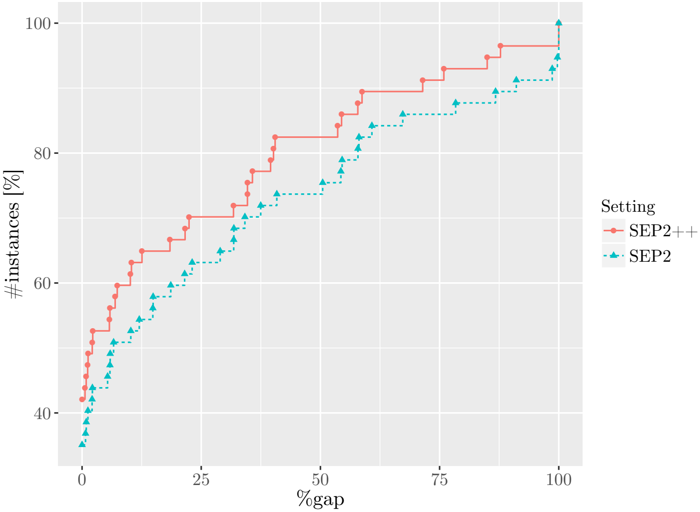
C, CPLEX 12.6.3, Intel Xeon E3-1220V2 3.1 GHz, four threads.

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.

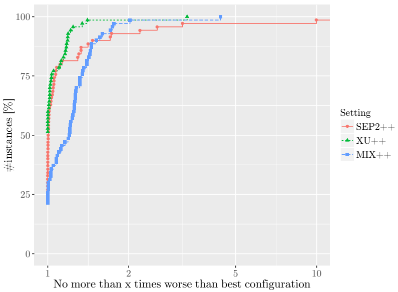
Combining FUB cuts and follower preprocessing

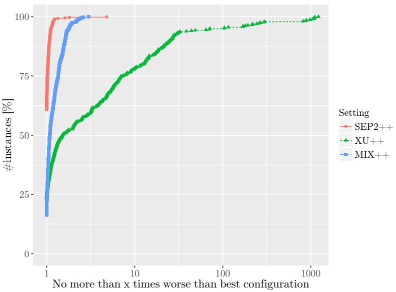
- Final gaps for settings `SEP2` and `SEP2++` for instance set `MIPLIB`, obtained when the time-limit of one hour is reached.



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` (continuous follower, with Hypercube ICs)

	MIX++											Xu and Wang [2014]
n_1	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$	$i=7$	$i=8$	$i=9$	$i=10$	avg	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)

		MIX++	Tang et al. [2016]	
n_1	k	t[s]	t[s]	#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

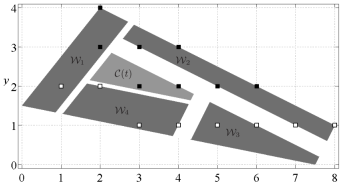
		MIX++	Tang et al. [2016]	
ν	d	t[s]	t[s]	#unsol
8	0.7	0.1	373.0	0
8	0.9	0.2	3600.0	10
10	0.7	0.3	3600.1	10
10	0.9	0.7	3600.2	10
12	0.7	0.8	3600.3	10
12	0.9	1.9	3600.4	10
15	0.7	2.2	3600.3	10
15	0.9	12.6	3600.2	10

Literature on computational MIP-based methods for solving MIBLPs

- Moore and Bard [1990]: branch-and-bound, no coupling constraints, and either the UL discrete or the LL continuous
- DeNegre [2011], DeNegre and Ralphs [2009]: branch-and-cut, no coupling constraints, cuts exploit the integrality property of UL and LL variables
- Xu and Wang [2014]: multi-way branching on the slacks of the LL
- Kleniati and Adjiman [2015]: branch-and-sandwich algorithm (bilevel MINLPs) using lower and upper bounds on $\Phi(x)$
- Lozano and Smith [2017]: discrete UL, bigM-approximation of $\Phi(x)$ using **extended HPR**
- Tahernejad et al. [2020]: branch-and-cut, no-good cuts, ICs, primal heuristics $\Rightarrow$ `MibS`, `https://coral.ise.lehigh.edu/~ted/software/#MIBS`

