Multiobjective Mixed-Integer Stackelberg Games

SCOTT DENEGRE
TED RALPHS

ISE Department
COR@L Lab
Lehigh University
tkralphs@lehigh.edu

EURO XXI, Reykjavic, Iceland

July 3, 2006

Outline

1. Stackelberg Games
   - General Setup
   - Zero-sum Stackelberg Games
   - Mixed-Integer Stackelberg Games
2. Biobjective Programs
   - Overview
   - Solution Techniques
3. Solving the LBMZSSPIC
   - Using the WCN Algorithm
   - Solving the Subproblems
   - A Branch and Bound Algorithm
4. Conclusion

Basic Setup

- We wish to study the following type of two-player game:

Scenario

- There are two players, a *leader* and a *follower*.
- Each player controls the level of certain *activities* subject to *constraints* and seeks to optimize a given objective function.
- The leader sets activity levels first.
- The follower then reacts based on the choices of the leader.
- The leader can affect the follower through constraints on the follower that involve the leader's choices.

- This is an example of what are generally known in the literature as *Stackelberg games*.
- Here, we consider *static* games in which there is only one round of decision-making.
- We can model such games mathematically as *bilevel programs*.

Bilevel Programs

- Suppose the variables x and y represent the activities controlled by the leader and follower, respectively.
- A *bilevel program* is a standard mathematical program with y constrained to be optimal for the secondary program

$$\min \{f(x,y) : h(x,y) \leq 0\}.$$

This yields the (continuous) bilevel programming problem (BPP):

A Bilevel Program

$$\begin{aligned} \min_{x,y} \quad & F(x,y) \\ \text{subject to} \quad & g(x,y) \leq 0 \\ & y \in \operatorname{argmin} \{f(x,y) : h(x,y) \leq 0\} \end{aligned} \tag{1}$$

- Bilevel programs are known to be $\mathcal{NP}$-hard, even if all functions are linear [Calamai and Vincente, 1994, Jeroslow, 1985, Ben-Ayed and Blair, 1990, Hansen et al., 1992].

Underlying Assumptions

- Implicit in formulation (1) are the following basic assumptions:

Assumptions

1. Perfect information: Each player has perfect information about the other's actions and strategy.
2. Rationality: Both players act optimally.
3. Determinism: Each player chooses deterministically among alternative optima.

- Assumption 3 is necessary in order for (1) to be well-defined.
- The presence of this assumption means that the real challenge is computing the leader's decision.
- The philosophy is similar to that of solving the first stage of a stochastic integer program.

Zero-sum Stackelberg Games

If the leader and follower are in direct opposition, then

$$F(x,y) = -f(x,y)$$

and we get the *zero-sum static Stackelberg game* (ZSSP):

ZSSP

$$\begin{aligned} \min_{x} \quad & f(x,y) \\ \text{subject to} \quad & g(x,y) \leq 0 \\ & y \in \operatorname{argmax}\{f(x,y) : h(x,y) \leq 0\} \end{aligned} \tag{2}$$

If all functions in (2) are linear, this is called the *linear maxmin problem* (LMM), which is still $\mathcal{NP}$-hard.

Introducing Integer Variables

A natural generalization is to introduce integer variables, yielding the mixed-integer zero-sum static Stackelberg problem (MZSSP):

MZSSP

$$\begin{aligned} \min_{x} \quad & f(x,y) \\ \text{subject to} \quad & g(x,y) \leq 0 \\ & x \in X_{INT} \\ & y \in \operatorname{argmax}\{f(x,y) : h(x,y) \leq 0, y \in Y_{INT}\} \end{aligned} \tag{3}$$

where X_{INT} and Y_{INT} represent integrality constraints on subsets of the leader and follower variables, respectively.

Interdiction Problems

We're interesed mainly in the special case of (3) where:

- $X_{INT} = \{0,1\}^n$
- The leader has a single budget constraint.
- The follower has variable upper bound constraints $0 \leq y \leq u(1-x)$.

