Reformulation Heuristics for Generalized Interdiction Problems

M. Fischetti[1] M. Monaci[2] M. Sinnl[3]

[1] DEI, University of Padua, Italy
[2] DEI, University of Bologna, Italy
[3] ISOR, University of Vienna, Austria

January 13^{th}, 2017
Aussois, France

Bilevel Optimization

General bilevel optimization problem

$$\min_{x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}} F(x, y)$$
$$G(x, y) \leq 0$$
$$y \in \arg\max_{y' \in \mathbb{R}^{n_2}} \{ f(x, y') : g(x, y') \leq 0 \}$$

- leader vs follower
- Stackelberg game: two-person sequential game
- optimistic vs pessimistic

Bilevel Optimization

General bilevel optimization problem

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

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

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

Leader

- leader vs follower
- Stackelberg game: two-person sequential game
- optimistic vs pessimistic

Bilevel Optimization

General bilevel optimization problem

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

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

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

Leader

Follower

- leader vs follower
- Stackelberg game: two-person sequential game
- optimistic vs pessimistic

Value function reformulation

- Optimal solution of the follower for a given $x \in \mathbb{R}^{n_1}$

$$\Phi(x) = \max_{y' \in \mathbb{R}^{n_2}} \{ f(x, y') : g(x, y') \leq 0 \}$$

- Reformulation of the bilevel problem

$$\begin{aligned} \min_{x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}} F(x, y) \\ G(x, y) \leq 0 \\ f(x, y) \geq \phi(x) \\ g(x, y) \leq 0 \end{aligned}$$

Standard Interdiction Problems

Class of bilevel optimization problems in which

- all objective functions and constraints are linear
- leader and follower have opposite objective functions
- leader may interdict a set N of items of follower
  - interdiction budget
  - discrete vs linear interdiction
- two-person, zero-sum sequential game
- studied mostly for network-based problems in the follower

$$\min_{\substack{x \in \mathbb{R}^{n_1} \\ G_x x \leq G_0}} \max_{y \in \mathbb{R}^{n_2}} d^T y$$

$$\begin{aligned} By &\leq b & \\ 0 \leq y_j &\leq UB_j(1 - x_j), & \forall j \in N \\ x_j &\in \{0, 1\}, & \forall j \in N \\ y_j &\text{ integer}, & \forall j \in J_y \end{aligned}$$

Standard Interdiction Problems

- leader has
  - variables $x \in \mathbb{R}^{n_1}$; *interdiction* variables x_j ($j \in N$) are binary
  - constraints $G_x x \leq G_0$
- follower has
  - variables $y \in \mathbb{R}^{n_2}$; variables y_j ($j \in J_y$) are integer
  - constraints $By \leq b$ plus interdiction constraints: $x_j = 1 \Rightarrow y_j = 0$
    $x_j = 0 \Rightarrow 0 \leq y_j \leq UB_j$
  - value function $\Phi(x) = \max_{y \in \mathbb{R}^{n_2}} \{d^T y : (9) - (10)\}$
- objective of leader and follower sum up to zero

$$\min_{x,y} \phi(x) \tag{1}$$

$$G_x x \leq G_0 \tag{2}$$

$$x_j \in \{0, 1\}, \qquad \forall j \in N \tag{3}$$

$$By \leq b \tag{4}$$

$$y_j \text{ integer}, \qquad \forall j \in J_y \tag{5}$$

$$0 \leq y_j \leq UB_j(1 - x_j), \qquad \forall j \in N \tag{6}$$

Generalized Interdiction Problems (*GIP*s)

We consider a generalization of Standard Interdiction Problems in which

- leader and follower may have different objective functions,
- leader constraints may involve both x and y variables
  $G_x x \leq G_0 \Rightarrow G_x x + G_y y \leq G_0$

These are Bilevel Mixed Integer Optimization Problems in which

- some leader variables (the *interdiction* variables) are binary
- no leader variables appear in the follower but the interdiction variables (that are in the interdiction constraints)

Generalized Interdiction Problems

$$(GIP) \quad \min_{x,y} c_x^T x + c_y^T y$$

$$
\begin{aligned}
G_x x + G_y y &\leq G_0 && \\
x_j &\in \{0,1\}, && \forall j \in N \\
x_j &\text{ integer}, && \forall j \in J_x \\
By &\leq b && \\
y_j &\text{ integer}, && \forall j \in J_y \\
d^T y &\geq \Phi(x) && \\
0 \leq y_j &\leq UB_j (1 - x_j), && \forall j \in N
\end{aligned}
$$

State of the art

Many exact and approximate algorithms for specific applications.