This talk was mainly based on

- Watermelon algorithm by Wang and Xu [2017]: multi-way branching
- PhD thesis by Xu [2012]
- Fischetti et al. [2017, 2018]: branch-and-cut with ICs, FUB cuts, preprocessing of the LL
- Binary code available: `https://msinnl.github.io/pages/bilevel.html`
- Input format: MPS (HPR) and AUX file (LL variables and OF) needed



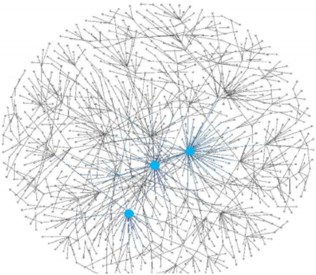
PART III: INTERDICTION PROBLEMS

Interdiction Games (IGs)

- special case of bilevel optimization problems
- leader and follower have **opposite objective functions**
- leader **interdicts** items of follower
  - type of interdiction: linear or **discrete**, cost increase or **destruction**
  - interdiction **budget**
- two-person, zero-sum sequential game
- studied mostly for network-based problems in the follower (Israeli and Wood [2002])
- but also there are interdiction problems where LL is an MILP

Interdiction Games (IGs): Attacker-Defender models

(a) Drug cartels



(b) Most voulnerable nodes

Figure: Applications of Interdiction

- Interdiction Problems: find leader's strategy that results in the worst outcome for the follower (min-max)
- Blocker Problems: find the minimum cost strategy for the leader that guarantees a limited outcome for the follower

Interdiction: Examples

APPLICATIONS: INTERDICTION

- Monitoring / halting an adversary's activity
  - Maximum-Flow Interdiction
  - Shortest-Path Interdiction
- Action:
  - Destruction of certain nodes / edges
  - Reduction of capacity / increase of cost
- The problems are NP-hard! Survey (Collado&Papp, 2012)
- Uncertainties:
  - Network characteristics
  - Follower's response

Interdiction: Examples

BILEVEL KNAPSACK WITH INTERDICTION CONSTRAINTS

$$\min_{x \in \mathbb{B}^n} \; p^T y$$

$$v^T x \leq C_l$$

where y solves the follower problem

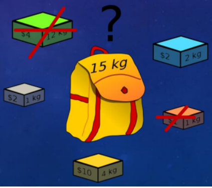
$$\max_{y \in \mathbb{B}^n} \; p^T y$$

$$w^T y \leq C_f$$

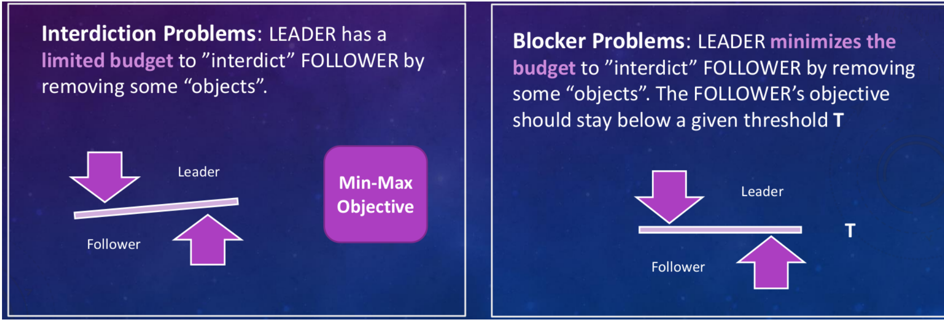
$$y_i \leq 1 - x_i \quad i = 1, \ldots, n$$

Marketing Strategy Problem (De Negre, 2011)

Companies A (leader) and B (follower).
Items are geographic regions.
Cost and benefit for each target region.
A dominates the market: whenever A and B target the same region, campaign of B is not effective



Interdiction vs Blocker Problems



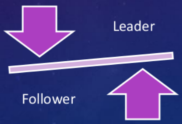
Interdiction vs Blocker Problems

VALUE FUNCTION REFORMULATION

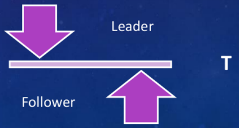
$$\min_{x \in X, w \in \mathbb{R}} \; w$$
$$w \geq \Phi(x)$$
$$v^T x \leq C_l$$
$$x_i \text{ binary}, \quad i \in N$$

$$\min_{x \in X} b^T x$$
$$T \geq \Phi(x)$$
$$x_i \text{ binary}, \quad i \in N$$

