Bilevel Integer Programming

Ted Ralphs[1]

Joint work with:
Scott DeNegre[1], Menal Guzelsoy[2],
Andrea Lodi[3], Fabrizio Rossi[4], Stefano Smriglio[4]

[1]COR@L Lab, Department of Industrial and Systems Engineering, Lehigh University
[2]ISyE, Georgia Institute of Technology
[3]DEIS, Universitá di Bologna
[4]Dipartimento di Informatica, Universitá di L'Aquila

COR@L
COMPUTATIONAL OPTIMIZATION
RESEARCH AT LEHIGH

Outline

Motivation

- A standard mathematical program models a set of decisions to be made *simultaneously* by a *single* decision-maker (i.e., with a *single* objective).
- Many decision problems arising both in real-world applications and in the theory of integer programming involve
  - multiple, independent decision-makers (DMs),
  - multiple, possibly conflicting objectives, and/or
  - hierarchical/multi-stage decisions.
- Modeling frameworks
  - Multiobjective Programming $\Leftarrow$ multiple objectives, single DM
  - Mathematical Programming with Recourse $\Leftarrow$ multiple stages, single DM
  - Multilevel Programming $\Leftarrow$ multiple stages, multiple DMs
- *Multilevel programming* generalizes standard mathematical programming by modeling hierarchical decision problems, such as Stackelberg games.
- Such models arises in a remarkably wide array of applications.

Bilevel (Integer) Linear Programming

Formally, a *bilevel linear program* is described as follows.

- $x \in X \subseteq \mathbb{R}^{n_1}$ are the *upper-level variables*
- $y \in Y \subseteq \mathbb{R}^{n_2}$ are the *lower-level variables*

Bilevel (Integer) Linear Program

$$\max \left\{ c^1 x + d^1 y \mid x \in \mathcal{P}_U \cap X, y \in \operatorname{argmin}\{ d^2 y \mid y \in \mathcal{P}_L(x) \cap Y \} \right\} \quad \text{(MIBLP)}$$

The *upper-* and *lower-level feasible regions* are:

$$\mathcal{P}_U = \left\{ x \in \mathbb{R}_+ \mid A^1 x \le b^1 \right\} \text{ and}$$
$$\mathcal{P}_L(x) = \left\{ y \in \mathbb{R}_+ \mid G^2 y \ge b^2 - A^2 x \right\}.$$

For most of the talk, we assume $X = \mathbb{Z}^{n_1}$ and $Y = \mathbb{Z}^{n_2}$.

Notation

Notation

$$\Omega^I = \{(x,y) \in (X \times Y) \mid x \in \mathcal{P}_U, y \in \mathcal{S}_L(x)\}$$

$$\Omega = \{(x,y) \in (\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}) \mid x \in \mathcal{P}_U, y \in \mathcal{S}_L(x)\}$$

$$M^I(x) = \operatorname{argmin}\{d^2 y \mid y \in (\mathcal{P}_L(x) \cap Y)\}$$

$$\mathcal{F}^I = \left\{(x,y) \mid x \in (\mathcal{P}_U \cap X), y \in M^I(x)\right\}$$

$$\mathcal{F} = \left\{(x,y) \mid x \in \mathcal{P}_U, y \in \operatorname{argmin}\{d^2 y \mid y \in \mathcal{S}_L(x)\}\right\}$$

- Underlying bilevel linear program (BLP):

$$\min_{(x,y)\in\mathcal{F}} c^1 x + d^1 y$$

- Underlying MILP:

$$\min_{(x,y)\in\Omega^I} c^1 x + d^1 y$$

- Underlying LP:

$$\min_{(x,y)\in\Omega} c^1 x + d^1 y$$

Applications

- Hierarchical decision systems
  - Government agencies
  - Large corporations with multiple subsidiaries
  - Markets with a single “market-maker.”
  - Decision problems with recourse
- Parties in direct conflict
  - Zero sum games
  - Interdiction problems
- Modeling “robustness”: leader represents external phenomena that cannot be controlled.
  - Weather
  - External market conditions
- Controlling optimized systems: follower represents a system that is optimized by its nature.
  - Electrical networks
  - Biological systems