This leads to the *mixed-integer zero-sum static Stackelberg problem with interdiction constraints* (MZSSPIC):

MZSSPIC

$$\begin{aligned} \min_{x} \quad & f(x,y) \\ \text{subject to} \quad & r(x) \leq R \\ & x \in \{0,1\}^n \\ & y \in \operatorname{argmax}\{f(x,y) : h(x,y) \leq 0, 0 \leq y \leq u(1-x), y \in Y_{INT}\} \end{aligned} \tag{4}$$

A Biobjective Interdiction Problem

- In many applications, the leader is not subject to a hard budget constraint.
- Instead, we may want to analyze the tradeoff between the budget and the follower's objective.
- We make the budget constraint a second objective and restrict all functions to be linear.
- This yields the *linear biobjective mixed-integer zero-sum static Stackelberg problem with interdiction constraints* (LBMZSSPIC):

LBMZSSPIC

$$\begin{aligned} \text{vmin} \quad & [rx, dy] \\ \text{subject to} \quad & x \in \{0,1\}^n \\ & y \in \operatorname{argmax}\{dy : Ay \leq b, 0 \leq y \leq u(1-x), y \in Y_{INT}\} \end{aligned} \tag{5}$$

Biobjective Integer Programs

Consider the general biobjective integer program (BIP):

$$\text{vmin}_{x \in X} [f_1(x), f_2(x)]. \tag{6}$$

We are looking for *efficient* solutions to (6).

Definition

A feasible solution $\hat{x} \in X$ is *efficient* if there is no other $x \in X$ such that

$$\begin{aligned} f_i(x) &\leq f_i(\hat{x}), \text{ for } i = 1, 2 \text{ and} \\ f_i(x) &< f_i(\hat{x}) \text{ for either } i = 1 \text{ or } i = 2. \end{aligned}$$

The set of *outcomes* is defined to be $Y = f(X) \subset \mathbb{R}^2$. The outcome corresponding to an efficient solution is called *Pareto*.

For simplicity, we generally work in outcome space.

Weighted Sum Objective

In outcome space, (6) can be restated as

$$\begin{array}{ll} \text{vmin} & y \\ \text{subject to} & y \in f(X), \end{array}$$

We can convert (6) into a single-objective problem with a nonnegative linear combination of the objective functions [Geoffrion, 1968]:

$$\min_{y \in f(X)} \alpha y_1 + (1-\alpha) y_2. \tag{7}$$

for $0 \leq \alpha \leq 1$. It is well-known that

- Solutions to (7) are
  - Pareto outcomes for any $0 \leq \alpha \leq 1$.
  - extreme points of $\text{conv}(Y)$.
- Not all Pareto outcomes are solutions to (7) for some $0 \leq \alpha \leq 1$.
- We call solutions to (7) *supported outcomes*.

The WCN Algorithm

- Substituting the *weighted Chebyshev norm* objective in (7), we get:

$$\min_{y \in f(X)} \left\{ \|y_1 - y_1^*, y_2 - y_2^*)\|_\infty^\beta \right\}, \tag{8}$$

where
  - $\|y\|_\infty^\beta = \max\{\beta|y_1|, (1-\beta)|y_2|\}$ and
  - (y_1^*, y_2^*) is the *ideal point* such that $y_i^* = \max_{y \in f(X)} y_i$.
- Every Pareto outcome is a solution to (8) for some $0 \leq \beta \leq 1$.
- In [Ralphs et al., 2004], we describe an algorithm for enumerating Pareto outcomes based on solving a sequence of linearized subproblems:

$P(\beta)$

$$\begin{aligned} \min \quad & z \\ \text{s.t.} \quad & z \geq \beta\,(y_1 - y_1^*) \\ & z \geq (1-\beta)\,(y_2 - y_2^*) \\ & y \in f(X) \end{aligned} \tag{9}$$

Back to LBMZSSPIC

Applying these results to LBMZSSPIC yields the subproblem:

$P(\beta)$

$$\begin{aligned} \max \quad & z \\ \text{subject to} \quad & z \geq \beta\,(rx - rx^*) \\ & z \geq (1-\beta)\,(dy - dy^*) \\ & x \in \{0,1\}^n \\ & y \in \operatorname{argmax}\{dy : Ay \leq b, 0 \leq y \leq u(1-x), y \in Y_{INT}\} \end{aligned} \tag{10}$$

for $0 \leq \beta \leq 1$.