- Mixed-Integer Bilevel Optimization
  - Exact approaches: DeNegre [2011], DeNegre and Ralphs [2009], Fischetti et al. [2016a,b], Moore and Bard [1990], Xu and Wang [2014]
  - Heuristics: DeNegre [2011]

- General Standard Interdiction
  - Exact approaches: branch-and-cut by Fischetti et al. [2016c] (requires monotonicity of the follower).
    Very effective in practice, but challenging to be implemented.
  - Heuristics: greedy algorithm by DeNegre [2011].
    Pick an interdiction policy by taking variables x_j ($j \in N$) according to non-increasing d_j values, until the leader budget is reached.
    Very simple and fast, but poor results.

GIP: Follower subproblem

$$\Phi(x) = \max_y \{ d^T y : By \le b, \quad 0 \le y_j \le UB_j \, (1 - x_j) \quad (j \in N)$$
$$y_j \text{ integer} \quad (j \in J_y) \}$$

- Interdiction constraints impose bilinear conditions

$$x_j \, y_j = 0 \quad \forall j \in N$$

- These conditions can be relaxed in a Lagrangian fashion and yield the penalized objective function

$$\max d^T y - \sum_{j \in N} M_j x_j \, y_j$$

where $M_j >> 0$

- Apparently, the objective function is bilinear ...
- ... but actually it is linear, as follower is solved for a given (fixed) x

Follower subproblem: reformulation

$$\Phi(x) = \max_y \{ d^T y : By \le b, \quad 0 \le y_j \le UB_j (1 - x_j) \quad (j \in N)$$
$$y_j \text{ integer} \quad (j \in J_y) \}$$

$$\Downarrow$$

$$\Phi(x) = \max_y \{ d^T(x) y : By \le b, \quad y_j \text{ integer} \quad (j \in J_y), \quad y \ge 0 \}$$

with

$$d_j(x) := \begin{cases} d_j - M_j x_j, & \text{if } j \in N \\ d_j, & \text{otherwise} \end{cases} \qquad \forall j \in N_y \tag{7}$$

Follower subproblem: LP relaxation

- Optimal value of the LP relaxation of the follower problem

$$\overline{\Phi}(x) := \max\{d(x)^T y : By \leq b, \quad y \geq 0\} \tag{8}$$

- Assuming problem (21) is bounded and feasible, standard LP duality gives

$$\overline{\Phi}(x) := \min\{u^T b : u^T B \geq d^T(x), \quad u \geq 0\}$$

- As $\overline{\Phi}(x) \geq \Phi(x)$ imposing $f(x, y) \geq \overline{\Phi}(x)$ in the value function reformulation produces a heuristic single-level reformulation for *GIP*:

$$
\begin{aligned}
(\overline{GIP}) \qquad & \min c_x^T x + c_y^T y \\
& G_x x + G_y y \leq G_0 \\
& x_j \in \{0, 1\}, && \forall j \in N \\
& x_j \text{ integer}, && \forall j \in J_x \\
& By \leq b \text{ and } y \geq 0 \\
& y_j \leq UB_j(1 - x_j), && \forall j \in N \\
& u^T B \geq d(x)^T \text{ and } u \geq 0 \\
& d^T y \geq u^T b.
\end{aligned}
$$

Relation between GIP and $\overline{GIP}$

- $\overline{GIP}$ is not a relaxation nor a restriction of the original GIP problem
  - integrality on the y variables is relaxed in both the leader and the follower
- $\overline{GIP}$ is a restriction of GIP in case integrality on the y is redundant in the leader
  - e.g., standard interdiction problems (no y in the leader)
- $\overline{GIP}$ is a relaxation of GIP in case integrality on the y is redundant in the follower
  - e.g., the follower constraint matrix is totally unimodular
- $\overline{GIP}$ coincides with GIP if integrality on the y is redundant in the both the leader and the follower
  - i.e., $J_y = \emptyset$
  - exact single-level reformulation of GIP

The ONE-SHOT heuristic

(1) Relax the integrality of the y variables;
(2) Restate the resulting problem as $(\overline{GIP})$;
(3) Solve the resulting single-level MILP (possibly with a time limit), and let $(\bar{x}, \cdot)$ be the optimal (or best) solution found;
(4) *Refine* $\bar{x}$ and obtain solution $(\bar{x}, \bar{y})$.

Step 4 computes a complete feasible *GIP* solution $(\bar{x}, \bar{y})$ starting from a leader vector $\bar{x}$ as follows:

(a) Solve the follower MILP for $x = \bar{x}$ to compute $\bar{\varphi} := \Phi(\bar{x})$;
(b) Restrict *GIP* by fixing $x = \bar{x}$ and replacing the nonlinear value function constraint with $d^T y \geq \varphi$;
(c) Solve the resulting MILP model to obtain $(\bar{x}, \bar{y})$
(no need of steps (b) and (c) for Standard Interdiction Problems)

Typically, the solution of this step is not time-consuming.