Bilevel Structure in Branch and Cut

- We consider an *integer program* (IP) of the form

$$\min\{c^\top x \mid x \in \mathcal{P} \cap \mathbb{Z}^n\}, \tag{1}$$

where $\mathcal{P} = \{x \in \mathbb{R}^n_+ \mid Ax \geq b\}$, $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$, $c \in \mathbb{Q}^n$.

- A *branch-and-cut algorithm* to solve this problem requires the solution of two fundamental decision problems.

Definition 1 *The separation problem for a polyhedron $\mathcal{Q}$ is to determine for a given $\hat{x} \in \mathbb{R}^n$ whether or not $\hat{x} \in \mathcal{Q}$ and if not, to produce an inequality $(\bar{\alpha}, \bar{\beta}) \in \mathbb{R}^{n+1}$ valid for $\mathcal{Q}$ and for which $\bar{\alpha}^\top \hat{x} < \bar{\beta}$.*

Definition 2 *The branching problem for a set $\mathcal{S}$ is to determine for a given $\hat{x} \in \mathbb{R}^n$ whether $\hat{x} \in \mathcal{S}$ and if not, to produce a disjunction*

$$\bigvee_{h \in \mathcal{Q}} A^h x \geq b^h, \; x \in \mathcal{S} \tag{2}$$

that is satisfied by all points in $\mathcal{S}$, but not satisfied by $\hat{x}$.

Bilevel Structure of the Separation Problem

- Often, we wish to select an inequality that maximizes violation, i.e.,

$$(\bar{\alpha}, \bar{\beta}) \in \operatorname{argmin}_{(\alpha,\beta)\in\mathbb{R}^{n+1}}\{\alpha^\top \hat{x} - \beta \mid \alpha^\top x \geq \beta \; \forall x \in \mathcal{Q}\} \tag{3}$$

- To make the problem tractable, we may restrict ourselves to a specific *template class* of valid inequalities with well-defined structure.
- Given a class $\mathcal{C}$, calculation of the right-hand side β required to ensure (α, β) is a member of $\mathcal{C}$ may itself be an optimization problem.
- The separation problem for the class $\mathcal{C}$ with respect to a given $\hat{x} \in \mathbb{R}^n$ can then in principle be formulated as the bilevel program:

$$\min \; \alpha^\top \hat{x} - \beta \tag{4}$$

$$\alpha \in \mathcal{C}_\alpha \tag{5}$$

$$\beta = \min\{\alpha^\top x \mid x \in \mathcal{F}\}, \tag{6}$$

where the set $\mathcal{C}_\alpha \subseteq \mathbb{R}^n$ is the projection of $\mathcal{C}$ into the space of coefficient vectors and $\mathcal{F}$ is the closure over the class $\mathcal{C}$.

Example: Disjunctive cuts

- Given a MIP in the form (1), Balas (1979) showed how to derive a valid inequality by exploiting any fixed disjunction

$$\pi^\top x \le \pi_0 \quad \text{OR} \quad \pi^\top x \ge \pi_0 + 1 \;\forall x \in \mathbb{R}^n, \tag{7}$$

  where $\pi \in \mathbb{Z}^n$ and $\pi_0 \in \mathbb{Z}$.
- A *disjunctive inequality* is one valid for the convex hull of union of $\mathcal{P}_1$ and $\mathcal{P}_2$, obtained by imposing the two terms of the disjunction.
- Conceptually, the separation problem can be written as the following bilevel program:

$$
\begin{aligned}
\min \quad & \alpha^\top \hat{x} - \beta && (8)\\
& \alpha \ge u^\top A - u_o \pi && (9)\\
& \alpha \ge v^\top A + v_o \pi && (10)\\
& u, v, u_0, v_0 \ge 0 && (11)\\
& u_0 + v_0 = 1 && (12)\\
& \beta = \min\{\alpha^\top x \mid x \in \mathcal{P}_1 \cup \mathcal{P}_2\} && (13)
\end{aligned}
$$