Comment

$P(\beta)$ is a linear MZSSP in the form (3).

Notation

The following notations, definitions, and examples are taken partially from Moore and Bard [1990] and apply to bilevel programs of the form (3):

Notation

$$
\begin{aligned}
\Omega &= \{(x,y) : g(x,y) \leq 0, h(x,y) \leq 0\} \\
\Omega^{\text{proj}} &= \{x : \exists y \text{ with } (x,y) \in \Omega\} \\
\Omega_{INT} &= \{(x,y) \in \Omega : x \in X_{INT}, y \in Y_{INT}\} \\
\Omega_{INT}^{\text{proj}} &= \{x \in X_{INT} : \exists y \text{ with } (x,y) \in \Omega_{INT}\} \\
M(\bar{x}) &= \text{argmax}\{f(\bar{x},y) : h(\bar{x},y) \leq 0\} \\
M_{INT}(\bar{x}) &= \text{argmax}\{f(\bar{x},y) : y \in Y_{INT}, h(\bar{x},y) \leq 0\} \\
\mathcal{F} &= \{(x,y) : x \in \Omega^{\text{proj}}, y \in M(x)\} \\
\mathcal{F}_{INT} &= \{(x,y) : x \in \Omega_{INT}^{\text{proj}}, y \in M_{INT}(x)\}
\end{aligned}
$$

Branch and Bound

In branch and bound for standard mixed-integer linear programs, we relax the integrality constraints and solve the resulting LP.

Pruning Rules

1. If the relaxed problem has no feasible solution, then neither does the original problem
2. If the relaxed problem has a solution, then the objective value is a valid lower bound on that of the original problem.
3. If the solution to the relaxed subproblem is integral, then it is optimal to the original problem.

Comment

If we optimize over $\mathcal{F}$, only the first of these rules holds, since $\mathcal{F}_{INT}$ may not be contained in $\mathcal{F}$!

Example 1

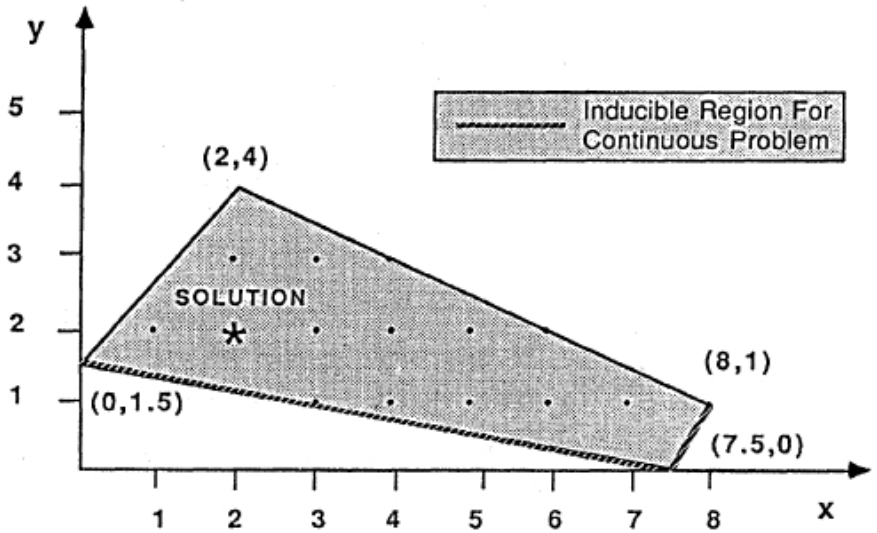
For the remainder of the talk, we will assume that all functions are linear. Consider the following mixed-integer linear BLP:

Example

$$
\begin{aligned}
\max_{x \in \mathbb{Z}^+} \quad & F(x,y) = x + 10y \\
\text{subject to} \quad & y \in \operatorname{argmin} \{ f(x,y) = y : -25x + 20y \leq 30 \\
& \qquad\qquad\qquad\qquad\quad x + 2y \leq 10 \\
& \qquad\qquad\qquad\qquad\quad 2x - y \leq 15 \\
& \qquad\qquad\qquad\qquad\quad 2x + 10y \geq 15 \\
& \qquad\qquad\qquad\qquad\quad y \in \mathbb{Z}^+ \} .
\end{aligned}
$$

Example 1 (cont)

Here we can see Ω, Ω_{INT}, $\mathcal{F}$, and $\mathcal{F}_{INT}$:



From this example, we can make the following observations:

1. OPT($\mathcal{F}$) = $\max_{(x,y)\in\mathcal{F}} F(x,y)$ is not a valid bound on the solution value of the original problem.
2. Integer solutions to $\max_{(x,y)\in\mathcal{F}} F(x,y)$ are in $\mathcal{F}_{INT}$, but are not necessarily optimal.

Example 2

Consider the following mixed-integer linear BLP:

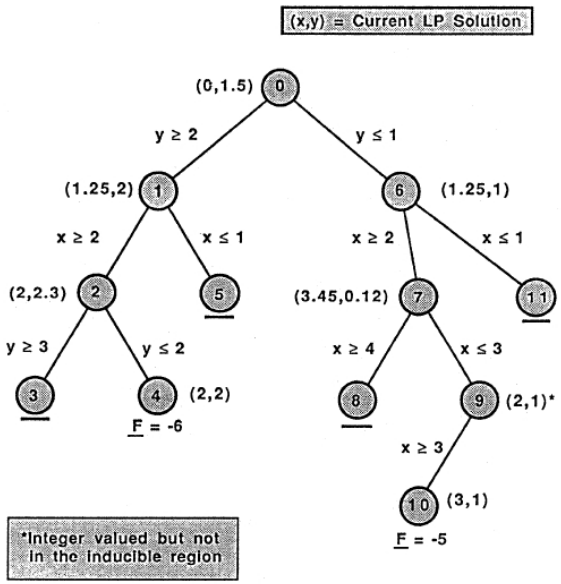
Example

$$\begin{aligned} \min_{x \in \mathbb{Z}_+} \quad & F(x,y) = x + 2y \\ \text{subject to} \quad & y \in \operatorname{argmax} \{ f(x,y) = y : -x + 2.5y \leq 3.75 \\ & \qquad\qquad\qquad\qquad\quad x + 2.5y \geq 3.75 \\ & \qquad\qquad\qquad\qquad\quad 2.5x + y \leq 8.75 \\ & \qquad\qquad\qquad\qquad\quad y \in \mathbb{Z}_+ \} . \end{aligned}$$

For this example, $\Omega_{INT} = \{(2,1),(2,2),(3,1)\}$, while $\mathcal{F}_{INT} = \{(2,2),(3,1)\}$. The optimal solution is $(3,1)$ and $F(3,1) = -5$.

Example 2 (cont)

Applying branch and bound using the continuous problem.



At node 9, optimizing over $\mathcal{F}$ with the additional constraint $y \leq 1$ yields $(x, y) = (2, 1)$, which is in Ω_{INT}, but not in $\mathcal{F}_{INT}$.

Determining whether a given point in Ω_{INT} is also in $\mathcal{F}_{INT}$ requires solution of a MIP.

Obtaining Lower Bounds

- Instead of removing the integrality constraints, we can remove the follower's objective.
- Doing so, we obtain the *integer low-point problem*, similar to that suggested in Moore and Bard [1990].

Integer Low-point Problem

$$\min_{(x,y)\in\Omega_{INT}} f(x,y) \tag{11}$$

- The low-point problem is a true relaxation and produces a valid bound, since $\mathcal{F}_{INT} \subseteq \Omega_{INT}$.
- If (11) is infeasible, then so is the original problem.

