Enhanced exact algorithms for discrete bilevel linear problems

Seminar: Selected Topics in bilevel optimization

Leon Eifler

February 1, 2017

1 Overview

We present the work of Caramia and Mari [1], who introduced two new exact algorithms for solving bilevel linear problems in which all the variables are discrete. These Discrete Bilevel Linear Problems (DBLPs) can be defined as:

$$
\begin{aligned}
\text{(DBLP)} \quad & \min_{x,y} F(x,y) = c_1^T x + c_2^T y \\
& \text{s.t.} \quad Cx + Dy \leq e \\
& \qquad\quad x \in \mathbb{Z}_+^n \\
& \qquad\quad y \in \arg\min_y \quad f(y) = d^T y \\
& \qquad\quad \text{s.t.} \qquad\quad Ax + By \leq b \\
& \qquad\qquad\qquad\qquad\quad y \in \mathbb{Z}_+^m
\end{aligned}
$$

The first Algorithm is a cutting plane method, where each added inequality separates multiple bilevel infeasible solutions. The second is a branch and cut method that exploits a geometric property of bilevel linear problems. The computational results are compared to the method proposed by DeNegre and Ralphs [2].

2 Preliminaries

We will introduce notation that we will need when discussing feasible solutions of (DBLP). First we define the *constrained region*, where we omit integrality requirements on the variables.

$$S = \{(x,y) \mid x \in \mathbb{Z}_+^m, y \in \mathbb{Z}_+^m, Cx + Dy \leq e, Ax + By \leq b\}$$

For each possible value of the leader variables x we define the *followers feasible set*

$$\Omega_y(x) = \{y \mid y \in \mathbb{Z}_+^m, By \leq b - Ax\}$$

and the set of all y that minimize the followers objective, called the *reaction set*

$$R_y(x) = \arg\min_y \{f(y) \text{ s.t. } y \in \Omega_y(x).$$

Finally the set of all bilevel feasible solutions, the *inducible region*, is defined as

$$IR = \{(x,y) \mid x \in \mathbb{Z}_+^n, Cx + Dy \leq e, y \in R_y(x)\}.$$

The relaxation that we will use to solve (DBLP) we call the Single Level linear Problem (SLP), defined as

$$
\begin{aligned}
\text{(SLP)} \quad & \min_{x,y} F(x,y) = c_1^T x + c_2^T y \\
& \text{s.t.} \quad (x,y) \in S.
\end{aligned}
$$

Note that this is indeed a relaxation as every optimal solution of (SLP) is a lower bound to the optimal solution of (DBLP).

3 A cutting plane method

The basic idea of the cutting plane method can be described as follows

Solve relaxation Solve (SLP) and obtain an optimal solution $(\bar{x}, \bar{y})$. If it is bilevel feasible, then (DBLP) is solved.

Add inequality If $(\bar{x}, \bar{y})$ is not bilevel feasible, add an inequality to (SLP) that cuts off all bilevel infeasible solutions at $\bar{x}$. Then solve the new relaxation again and repeat until (DBLP) is solved.

We will now give a detailed explanation of this inequality. Let $(\bar{x}, \bar{y})$ be an optimal solution of (SLP) that is not bilevel feasible. Then there exists a bilevel feasible point $(\bar{x}, \hat{y})$ such that $f(\hat{y}) < f(\bar{y})$. Now we introduce a valid inequality that is only active at $x = \bar{x}$

$$f(y) \leq f(\hat{y}) + L\|x - \bar{x}\|_\infty.$$

This non-linear inequality can be reformulated as an optimization problem

$$f(y) \leq f(\hat{y}) + L\hat{z},$$

where $\hat{z}$ is the optimal solution of

$$\begin{aligned} (P_{\hat{z}}) \quad & \min_z z \\ \text{s.t.} \quad & z \geq x_i - \bar{x}_i \quad i = 1, \ldots, n \\ & z \geq \bar{x}_i - x_i \quad i = 1, \ldots, n. \end{aligned}$$

Adding this to (SLP) turns it into a bilevel problem itself, but with continious follower variables. Note that bilevel problems with continuous follower variables can be transformed into a single level problem using KKT conditions, as described by Fortuny-Amat and McCarl [3]. The advantage of this cutting plane method is that a large number of bilevel-infeasible solutions are cut off at each iteration. However, as each inequality adds a follower problem with $2n$ constraints the iterations are computationally expensive.