Example: Disjunctive Cuts (cont.d)

- Equation (13) requires β to have the largest value consistent with validity.
- To ensure the cut is valid, we need only ensure that

$$\beta \leq \min\{u^\top b - u_0\pi_0, v^\top b + v_0(\pi_0 + 1)\}. \tag{14}$$

- Using the standard modeling trick, we can rewrite (14) as

$$\beta \leq u^\top b - u_0\pi_0 \tag{15}$$
$$\beta \leq v^\top b + v_0(\pi_0 + 1). \tag{16}$$

- The sense of the optimization ensures that (14) holds at equality.

Example: Capacity Constraints for CVRP

- In the Capacitated Vehicle Routing Problem (CVRP), the *capacity constraints* are of the form

$$\sum_{\substack{e=\{i,j\}\in E \\ i\in S, j\notin S}} x_e \geq 2b(S) \quad \forall S \subset N,\ |S| > 1, \tag{17}$$

  where $b(S)$ is any lower bound on the number of vehicles required to serve customers in set S.
- By defining binary variables
  - $y_i = 1$ if customer i belongs to $\bar{S}$, and
  - $z_e = 1$ if edge e belongs to $\delta(\bar{S})$,

  we obtain the following bilevel formulation for the separation problem:

$$\min \sum_{e\in E} \hat{x}_e z_e - 2b(\bar{S}) \tag{18}$$

$$z_e \geq y_i - y_j \qquad \forall e \in E \tag{19}$$

$$z_e \geq y_j - y_i \qquad \forall e \in E \tag{20}$$

$$b(\bar{S}) = \max\{\bar{S} \mid b(\bar{S}) \text{ is a valid lower bound}\} \tag{21}$$

Example: Capacity Constraints for CVRP (cont.d)

If the bin packing problem is used in the lower-level, the formulation becomes:

$$\min \sum_{e \in E} \hat{x}_e z_e - 2b(\bar{S}) \tag{22}$$

$$z_e \geq y_i - y_j \qquad \forall e = \{i, j\} \tag{23}$$

$$z_e \geq y_j - y_i \qquad \forall e = \{i, j\} \tag{24}$$

$$b(\bar{S}) = \min \sum_{\ell=1}^{n} h_\ell \tag{25}$$

$$\sum_{\ell=1}^{n} w_i^\ell = y_i \qquad \forall i \in N \tag{26}$$

$$\sum_{i \in N} d_i w_i^\ell \leq K h_\ell \qquad \ell = 1, \ldots, n, \tag{27}$$

where we introduce the additional binary variables

- $w_i^\ell = 1$ if customer i is served by vehicle ℓ, and
- $h_\ell = 1$ if vehicle ℓ is used.

Bilevel Structure of the Branching Problem

- A typical criteria for selecting a branching disjunction is to maximize the bound increase resulting from imposing the disjunction.
- The problem of selecting the disjunction whose imposition results in the largest bound improvement has a natural *bilevel structure*.
  - The upper-level variables can be used to model the choice of disjunction (we'll see an example shortly).
  - The lower-level problem models the bound computation after the disjunction has been imposed.
- In strong branching, we are solving this problem essentially by enumeration.
- The bilevel branching paradigm is to select the branching disjunction directly by solving a bilevel program.

Example: Interdiction Branching

The following is a bilevel programming formulation for the problem of finding a smallest branching set in interdiction branching:

$$
\begin{array}{ll}
\text{(BBP)} & \max \sum c^\top x \\
\text{s.t.} & \\
& c^\top x \leq \bar{z} \\
& y \in \mathbb{B}^n \\
& x \in \arg\max_x c^\top x \\
& \text{s.t.} \\
& \qquad x_i + y_i \leq 1, \quad i \in N^a \\
& \qquad x \in \mathcal{F}^a
\end{array}
$$

where $\mathcal{F}$ is the feasible region of a given relaxation of the original problem used for computing the bound.