INTERDICTION: Min-max



BLOCKING: Min-num or Min-sum



Interdiction Games (IGs)

We focus on:

$$\min_{x \in X} \max_{y \in \mathbb{R}^{n_2}} \; d^T y \tag{15}$$

$$Q\, y \leq q_0 \tag{16}$$

$$0 \leq y_j \leq u_j(1 - x_j), \qquad \forall j \in N \tag{17}$$

$$y_j \text{ integer}, \qquad \forall j \in J_y \tag{18}$$

- $X = \{x \in \mathbb{R}^{n_1} : Ax \leq b,\ x_j \text{ integer } \forall j \in J_x,\ x_j \text{ binary } \forall j \in N\}$ (feasible interdiction policies).
- n_1 and n_2 are the number of leader (x) and follower (y) variables, resp.
- d, Q, q_0, u, A, b are given rational matrices/vectors of appropriate size.
- u: finite upper bounds on the follower variables y_j that can be interdicted.
- The concept easily extends to blocker problems as well.

PROBLEM REFORMULATION

Problem Reformulation

For a given $x \in X$ we define the value function:

$$\Phi(x) = \max_{y \in \mathbb{R}^{n_2}} d^T y \tag{19a}$$

$$Q\,y \leq Q_0 \tag{19b}$$

$$0 \leq y_j \leq u_j(1 - x_j), \qquad \forall j \in N \tag{19c}$$

$$y_j \text{ integer}, \qquad \forall j \in J_y \tag{19d}$$

so that problem can be restated in the $\mathbb{R}^{n_1+1}$ space as

$$\min_{x \in \mathbb{R}^{n_1}, w \in \mathbb{R}} w \tag{20a}$$

$$w \geq \Phi(x) \tag{20b}$$

$$Ax \leq b \tag{20c}$$

$$x_j \text{ integer}, \qquad \forall j \in J_x \tag{20d}$$

$$x_j \in \{0, 1\}, \qquad \forall j \in N. \tag{20e}$$

Try to replace the constraints (20b) by linear constraints.

Benders-Like Reformulation

Find (sufficiently large) M_j's and reformulate the follower Wood [2010]

$$\Phi(x) = \max\{d^T y - \sum_{j \in N} M_j x_j y_j : y \in Y\}, \tag{21}$$

where

$$Y = \{y \in \mathbb{R}^{n_2} : Q\, y \leq q_0, \quad 0 \leq y_j \leq u_j \; \forall j \in N, \quad y_j \text{ integer } \forall j \in J_y\}.$$

Let $\hat{Y}$ be extreme points of $\operatorname{conv} Y$. The, $\Phi(x)$ is a convex piecewise linear function:

$$\Phi(x) = \max_{\hat{y} \in \hat{Y}} \{d^T \hat{y} - \sum_{j \in N} M_j x_j \hat{y}_j\}.$$

Convexificaton by Penalization

Benders-Like Reformulation Wood [2010]

$$\min_{x \in \mathbb{R}^{n_1}, w \in \mathbb{R}} w \tag{22a}$$

$$w \geq d^T \hat{y} - \sum_{j \in N} M_j x_j \hat{y}_j \qquad \forall \hat{y} \in \hat{Y} \tag{22b}$$

$$Ax \leq b \tag{22c}$$

$$x_j \text{ integer,} \qquad \forall j \in J_x \tag{22d}$$

$$x_j \text{ binary,} \qquad \forall j \in N. \tag{22e}$$

INTERDICTION GAMES WITH MONOTONICITY PROPERTY

Interdiction Problems with Monotonicity Property

The follower:

$$\Phi(x) = \max_{y \in \mathbb{R}^{n_2}} \; d_N^T y_N + d_R^T y_R$$

$$Q_N\, y_N + Q_R\, y_R \leq q_0$$

$$0 \leq \; y_j \leq u_j(1 - x_j), \qquad \forall j \in N$$

$$y_j \text{ integer}, \qquad \forall j \in J_y$$

- $y_N = (y_j)_{j \in N}$ variables that can be interdicted,
- $y_R = (y_j)_{j \in R}$ the remaining follower variables.
- Associated $Q = (Q_N, Q_R)$ and $d^T = (d_N^T, d_R^T)$.