The ITERATE heuristic

(1) Relax the integrality of the y variables;

(2) Restate the resulting problem as ($\overline{GIP}$);

(3) Solve the resulting single-level MILP (possibly with a time limit), and let $(\bar{x}^1, \cdot)$, $(\bar{x}^2, \cdot)$, $\ldots$, $(\bar{x}^K, \cdot)$ be a collection of solutions found;

(4) Refine each such solution, possibly updating the incumbent;

(5) Add a no-good constraint for each solution $(\bar{x}^k, \cdot)$, and repeat steps 3 and 4 until the time limit is met.

The ITERATE & CUT heuristic

Observation: the smaller the follower integrality gap, the better the single-level MILP reformulation ($\overline{GIP}$) approximates GIP.

- At each iteration, strengthen the follower MILP by adding valid inequalities, that exploit integrality of the y variables.

  Recall: x variables appear only in the objective function in the follower
  $\Rightarrow$ all feasibility-based cuts that can be derived by the follower are valid $\forall x$.

- The new cuts are dualized on the fly adding new dual variables
- This gives an extended formulation
  - that is sometimes harder to solve
  - but which provides a better approximation of GIP.

Computational settings

- CPLEX 12.6.3, Intel Xeon E3-1220V2 3.1 GHz, single thread
- Only standard interdiction instances from the literature (so far)
  - Knapsack Interdiction Problem (KIP): sets `CCLW`, `TRSK` and `D`
  - Multiple Knapsack Interdiction Problem (MKIP): set `SAC`
  - Clique Interdiction Problem (CIP): set `TRSC`
  - Firefighter Problem (FP): set `FIRE`
- For all these instances the optimal solution value has been computed using the general-purpose exact algorithms by Fischetti et al. [2016b, 2016c] ...
- ... though some problems are extremely challenging – optimal solution for some instances required one or more hours of computing time to the exact algorithm
- Very short time limit for the heuristics: 10 seconds per instance

Algorithms tested

- ONE-SHOT (OS)
- ONE-SHOT+ (OS+): same as ONE-SHOT, but several solutions of the reformulation are generated by using Cplex's POPULATE (and then refined)
- ITERATE (I)
- ITERATE+ (I+): same as ITERATE, but several solutions of the reformulation are generated through POPULATE (and then refined) at each iteration of the while loop
- ITERATE & CUT+ (IC+): same as ITERATE+, but the Cplex's root cuts generated during the REFINE procedure are collected and added to the follower model
- GREEDY (GD): greedy heuristic proposed by DeNegre [2011]
- GRASP (GR): GRASP variant of GREEDY

Very fast heuristics

- #opt = number of optimal solutions
- %gap = average primal gap, computed as $100 \cdot |z^{heu} - z^{opt}|/(|z^{opt}| + 10^{-10})$.

		GD		OS		OS+	
set	#inst	#opt	%gap	#opt	%gap	#opt	%gap
CCLW	50	5	17.56	44	0.12	49	0.00
TRSK	150	37	11.79	93	2.00	140	0.35
D	160	58	14.24	154	0.16	160	0.00
SAC	144	7	19.68	121	0.14	136	0.07
TRSC	80	1	98.96	5	73.96	14	44.17
FIRE	72	5	30.43	15	29.69	52	5.68
sum	656	113		432		551	
average			32.11		17.68		8.38

Fast heuristics (time limit = 10 seconds)

- #opt = number of optimal solutions
- %gap = average primal gap, computed as $100 \cdot |z^{heu} - z^{opt}|/(|z^{opt}| + 10^{-10})$.

		GR		I		I+		IC+	
set	#inst	#opt	%gap	#opt	%gap	#opt	%gap	#opt	%gap
CCLW	50	10	6.93	50	0.00	50	0.00	50	0.00
TRSK	150	79	3.82	150	0.00	150	0.00	149	0.08
D	160	115	1.89	160	0.00	160	0.00	160	0.00
SAC	144	30	8.75	142	0.04	141	0.05	143	0.00
TRSC	80	5	70.62	22	37.92	26	29.79	52	14.17
FIRE	72	20	13.27	53	5.67	56	4.24	55	4.58
sum	656	259		577		583		609	
average			17.55		7.27		5.68		3.14

Conclusions

- Fast and effective heuristics for Generalized Interdiction Problems
- Simplest algorithm is very simple to implement;
  - implementation hardness is comparable to that of the greedy algorithm
  - though it provides much better results
  - and computes the optimal solution in about 2/3 of the instances in our testbed.
- More sophisticated versions of the approach provide even better results:
  - in about 92% of the cases an optimal solution is computed within 10 seconds
  - and the average primal gap is below 3%

- To do: extensive computational analysis on *GIP* instances
- Possible extension to general Bilevel Optimization Problems?