Algorithms: Technical Assumptions

We make the following assumptions to simplify and ensure the problem has a solution.

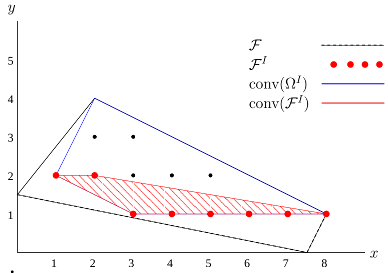
Assumptions

1. For every action by the leader, the follower has a rational reaction ($\mathcal{P}_L(x) \cap Y \neq \emptyset$ for all $x \in \mathcal{P}_U \cap X$).
2. The follower is semi-cooperative (the leader may choose among alternative members of $M^I(x)$).
3. The feasible set $\mathcal{F}^I$ is nonempty and compact.

Back to the Example

Consider again the following instance of (MIBLP) from Moore and Bard (1990).

$$\begin{aligned} \min_{x \in \mathbb{Z}} \quad & -x - 10y \\ \text{subject to} \quad & y \in \operatorname{argmin} \{ y : 25x - 20y \geq -30 \\ & \qquad -x - 2y \geq -10 \\ & \qquad -2x + y \geq -15 \\ & \qquad 2x + 10y \geq 15 \\ & \qquad y \in \mathbb{Z} \ \} \end{aligned}$$



From the figure, we can make several observations:

1. $\mathcal{F} \subseteq \Omega$, $\mathcal{F}^I \subseteq \Omega^I$, and $\Omega^I \in \Omega$
2. $F^I \not\subseteq \mathcal{F}$
3. Solutions to (MIBLP) do not occur at extreme points of $\text{conv}(\Omega^I)$

Properties of MIBLPs

In this example:

- Optimizing over $\mathcal{F}$ yields the *integer* solution $(8, 1)$, with the upper-level objective value 18.
- Imposing integrality yields the solution $(2, 2)$, with upper-level objective value 22

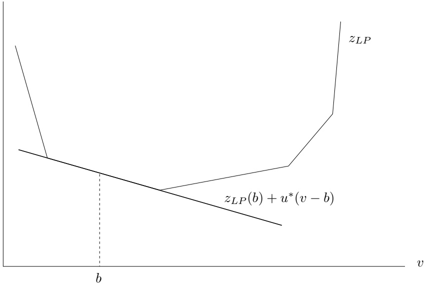
From this we can make two important observations:

1. The objective value obtained by relaxing integrality is not a valid bound on the solution value of the original problem,
2. Even when solutions to $\max_{(x,y)\in\mathcal{F}} c^1x + d^1y$ are in $\mathcal{F}^I$, they are not necessarily optimal.

Thus, some familiar properties from the MILP case do not hold here.

Special Case: Recourse Problems

- If $d^1 = -d^2$, we can view this as a *mathematical program with recourse*.
- Two-stage stochastic programs with recourse are a special case (under mild conditions).
- The resulting problem can be solved as a standard mathematical program.
- For the case when $Y = \mathbb{R}^n$, we can solve the problem by Benders Decomposition.
- Note that the value function of the lower-level problem is convex in the upper-level variables, so we can also reformulate as a convex program
- This is a useful way of visualizing the situation.



Special Case: Continuous BLPs

- In the continuous case, the lower-level problem can be replaced with its optimality conditions.
- The optimality conditions for the lower-level optimization problem are

$$
\begin{aligned}
G^2 y &\geq b^2 - A^2 x \\
u G^2 &\leq d^2 \\
u(b^2 - G^2 - A^2 x) &= 0 \\
(d^2 - u G^2) y &= 0 \\
u, y &\in \mathbb{R}_+
\end{aligned}
$$

- Note that this is a special case of a class of non-linear mathematical programs known as *mathematical programs with equilibrium constraints* (MPECs).
- This can be solved in a number of ways, including converting it to standard integer program.
- Note that in this case, the value function of the lower-level problem is piecewise linear, but not necessarily convex.