A possible modification to reduce the number of added follower problems is to remove added inequalities under certain conditions. If a bilevel infeasible solution $(\bar{x}, \bar{y})$ was cut off and the next bilevel infeasible solution (x', y') is far from $(\bar{x}, \bar{y})$ in S, then it is reasonable that the descent direction of the objective leads away from $(\bar{x}, \bar{y})$. Therefore the subproblem that $P_{\bar{z}}$ that cut off $(\bar{x}, \bar{y})$ is replaced by the inequality $F(x, y) \geq F(x', y')$. To avoid cycling, if $P_{\bar{z}}$ is reintroduced at some point, it will not be dropped again.

4 A branch and cut algorithm

The constrained region S that is used as a relaxation to solve DBLP's usually contains a large amount of integer points that are not bilevel feasible. The motivation behind the branch and cut algorithm presented here is to split the constrained region into two parts S', S'', where S' is likely to contain the optimal solution. We first solve the max-min problem:

$$
\begin{aligned}
(BLP^{max}_{min}) \quad & \max_{x,y} f(y) = d^T y \\
& \text{s.t.} \quad x \in \mathbb{R}^n_+ \\
& \qquad y \in \arg\min_{y} f(y) = d^T y \\
& \qquad \text{s.t.} \quad Ax + By \leq b \\
& \qquad\quad y \in \mathbb{R}^m_+
\end{aligned}
$$

So we take the maximum of the followers objective over the set of all continuous bilevel feasible points. Let $(\hat{x}, \hat{y})$ the the optimal solution of BLP^{max}_{min}. Then the inequality

$$f(y) \leq \lceil f(\hat{y}) \rceil$$

is valid for bilevel linear problems. For the corresponding DBLP it might not be valid but it is likely that most discrete bilevel feasible points fullfill $f(y) \leq \lceil f(\hat{y}) \rceil$. Therefore we split the constrianed region S into

$$
\begin{aligned}
S' &= \{(x, y) \in S \mid f(y) \leq \lceil f(\hat{y}) \rceil\}, \\
S'' &= \{(x, y) \in S \mid f(y) \geq \lceil f(\hat{y}) \rceil + 1\}.
\end{aligned}
$$

Then the framework for the branch and cut algorithm is the following

Step 0 Determine S' and S''.

Step 1 Solve the subproblem $(DBLP')$ induced by S'. Use continuous single-level relaxation.

Step 2 Branch if solution is not integral, add cut if solution is integral but not bilevel feasible. Repeat until $(DBLP')$ is solved.

Step 3 Compute LB'' for problem $(DBLP'')$ induced by S''. If this is worse than the UB found before, stop. Otherwise solve $(DBLP'')$.

Note that this is just a general framework that can be used with any cutting plane method and any branching rule. The authors use the valid inequality proposed by DeNegre and Ralphs [2] to cut off bilevel infeasible points. This general framework can be used to incorporate the cutting plane method from section 3 in a hybrid branch and cut algorithm. The authors propose to use branch and cut to solve $(DBLP')$ and then use the cutting plane method for $(DBLP'')$. In Step 3 we expect that we just have to find a decent lower bound for $(DBLP'')$ in order to prove optimality for the solution found in Step 2. The cutting plane method is efficient in finding tight lower bounds.