Downward Monotonicity: Assume $Q_N \geq 0$

"if $\hat{y} = (\hat{y}_N, \hat{y}_R)$ is a feasible follower for a given x and $y' = (y'_N, \hat{y}_R)$ satisfies integrality constraints and $0 \leq y'_N \leq \hat{y}_N$, then y' is **also feasible** for x".

Independence Systems (y are binary and $R = \emptyset$)

$\mathcal{S} := \{S \subseteq N : Q\,\chi_S \leq q_0\} \subseteq 2^N$ forms an **independence system** (i.e., hereditary property holds).

Interdiction Problems with Monotonicity Property

The follower:

$$\Phi(x) = \max_{y \in \mathbb{R}^{n_2}} \; d_N^T y_N + d_R^T y_R$$

$$Q_N\, y_N + Q_R\, y_R \le q_0$$

$$0 \le \; y_j \le u_j(1 - x_j), \qquad \forall j \in N$$

$$y_j \text{ integer}, \qquad \forall j \in J_y$$

- $y_N = (y_j)_{j \in N}$ variables that can be interdicted,
- $y_R = (y_j)_{j \in R}$ the remaining follower variables.
- Associated $Q = (Q_N, Q_R)$ and $d^T = (d_N^T, d_R^T)$.

Downward Monotonicity: Assume $Q_N \ge 0$

"if $\hat{y} = (\hat{y}_N, \hat{y}_R)$ is a feasible follower for a given x and $y' = (y'_N, \hat{y}_R)$ satisfies integrality constraints and $0 \le y'_N \le \hat{y}_N$, then y' is **also feasible** for x".

Independence Systems (y are binary and $R = \emptyset$)

$\mathcal{S} := \{S \subseteq N : Q\,\chi_S \le q_0\} \subseteq 2^N$ forms an **independence system** (i.e., hereditary property holds).

Even with Monotonicity the Problems Remain Hard...

Complexity

- Even when the follower is a pure LP, the problem remains NP-hard (Zenklusen [2010], Dinitz and Gupta [2013]).
- In general, already knapsack interdiction is Σ_2^P-hard (Caprara et al. [2013]).

Examples

Interdicting/Blocking:

- set packing problem
- (multidimensional) knapsack problem
- prize-collecting Steiner tree
- orienteering problem
- maximum clique problem
- all kind of hereditary problems on graphs

The Choice of M_j's is Crucial

Theorem (Fischetti et al. [2019])

For Interdiction Games with Monotonicity $M_j = d_j$, i.e., we have:

$$
\begin{aligned}
\min_{x \in \mathbb{R}^{n_1}, w \in \mathbb{R}} \quad & w \\
& w \geq \sum_{j \in R} d_j \hat{y}_j + \sum_{j \in N} d_j \hat{y}_j (1 - x_j) && \forall \hat{y} \in \hat{Y} \\
& Ax \leq b \\
& x_j \text{ integer}, && \forall j \in J_x \\
& x_j \text{ binary}, && \forall j \in N.
\end{aligned}
$$

- Branch-and-cut: separation of interdiction cuts is done by solving the follower's subproblem with given x^* (lazy cut separation).
- Important: separation of maximal solutions (monotonicity property)
- Specialized procedures/algorithms for the follower's subproblem could be exploited.

Interdiction Cuts Could be Lifted/Modified

Assumption 2

All follower variables y_N are binary and $u_j = 1$.