General Case: Discrete BLPs

- When some/all of the variables are discrete, the situation is a bit more difficult.
- Because the duals that exist for general integer programs are not tractable in general, we cannot use the same approach as we did for the continuous case.
- In fact, going from the continuous case to the discrete case in the bilevel setting poses significantly different challenges than for standard MILPs.
- Nevertheless, we have developed a *branch-and-cut algorithm* that attempts to generalize techniques from MILP.

Lower Bounds

Relaxing integrality conditions *and* the requirement $y \in M^I(x)$ yields the relaxation

$$\max_{(x,y)\in\Omega} c^1x + d^1y. \qquad \text{(LR)}$$

- The resulting bound can be used in combination with a standard variable branching scheme to yield an algorithm that solves (MIBLP).
- Unfortunately, the bound is too weak to be effective on interesting problems.
- As usual, we strengthen the linear relaxation by exploiting disjunctions valid for the bilevel feasible region.

Valid Disjunctions for MIBLP

Bilevel Feasibility Conditions

1. $(x, y) \in \Omega$,
2. $(x, y) \in X \times Y$, and
3. $y \in M^I(x)$.

- To develop a successful branch-and-cut algorithm, we would like to derive disjunctions arising from violation of these conditions.
- Violations of Conditions 1 and 2 can be dealt with as in the MILP case.
- Violations of Condition 3 are both difficult to detect in general and difficult to exploit.

Bilevel Feasibility Check

- Let $(\hat{x}, \hat{y})$ be a solution to LR.
- We fix $x = \hat{x}$ and solve the lower-level problem

$$\min_{y \in \mathcal{P}_L^I(\hat{x})} d^2 y \tag{28}$$

with the fixed upper-level solution $\hat{x}$.

- Let y^* be the solution to (28).
  - $(\hat{x}, y^*)$ is bilevel feasible $\Rightarrow c^1\hat{x} + d^1 y^*$ is a valid upper bound on the optimal value of the original MIBLP
  - Either
    1. $d^2\hat{y} = d^2 y^* \Rightarrow (\hat{x}, \hat{y})$ is bilevel feasible.
    2. $d^2\hat{y} > d^2 y^* \Rightarrow$ generate a valid inequality violated by $(\hat{x}, \hat{y})$.

Bilevel Feasibility Cut (Pure Integer Case)

Let

$$A := \begin{bmatrix} A^1 \\ A^2 \end{bmatrix}, \quad G := \begin{bmatrix} 0 \\ G^2 \end{bmatrix}, \quad \text{and} \quad b := \begin{bmatrix} b^1 \\ b^2 \end{bmatrix}.$$

A basic feasible solution $(\hat{x}, \hat{y}) \in \Omega^I$ to (LR) is the *unique* solution to

$$a_i' x + g_i' y = b_i, \quad i \in I$$

where I is the set of active constraints at $(\hat{x}, \hat{y})$.

This implies that

$$\left\{ (x, y) \in \Omega^I \mid \sum_{i \in I} a_i' x + g_i' y = \sum_{i \in I} b_i \right\} = \left\{ (\hat{x}, \hat{y}) \right\}$$

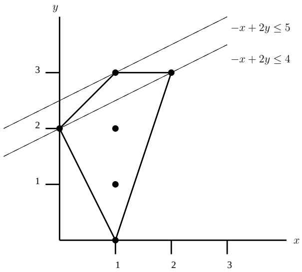
and $\sum_{i \in I} a_i' x + g_i' y \leq \sum_{i \in I} b_i$ is valid for Ω.

Bilevel Feasibility Cut (cont.)