Obtaining Upper Bounds

- Although OPT($\mathcal{F}$) is not a valid lower bound, optimizing over $\mathcal{F}$ may still be a useful exercise.
- If $(\hat{x}, \hat{y})$ is a solution to OPT($\mathcal{F}$) and $\hat{x} \in X_{INT}$, we can solve the (unrestricted) follower's problem with x fixed to $\hat{x}$.
- The resulting solution (if there is one) must then be in $\mathcal{F}_{INT}$.
- In fact, we may want to optimize over $\mathcal{F}$ with the additional requirement that $x \in X_{INT}$.

Branching

- The only way to verify optimality of a solution is when it is a low-point solution and also a member of $\mathcal{F}_{INT}$
- Otherwise, we have no verifiable optimality conditions.
- This means we may have to branch, even when all variables are integer.
- This is only necessary for the leader's variables.
- There are many ways of performing such integer branching.
- We will cover the details in the case of the LBMZSSPIC below.

Basic Branch and Bound Algorithm

The following is the loop for processing a node in the branch-and-bound algorithm for solving a mixed-integer linear bilevel program.

Processing a Node

1. Solve the low-point problem (11) to obtain $(\bar{x}, \bar{y})$.
   1. If infeasible, fathom.
   2. If feasible and the bound exceeds the value of the incumbent, fathom.
2. Optimize over $\mathcal{F}$ (optionally requiring $x \in X_{INT}$) to obtain $(\hat{x}, \hat{y})$.
3. If $\hat{x} \in X_{INT}$, solve the follower's problem with $x = \hat{x}$ and update the incumbent if possible.
4. If all leader variables are fixed, then fathom.
5. Otherwise, branch.

This is similar to the scheme proposed by Moore and Bard [1990].

Optimizing over $\mathcal{F}$

Comment

In the continuous BLP, the follower's problem is an LP, so we can replace the optimality constraint with KKT conditions.

Taking this approach yields a nonconvex NLP. Two main approaches have been taken to solve this problem:

1. Linearize complementary slackness constraints by introducing binary variables and solve the 0-1 program with a MIP solver [Fortuny-Amat and McCarl, 1981].
2. Relax the complementary slackness conditions and branch on KKT multipliers, checking the complementary slackness conditions at each iteration [Bard and Moore, 1990].

Currently, we take the first approach, which allows us to easily enforce $x \in X_{INT}$.

Solver for the LBMZSSPIC

- So far, we have focused on generating supported solutions of (5).
- This requires solving a weighted sum subproblem with only bound constraints on the leader's variables.
- The low-point problem is a MILP and can be solved using standard methods.
- We use the following integer branching rule:

Pair Branching

- Let $(\hat{x}, \hat{y})$ be the solution obtained in Step 2 of the node processing loop.
- Let i and j be the indices of two unfixed leader variables.
- Create three child nodes as follows
  1. $x_i = 1 - \hat{x}_i$ and $x_j = \hat{x}_j$
  2. $x_i = \hat{x}_i$ and $x_j = 1 - \hat{x}_j$
  3. $x_i = 1 - \hat{x}_i$ and $x_j = 1 - \hat{x}_j$

Implementation

The algorithm has been implemented using software available from the Computational Infrastructure for Operations Research (COIN-OR) repository.

COIN-OR Components Used

- The Abstract Library for Parallel Search (ALPS) 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.

Please visit `www.coin-or.org` for information on obtaining these codes.