Theorem

Take any $\hat{y} \in \hat{Y}$. Let $a, b \in N$ with $\hat{y}_a = 1$, $\hat{y}_b = 0$, $d_a < d_b$ and $Q_a \geq Q_b$. Then the following **lifted interdiction cut** is valid:

$$w \geq \sum_{j \in R} d_j \hat{y}_j + \sum_{j \in N} d_j \hat{y}_j (1 - x_j) + (d_b - d_a)(1 - x_b).$$

Theorem

Take any $\hat{y} \in \hat{Y}$. Let $a, b \in N$ with $\hat{y}_a = 1$, $\hat{y}_b = 0$ and $Q_a \geq Q_b$. Then the following **modified interdiction cut** is valid:

$$w \geq \sum_{j \in R} d_j \hat{y}_j + \sum_{j \in N} d_j \hat{y}_j (1 - x_j) + d_b(x_a - x_b).$$

COMPUTATIONAL RESULTS KNAPSACK INTERDICTION

The Knapsack Interdiction Problem

Runtime to optimality. Our approach (B&C) vs. the cutting plane (CP) and CCLW approaches from Caprara et al. [2016].

size	instance	z^*	CP	CCLW	B&C	size	instance	z^*	CP	CCLW	B&C
35	1	279	0.34	0.79	0.12	45	1	427	1.81	2.37	0.23
	2	469	1.59	2.57	0.21		2	633	13.03	11.64	0.37
	3	448	55.61	40.39	0.66		3	548	TL	344.01	1.81
	4	370	495.50	1.48	0.87		4	611	TL	38.90	3.30
	5	467	TL	0.72	0.93		5	629	TL	3.42	2.78
	6	268	71.43	0.06	0.11		6	398	3300.76	0.07	0.17
	7	207	144.46	0.06	0.07		7	225	60.43	0.04	0.09
	8	41	0.50	0.04	0.07		8	157	60.88	0.05	0.10
	9	80	0.97	0.03	0.07		9	53	0.83	0.05	0.10
	10	31	0.12	0.03	0.08		10	110	0.40	0.05	0.11
40	1	314	0.66	1.06	0.16	50	1	502	2.86	4.55	0.21
	2	472	6.67	7.50	0.36		2	788	1529.16	1520.56	2.38
	3	637	324.61	162.80	1.02		3	631	TL	105.59	2.40
	4	388	1900.03	0.34	0.82		4	612	TL	3.64	1.27
	5	461	TL	0.22	0.58		5	764	TL	0.60	4.82
	6	399	2111.85	0.09	0.13		6	303	1046.85	0.05	0.14
	7	150	83.59	0.05	0.08		7	310	2037.01	0.09	0.11
	8	71	1.73	0.04	0.09		8	63	2.79	0.05	0.12
	9	179	137.16	0.08	0.09		9	234	564.97	0.10	0.12
	10	0	0.03	0.03	0.04		10	15	0.09	0.04	0.13

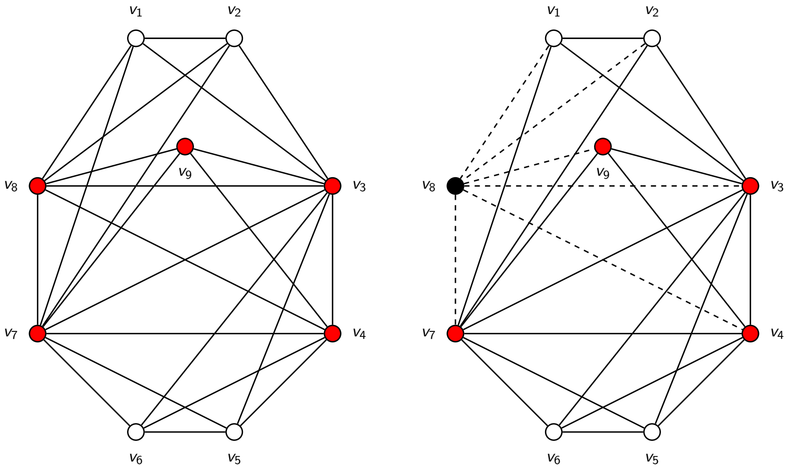
The Knapsack Interdiction Problem

Instances from Tang et al. [2016] (TRS). Comparison with MIX++. Average results over ten instances per row. N^* #instances unsolved.