A Valid Inequality

$\sum_{i \in I} a_i' x + g_i' y \leq \sum_{i \in I} b_i - 1$ is valid for $\Omega^I \setminus \{(x, y)\}$.

$$\max_x \min_y \{y \mid -x + y \leq 2, -2x - y \leq -2, 3x - y \leq 3, y \leq 3, x, y \in \mathbb{Z}_+\}.$$



This yields a finite algorithm in the pure integer case.

Exploiting the Value Function

The value function of an MILP is a function $z : \mathbb{R} \to \mathbb{R} \cup \{\pm\infty\}$

MILP Value Function

$$z(d) = \min_{x \in S(d)} cx, \tag{29}$$

where, for a given right-hand side vector $d \in \mathbb{R}^m$,

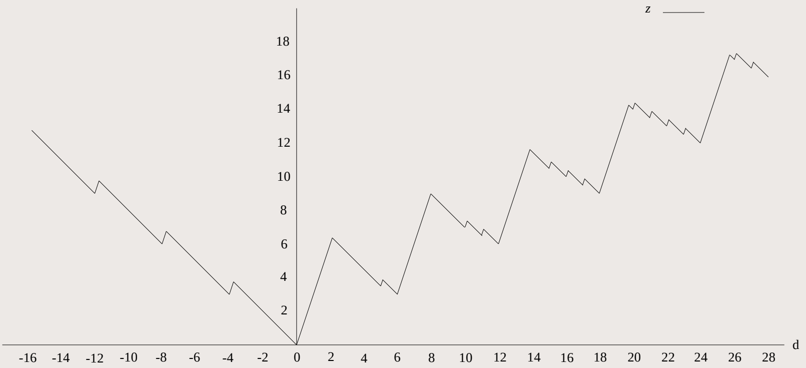
$$S(d) = \{x \in \mathbb{Z}^p_+ \times \mathbb{R}^{n-p}_+ \mid Ax \leq d\}.$$

If we knew the value function, we could reformulate as follows:

$$\begin{aligned} \min \quad & c^1x + d^1y \\ \text{subject to} \quad & A^1x \geq b^1 \\ & A^2x + G^2y \geq b^2 \\ & d^2y = z(b^2 - A^2x) \\ & x \in X, y \in Y. \end{aligned}$$

Example

$$\begin{array}{rl} \min & 3x_1 + \frac{7}{2}x_2 + 3x_3 + 6x_4 + 7x_5 + 5x_6 \\ \text{s.t} & 6x_1 + 5x_2 - 4x_3 + 2x_4 - 7x_5 + x_6 = b \quad \text{and} \\ & x_1, x_2, x_3 \in \mathbb{Z}_+, x_4, x_5, x_6 \in \mathbb{R}_+. \end{array} \qquad \text{(SP)}$$

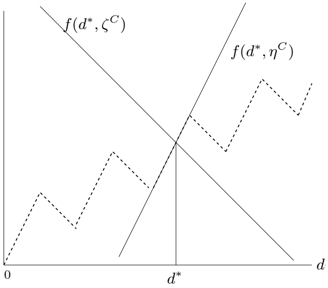


Bounding the Value Function

To generate valid disjunctions violated by solutions not satisfying Condition 3, we must somehow bound the value function.

- Upper bounds can be derived by considering the value function of *restrictions* of the original problem.
  $\Rightarrow$ Fix some integer variables.
- Lower bounds can be derived by considering the value function of *relaxations* of the original problem.
  $\Rightarrow$ Relax integrality of some variables.
- Lower bounds can also be obtained by considering so-called *dual functions* that can be constructed in a number of ways.

Upper Bound (Single Constraint Case)



For any $d \leq d^*$,

$$z(d) \leq \max\{f(d^*, \zeta^C), f(d^*, \eta^C)\} = f(d^*, \zeta^C).$$

Similarly, for any $d \geq d^*$,

$$z(d) \leq \max\{f(d^*, \zeta^C), f(d^*, \eta^C)\} = f(d^*, \eta^C).$$

Bilevel Feasibility Disjunctions (Single Constraint Case)

Thus, we have the following disjunction.