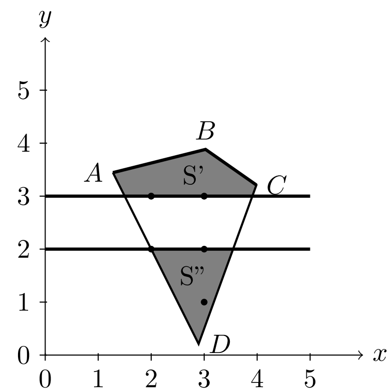
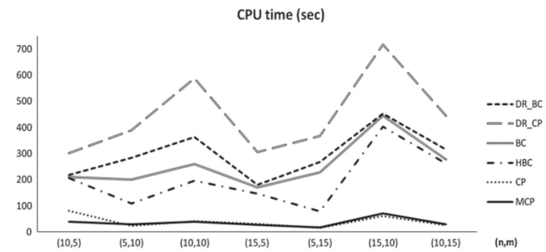


Figure 1: Example of splitting of S. Inducible region of linear problem is (A - B - C)

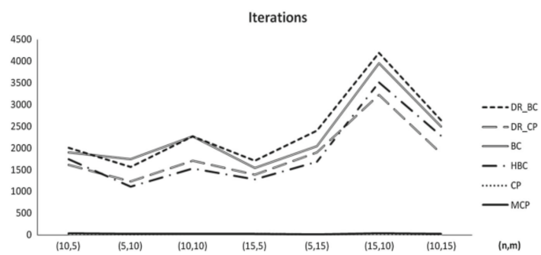
5 Computational Analysis

The cutting plane method (CP) as well as the modified version where used cuts are deleted (MCP) are compared to the cutting plane method by DeNegre and Ralphs (DN_CP). In the same way the branch and cut algorithm (BC) as well as the hybrid variant (HBC) are compared to the branch and cut algorithm by DeNegre and Ralphs (DN_BC).

The authors implemented all algorithms including (BC) and (DN_BC) in C programming language and tested them on a PC Pentium Core 2 Duo with a 2 GHz processor and 1 GB RAM, with CPLEX 12.3 as the solver. Two sets of test instances were used, a set of random instances and a set of modified MIPLIB 2010 instances. The cutting plane methods need on average 90% less computational time than (DN_CP). For the branch and cut algorithms both (BC) and (HBC) were faster than (DN_BC), where the difference in running time ranges from -0.5% to -70.6%. The overall comparison on both sets can be seen in the following two diagrams.

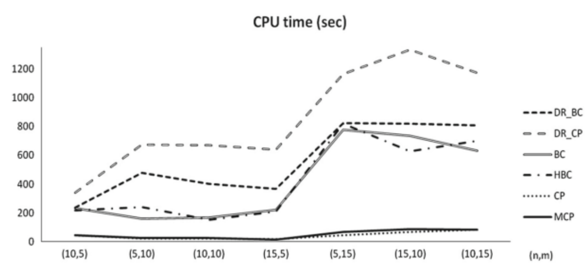


(a) Computational time

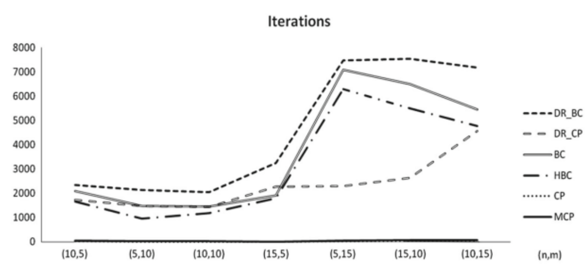


(b) Number of iterations

Figure 2: Computational comparison for random set



(a) Computational time



(b) Number of iterations

Figure 3: Computational comparison for miplib set

References

[1] M. Caramia and R. Mari. Enhanced exact algorithms for discrete bilevel linear problems. *Optimization Letters*, 9(7):1447–1468, 2015.

[2] S. T. DeNegre and T. K. Ralphs. *A Branch-and-cut Algorithm for Integer Bilevel Linear Programs*, pages 65–78. Springer US, Boston, MA, 2009.

[3] J. Fortuny-Amat and B. McCarl. A representation and economic interpretation of a two-level programming problem. *Journal of the Operational Research Society*, 32(9):783–792, 1981.