$\|N\|$	k	TRS t[s]	TRS N^*	MIX++ t[s]	B&C t[s]
20	5	721.4	0	5.4	0.1
20	10	2992.6	3	1.7	0.1
20	15	129.5	0	0.2	0.1
22	6	1281.2	6	10.3	0.1
22	11	3601.8	10	2.3	0.1
22	17	248.2	0	0.2	0.1
25	7	3601.4	10	33.6	0.2
25	13	3602.3	10	8.0	0.2
25	19	1174.6	0	0.4	0.1
28	7	3601.0	10	97.9	0.3
28	14	3602.5	10	22.6	0.3
28	21	3496.9	8	0.5	0.1
30	8	3601.0	10	303.0	0.3
30	15	3602.3	10	31.8	0.3
30	23	3604.5	10	0.6	0.1

The Clique Interdiction Problem

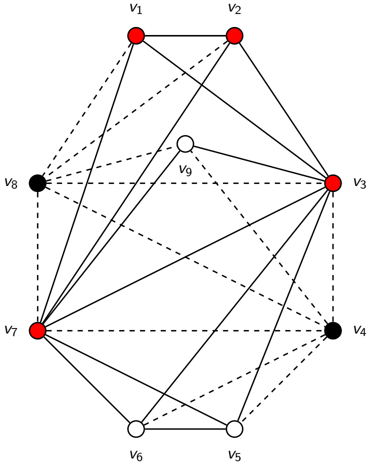
Example: $\omega(G) = 5$ and $k = 1$

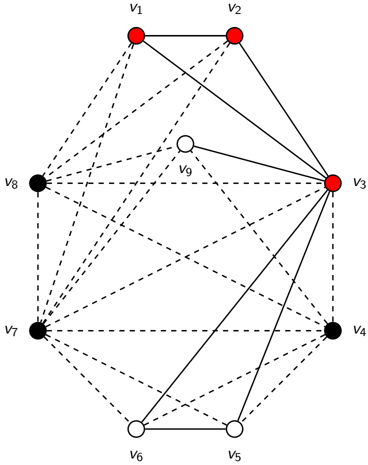


Maximum Clique $\tilde{K} = \{v_3, v_4, v_7, v_8, v_9\}$ Optimal interdiction policy $\{v_8\}$

The Clique Interdiction Problem

Example: $\omega(G) = 5$ and $k = 2, k = 3$





Optimal interdiction policy $\{v_4, v_8\}$

Optimal interdiction policy $\{v_4, v_7, v_8\}$

Branch-and-Cut for Clique Interdiction, Furini et al. [2019]

Benders-Like Reformulation

$\mathcal{K}$: set of all cliques in G.

$$\begin{aligned} \min \quad & w \\ & w + \sum_{u \in K} x_u \geq |K| \qquad K \in \mathcal{K} \\ & \sum_{u \in V} x_u \leq k \\ & x_u \in \{0,1\} \quad u \in V. \end{aligned}$$

Ingredients:

- State-of-the-art clique solver from San Segundo et al. [2016].
- Facets, lifting.
- Combinatorial primal and dual bounds.
- Graph reductions.

Comparison with MIX++

		CLIQUE-INTER				MIX++			
$\lvert V\rvert$	#	# solved	time	exit gap	root gap	# solved	time	exit gap	root gap
50	44	**44**	**0.01**	-	0.16	28	68.58	6.44	8.50
75	44	**44**	**1.45**	-	0.41	14	120.19	9.47	10.91
100	44	**37**	**9.30**	1.00	0.98	7	164.42	12.65	13.11
125	44	**35**	**13.43**	1.33	1.20	2	135.33	13.88	14.73
150	44	**33**	**27.23**	1.91	1.43	1	397.52	16.42	16.39