Bilevel Feasibility Disjunction

$$b^2 - A^2x \leq b^2 - A^2\hat{x} \quad \text{AND} \quad d^2y \leq f(b^2 - A^2\hat{x}, \zeta^C)$$

$$\text{OR}$$

$$b^2 - A^2x \geq b^2 - A^2\hat{x} \quad \text{AND} \quad d^2y \leq f(b^2 - A^2\hat{x}, \eta^C).$$

Such a disjunction can be used to either branch or cut when solutions $(\hat{x}, \hat{y}) \in \Omega^I$ such that $\hat{y} \notin M^I(\hat{x})$ are found.

Numerical Example (Disjunctive Cut)

MIBLP Example

$$
\begin{aligned}
\min \quad & 8x_1 + x_2 + 2x_3 + 3x_5 + 4x_6 \\
& + y_1 + 3y_2 + 2y_3 + 4y_4 + 2y_6 \\
\text{subject to} \quad & 4x_1 + 2x_2 - 4x_3 + 3x_4 + 6x_5 + x_6 = 24 \\
& x_1, x_2, x_3 \in \mathbb{Z}_+, x_4, x_5, x_6 \in \mathbb{R}_+ \\
& y \in \operatorname{argmin} \{ \, 2y_1 + 2y_3 + 8y_4 + 4y_5 + 3y_6 : \\
& \qquad\qquad 2x_1 - x_2 + 4x_3 - 4x_4 + 3x_5 + x_6 \\
& \qquad\qquad\quad + 4y_1 - 6y_2 + 6y_3 + 4y_4 - 4y_5 + \frac{4}{3}y_6 = 16, \\
& \qquad\qquad y_1, y_2, y_3 \in \mathbb{Z}_+, y_3, y_5, y_6 \in \mathbb{R}_+ \, \},
\end{aligned}
$$

Numerical Example (cont.)

- The initial LP lower bound is 3, with a solution of

$$x_4 = 8, \quad y_1 = 12.$$

- The cutting plane procedure yields the cut

$$\frac{3}{4}x_1 + \frac{7}{24}x_2 + \frac{1}{3}x_4 + \frac{9}{8}x_5 + \frac{5}{24}x_6 - \frac{1}{3}y_1 - \frac{1}{6}y_2 - \frac{1}{8}y_3 - \frac{1}{12}y_4 - \frac{1}{12}y_g \geq \frac{10}{3}.$$

- This yields a new lower bound of 3.224 and a solution of

$$x_4 = 0.8163, \quad x_5 = 3.5918, \quad y_1 = 2.1225.$$

- The optimal value is 3.25 and an optimal solution is

$$x_5 = 4, \quad y_1 = 1.$$

Jeroslow Formula for General MILP

Let the set $\mathbb{E}$ consist of the index sets of dual feasible bases of the linear program

$$\min\{\frac{1}{M}c_C x_C \ : \ \frac{1}{M}A_C x_C = b, x \geq 0\}$$

where $M \in \mathbb{Z}_+$ such that for any $E \in \mathbb{E}$, $MA_E^{-1}a^j \in \mathbb{Z}^m$ for all $j \in I$.

Theorem 1 (Jeroslow Formula) *There is a $g \in \mathscr{G}^m$ such that*

$$z(d) = \min_{E \in \mathbb{E}} g(\lfloor d \rfloor_E) + v_E(d - \lfloor d \rfloor_E) \ \ \forall d \in \mathbb{R}^m \ \ \textit{with} \ \ \mathcal{S}(d) \neq \emptyset,$$

where for $E \in \mathbb{E}$, $\lfloor d \rfloor_E = A_E \lfloor A_E^{-1} d \rfloor$ and v_E is the corresponding basic feasible solution.