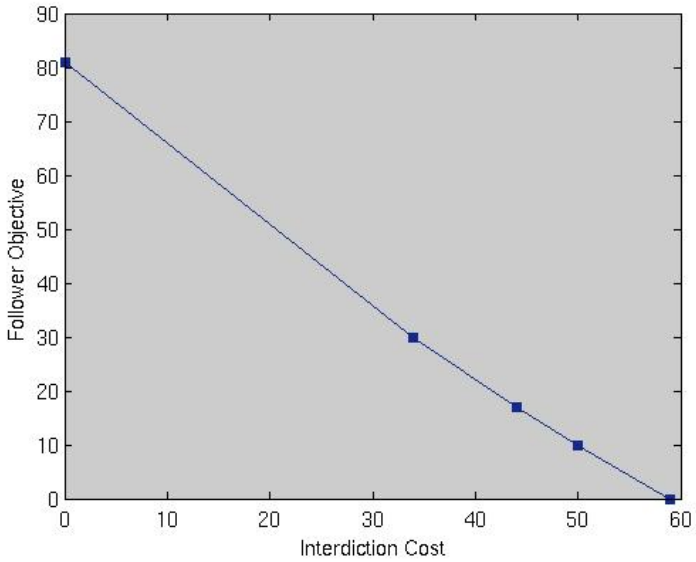
Illustrating the Tradeoff

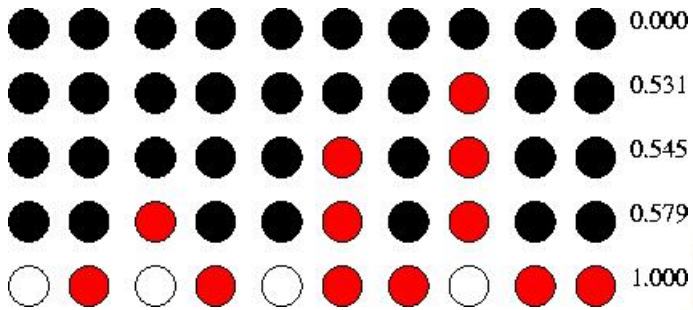
From this example we can see how the solution evolves:

$$
\begin{aligned}
\text{vmin} \quad & [rx, dy] \\
\text{subject to} \quad & x \in \{0,1\}^n \\
& y \in \operatorname{argmax}\{dy : Ay \leq b \\
& \qquad\qquad 0 \leq y \leq 1 - x \\
& \qquad\qquad y \in \{0,1\}^n\}
\end{aligned}
$$

where

$$
\begin{aligned}
r &= (5, 5, 10, 3, 3, 6, 8, 9, 3, 7), \\
d &= (11, 12, 13, 15, 15, 7, 12, 10, 19, 16), \\
A &= (5, 6, 7, 8, 9, 2, 5, 7, 10, 9), \\
b &= 40
\end{aligned}
$$





n	r	range	problems solved	cpu(s)
10	0.50	[1,300]	13	131.84
10	0.50	[1,300]	13	188.73
10	0.50	[1,300]	9	83.55

An example of the generated frontier:

Future Directions

The following directions are planned for the near future:

- Better methods for optimizing over $\mathcal{F}$.
- Better methods for finding heuristic interdiction plans.
- Implementation of the WCN algorithm.
- Parallel solution of subproblems and global solution procedure.
- Extensive testing of interdiction for various problem classes
  - Combinatorial Problems
  - Generic MILPs

References

J.F. Bard and J.T. Moore. A branch and bound algorithm for the bilevel programming problem. *SIAM Journal on Scientific and Statistical Computing*, 11(2):281–292, 1990.

O. Ben-Ayed and C. Blair. Computational difficulties of bilevel linear programming. *Operations Research*, 38: 556–560, 1990.

P. Calamai and L. Vincente. Generating quadratic bilevel programming problems. *ACM Transactions on Mathematical Software*, 20:103–119, 1994.

J. Fortuny-Amat and B. McCarl. A representation and economic interpretation of a two-level programming problem. *Journal of the Operations Research Society*, 32:783–792, 1981.

A.M. Geoffrion. Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, 22:618–630, 1968.

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.

R. Jeroslow. The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming*, 32: 146–164, 1985.

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

T.K Ralphs, M.J. Saltzman, and M.M. Wiecek. An improved algorithm for biobjective integer programming and its application to network routing problems. Technical Report 04T-004, Lehigh University Industrial and Systems Engineering, 2004.