Results on Real-world (sparse) networks

	$\lvert V\rvert$	$\lvert E\rvert$	ω [s]	$k = \lceil 0.005 \cdot \lvert V\rvert \rceil$ [s]	$k = \lceil 0.005 \cdot \lvert V\rvert \rceil$ $\lvert V_p\rvert$	$k = \lceil 0.01 \cdot \lvert V\rvert \rceil$ [s]	$k = \lceil 0.01 \cdot \lvert V\rvert \rceil$ $\lvert V_p\rvert$
socfb-UIllinois	30,795	1,264,421	0.5	24.4	10,456	41.6	8290
ia-email-EU	32,430	54,397	0.0	0.6	30,375	0.5	29,212
rgg_n_2_15_s0	32,768	160,240	0.0	-	-	0.2	30,848
ia-enron-large	33,696	180,811	0.0	2.2	27,791	29.5	26,651
socfb-UF	35,111	1,465,654	0.3	17.8	14,264	87.8	10,708
socfb-Texas84	36,364	1,590,651	0.3	24.6	10,706	74.3	8,704
tech-internet-as	40,164	85,123	0.0	1.4	31,783	-	-
fe-body	45,087	163,734	0.1	1.8	2,259	1.8	2259
sc-nasasrb	54,870	1,311,227	0.1	-	-	145.5	1,195
soc-themarker_u	69,413	1,644,843	2.1	T.L.	35,678	T.L.	31,101
rec-eachmovie_u	74,424	1,634,743	0.7	-	-	367.3	13669
fe-tooth	78,136	452,591	0.5	18.9	7	19.0	7
sc-pkustk11	87,804	2,565,054	1.1	70.7	2,712	57.1	2,712
soc-BlogCatalog	88,784	2,093,195	11.7	T.L.	51,607	T.L.	46,240
ia-wiki-Talk	92,117	360,767	0.2	49.2	72,678	87.4	72,678

Conclusions

Branch-and-Cuts for

- General Mixed Integer Bilevel Programs (intersection cuts)
- Interdiction-Like Bilevel Programs (interdiction cuts)
- Interdiction problems easier, and it pays off to exploit the structure
- Use interdiction cuts for blocker-type problems too

Open questions, directions for future research

- Other bilevel-free sets, tighter cuts for the generic case?
- Non-linear mixed integer bilevel problems?
- General purpose solvers for bilevel pricing problems?
- Three-level and multi-level optimization problems, DAD models?

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.

A. Caprara, M. Carvalho, A. Lodi, and G. J. Woeginger. A complexity and approximability study of the bilevel knapsack problem. In **Proceedings of the 16th International Conference on Integer Programming and Combinatorial Optimization**, IPCO'13, pages 98–109. Springer, 2013.

A. Caprara, M. Carvalho, A. Lodi, and G. J. Woeginger. Bilevel knapsack with interdiction constraints. **INFORMS Journal on Computing**, 28(2):319–333, 2016.

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

Literature II

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

M. Dinitz and A. Gupta. Packing interdiction and partial covering problems. In **Proceedings of the 16th International Conference on Integer Programming and Combinatorial Optimization**, IPCO'13, pages 157–168. Springer-Verlag, 2013.

M. Fischetti, I. Ljubić, 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.

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

M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. Interdiction games under monotonicity. **INFORMS Journal on Computing**, 31(2):390–410, 2019.

Literature III

F. Furini, I. Ljubić, P. San Segundo, and S. Martin. The maximum clique interdiction problem. **European Journal of Operational Research**, 277(1): 112–127, 2019.

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.

E. Israeli and R. K. Wood. Shortest-path network interdiction. **Networks**, 40(2): 97–111, 2002.

P.-M. Kleniati and C. S. Adjiman. A generalization of the branch-and-sandwich algorithm: From continuous to mixed-integer nonlinear bilevel problems. **Computers & Chemical Engineering**, 72:373 – 386, 2015.

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.

Literature IV

L. Lozano and J. C. Smith. A value-function-based exact approach for the bilevel mixed-integer programming problem. **Operations Research**, 65(3):768–786, 2017.

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/.

P. San Segundo, A. Lopez, and P. M. Pardalos. A new exact maximum clique algorithm for large and massive sparse graphs. **Computers & OR**, 66:81–94, 2016.

S. Tahernejad, T. Ralphs, and S. DeNegre. A Branch-and-Cut Algorithm for Mixed Integer Bilevel Linear Optimization Problems and Its Implementation. **Mathematical Programming Computation (online first)**, 2020. doi: 10.1007/s12532-020-00183-6. URL http://coral.ie.lehigh.edu/~ted/files/papers/MIBLP16.pdf.

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 V

L. Vicente, G. Savard, and J. Júdice. Discrete linear bilevel programming problem. **J. Optim. Theory Appl.**, 89(3):597614, 1996.

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

R. K. Wood. **Bilevel Network Interdiction Models: Formulations and Solutions**. John Wiley & Sons, Inc., 2010.

P. Xu. **Three essays on bilevel optimization algorithms and 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.

R. Zenklusen. Matching interdiction. **Discrete Applied Mathematics**, 158(15): 1676 – 1690, 2010.