Generalizing

- The question of how to derive practical disjunctions based on local structure in the general case is still largely unanswered.
- The Jeroslow formula (among others) tells us what the local structure looks like.
- We can derive local structure from the branch-and-bound tree constructed when we do the bilevel feasibility check.
- There is an obvious combinatorial explosion.

Implementation

The Mixed Integer Bilevel Solver (MibS) implements the branch and bound framework described here using software available from the Computational Infrastructure for Operations Research (COIN-OR) repository.

COIN-OR Components Used

- The COIN High Performance Parallel Search (CHiPPS) framework to perform the branch and bound.
- The COIN Branch and Cut (CBC) framework for solving the MILPs.
- The COIN LP Solver (CLP) framework for solving the LPs arising in the branch and cut.
- The Cut Generation Library (CGL) for generating cutting planes within CBC.
- The Open Solver Interface (OSI) for interfacing with CBC and CLP.

What Is Implemented

MibS is still in its infancy and is not fully general. Currently, we have:

- Bilevel feasibility cuts (pure integer case).
- Specialized methods (primarily cuts) for pure binary at the upper level.
- Specialized methods for interdiction problems.
- Disjunctive cuts based on the value function for lower-level problems with a single constraint.
- Several primal heuristics.
- Simple preprocessing.

Preliminary Results from Knapsack Interdiction

	Maximum Infeasibility			Strong Branching		
$2n$	Avg Nodes	Avg Depth	Avg CPU (s)	Avg Nodes	Avg Depth	Avg CPU (s)
20	359.30	8.65	9.32	358.30	8.65	11.07
22	658.40	9.85	18.50	658.20	9.85	18.92
24	1414.80	10.85	46.03	1410.80	10.75	46.46
26	2725.00	12.05	97.55	2723.50	12.05	100.17
28	5326.40	12.90	214.97	5328.60	12.95	220.26
30	10625.00	14.05	482.70	10638.00	14.10	538.32

- Interdiction problems in which the lower-level problems are binary knapsack problems with a single constraint.
- Data was taken from the *Multiple Criteria Decision Making* library and modified to suit our setting.
- Results for each problem size reflect the average of 20 instances.
- These instances were running using the interdiction customization.

Preliminary Results from Assignment Interdiction

Instance	Nodes	Depth	CPU (s)
2AP05-1	6203	33	290.25
2AP05-2	3881	32	384.97
2AP05-3	3909	32	205.93
2AP05-4	2441	36	102.66
2AP05-5	3505	33	119.18
2AP05-6	2031	35	80.31
2AP05-7	2957	29	153.02
2AP05-8	3549	32	224.77
2AP05-9	2271	33	111.13
2AP05-10	3299	31	211.07
2AP05-11	707	33	35.13
2AP05-12	407	18	29.51
2AP05-13	391	18	23.80
2AP05-14	3173	28	261.08
2AP05-15	2509	32	127.05
2AP05-16	1699	29	44.61
2AP05-17	5417	29	201.34
2AP05-18	5785	32	176.67
2AP05-19	2259	32	79.70
2AP05-20	2585	31	77.35
2AP05-21	6039	33	161.44
2AP05-22	2479	29	48.06
2AP05-23	1519	25	49.40
2AP05-24	15	5	1.32
2AP05-25	3857	31	115.97

- Here, the lower-level problems are binary assignment problems.
- Data also taken from *Multiple Criteria Decision Making* library.
- Problems have 50 variables and 45 constraints.

Conclusions and Future Work

- Preliminary testing to date has revealed that these problems can be extremely difficult to solve in practice.
- What we have implemented so far has only scratched the surface.
- Currently, we are focusing on special cases where we can get traction.
  - Interdiction problems
  - Stochastic integer programs
- Much work remains to be done.
- Please join us!

References I

Balas, E. 1979. Disjunctive programming. *Annals of Discrete Mathematics* **5**, 3–51.

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