Reformulation Heuristics for Generalized Interdiction Problems

M. Fischetti\textsuperscript{1}  M. Monaci\textsuperscript{2}  M. Sinnl\textsuperscript{3}

\textsuperscript{1} DEI, University of Padua, Italy  
\textsuperscript{2} DEI, University of Bologna, Italy  
\textsuperscript{3} ISOR, University of Vienna, Austria

January 13\textsuperscript{th}, 2017  
Aussois, France
Bilevel Optimization

General bilevel optimization problem

\[
\begin{align*}
\min_{x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}} & \quad F(x, y) \\
& \quad G(x, y) \leq 0 \\
& \quad y \in \arg \max_{y' \in \mathbb{R}^{n_2}} \{ f(x, y') : g(x, y') \leq 0 \}
\end{align*}
\]

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

General bilevel optimization problem

\[
\begin{align*}
\min_{x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}} & \quad F(x, y) \\
G(x, y) & \leq 0 \\
y & \in \arg\max_{y' \in \mathbb{R}^{n_2}} \{ f(x, y') : g(x, y') \leq 0 \}
\end{align*}
\]

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

General bilevel optimization problem

\[
\begin{align*}
\min_{x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}} & \quad F(x, y) \\
G(x, y) & \leq 0
\end{align*}
\]

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

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

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

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

$$f(x, y) \geq \phi(x)$$

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

\[
\begin{align*}
\min_{x \in \mathbb{R}^n_1} \quad & \max_{y \in \mathbb{R}^n_2} \quad d^T y \\
G_x x & \leq G_0 \\
By & \leq b \\
0 & \leq y_j \leq UB_j(1 - x_j), \quad \forall j \in N \\
x_j & \in \{0, 1\}, \quad \forall j \in N \\
y_j & \text{integer}, \quad \forall j \in J_y
\end{align*}
\]
Standard Interdiction Problems

- **leader** has
  - variables $x \in \mathbb{R}^n$; *interdiction* variables $x_j$ ($j \in N$) are binary
  - constraints $G_x x \leq G_0$

- **follower** has
  - variables $y \in \mathbb{R}^n$; 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

\[
\min_{x,y} \phi(x) \quad (1)
\]

\[
G_x x \leq G_0 \quad (2)
\]

\[
x_j \in \{0, 1\}, \quad \forall j \in N \quad (3)
\]

\[
By \leq b \quad (4)
\]

\[
y_j \text{ integer}, \quad \forall j \in J_y \quad (5)
\]

\[
0 \leq y_j \leq UB_j(1 - x_j), \quad \forall j \in N \quad (6)
\]
Generalized Interdiction Problems (GIPs)

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

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

\[G_x x + G_y y \leq G_0\]
\[x_j \in \{0, 1\}, \quad \forall j \in N\]
\[x_j \text{ integer}, \quad \forall j \in J_x\]

\[B y \leq b\]
\[y_j \text{ integer}, \quad \forall j \in J_y\]

\[d^T y \geq \Phi(x)\]

\[0 \leq y_j \leq UB_j (1 - x_j), \quad \forall j \in N\]
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**

\[ \Phi(x) = \max_y \left\{ d^T y : By \leq b, \quad 0 \leq y_j \leq UB_j (1 - x_j) \quad (j \in N) \right\} \]

\[ 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 \( M_j \gg 0 \)

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

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

\[
\downarrow
\]

\[
\Phi(x) = \max_y \{ d^T(x) y : By \leq b, \quad y_j \text{ integer} \quad (j \in J_y), \quad y \geq 0 \}
\]

with

\[
d_j(x) := \begin{cases} 
    d_j - M_j x_j, & \text{if } j \in N \\
    d_j, & \text{otherwise}
\end{cases} \quad \forall j \in N_y \quad (7)
\]
Follower subproblem: LP relaxation

- Optimal value of the LP relaxation of the follower problem
  \[
  \Phi(x) := \max \{d(x)^T y : By \leq b, \quad y \geq 0\} \tag{8}
  \]

- Assuming problem (21) is bounded and feasible, standard LP duality gives
  \[
  \Phi(x) := \min \{u^T b : u^T B \geq d^T(x), \quad u \geq 0\}
  \]

- As \(\Phi(x) \geq \Phi(x)\) imposing \(f(x, y) \geq \Phi(x)\) in the value function reformulation produces a heuristic single-level reformulation for \(GIP\):
  \[
  (GIP) \quad \min c_x^T x + c_y^T y \\
  G_x x + G_y y \leq G_0 \\
  x_j \in \{0, 1\}, \quad \forall j \in N \\
  x_j \text{ integer}, \quad \forall j \in J_x \\
  By \leq b \text{ and } y \geq 0 \\
  y_j \leq UB_j (1 - x_j), \quad \forall j \in N \\
  u^T B \geq d(x)^T \text{ and } u \geq 0 \\
  d^T y \geq u^T b.
  \]
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 $(\overline{x}, \cdot)$ be the optimal (or best) solution found;
4. **Refine** $\overline{x}$ and obtain solution $(\overline{x}, \overline{y})$.

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

(a) Solve the follower MILP for $x = \overline{x}$ to compute $\overline{\varphi} := \Phi(\overline{x})$;
(b) Restrict $GIP$ by fixing $x = \overline{x}$ and replacing the nonlinear value function constraint with $d^T y \geq \varphi$;
(c) Solve the resulting MILP model to obtain $(\overline{x}, \overline{y})$ (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 \((\overline{x}^1, \cdot), (\overline{x}^2, \cdot), \ldots, (\overline{x}^K, \cdot)\) be a collection of solutions found;
4. Refine each such solution, possibly updating the incumbent;
5. Add a no-good constraint for each solution \((\overline{x}^k, \cdot)\), 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 $100 \cdot \frac{|z^{heu} - z^{opt}|}{(|z^{opt}| + 10^{-10})}$.

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

M. Monaci (uniBO) Reformulation Heuristics for GIPs
Fast heuristics (time limit = 10 seconds)

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

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