A Branch and Bound Algorithm for Solving the Binary Bi-level Linear Programming Problem

John Karlof and Peter Hocking

Mathematics and Statistics Department
University of North Carolina Wilmington
Wilmington, NC 28480

1 Introduction

The standard mathematical programming problem involves finding an optimal solution for just one decision maker. But many planning problems contain an hierarchical decision structure, each with independent, and often conflicting objectives. These types of problems can be modeled using a multilevel programming approach. Bilevel programming is the simpliest class of multilevel programming problems in which there are two independent decision makers. An upper level decision maker and a lower level decision maker. One example might be the CEO of a company as the upper level decision maker and the head of a division of the company as the lower level decision maker.

Algorithms have been proposed to solve the bilevel linear programming problem and the mixed integer binary bilevel linear programming problem. In this paper, we develop an branch and bound algorithm for solving the binary bilevel linear programming problem. In section 2, we present the formulations for the various bi-level linear programming problems and discuss previous results. Section 3 contains results we use to establish upper bounds for the branches of the branch and bound tree. Our algorithm is contained in section 4 and section 5 contains computational results.

2 Bilevel Programming

Bilevel programming problems are characterized by two levels of hierarchical decision making. The top planner makes its decision in full view of the bottom planner. Each planner attempts to optimize its objective function and is affected by the actions of the other planner. The properties of bilevel programming problems are summarized as follows: [2]

1. The system has interacting decision making units within a hierarchical structure.
2. The lower unit executes its policies after, and in view of, the decisions of the higher unit.
3. Each level maximizes net benefits independently, no compromise is possible.
4. The effect of the upper decision maker on the lower problem is reflected in both its objective function and set of feasible decisions.

This idea was further developed by Bard and Falk in a 1989 paper [1] where they considered the following mixed integer bi-level linear programming problem. We use BLPP to denote bi-level linear programming problem.

Let x^1 be an n_{11}-dimensional vector of continuous variables and x^2 be an n_{12}-dimensional vector of discrete variables, where $\mathbf{x} \equiv (x^1, x^2)$ and $n_1 = n_{11}+n_{12}$. Similarly let y^1 be an n_{21}-dimensional vector of continuous variables and y^2 be an n_{22}-dimensional vector of discrete variables, where $\mathbf{y} \equiv (y^1, y^2)$ and $n_2 = n_{21} + n_{22}$.
This leads to

$$\max_{x} \; F(\mathbf{x},\mathbf{y}) = c^{11}x^1 + c^{12}x^2 + d^{11}y^1 + d^{12}y^2 \tag{1a}$$

subject to

$$\max_{y} \; f(\mathbf{y}) = d^{21}y^1 + d^{22}y^2 \tag{1b}$$

subject to

$$g(\mathbf{x},\mathbf{y}) = A^1x^1 + A^2x^2 + B^1y^1 + B^2y^2 \leq b \tag{1c}$$

$$\mathbf{x} \geq 0, \quad \mathbf{y} \geq 0 \quad x^2, y^2 \text{ integer.} \tag{1d}$$

Definition 1 *For $\boldsymbol{x} \geq 0$, x^2 integer, let $\Omega(\boldsymbol{x}) = \{\boldsymbol{y} : \boldsymbol{y} \geq 0, y^2 \text{ integer}, \ g(\boldsymbol{x},\boldsymbol{y}) \leq b\}$ and $M(\boldsymbol{x}) = \{\boldsymbol{y} : arg\max\{\mathrm{f}(\boldsymbol{y}') : \boldsymbol{y}' \in \Omega(\boldsymbol{x})\}\}$.*

Definition 2 *If $\bar{\boldsymbol{y}} \in M(\bar{\boldsymbol{x}})$ then $\bar{\boldsymbol{y}}$ is said to be optimal with respect to $\bar{\boldsymbol{x}}$; such a pair is said to be* bi-level feasible.

Definition 3 *A point $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is said to be an optimal solution to the BLPP if*

a. $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is bi-level feasible ; and,

b. for all bi-level feasible pairs $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}), F(\boldsymbol{x}^*, \boldsymbol{y}^*) \geq F(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$.

The three normal fathoming rules when using a branch and bound method for a general mixed integer programming problem are:

Rule 1 *The relaxed subproblem has no feasible solution.*

Rule 2 *The solution of the relaxed subproblem is no greater than the value of the current best feasible solution.*

Rule 3 *The solution of the relaxed subproblem is feasible to the original problem.*

Bard and Falk [1] showed that when solving the BLPP only rule 1 can be applied. They produced counter examples to show that rules 2 and 3 do not always apply when solving the (mixed) integer BLPP. Thus a new branch and bound strategy needed to be developed to solve the (mixed) integer BLPP.

In this paper, we consider the following (purely integer) binary bilevel linear programming problem.

$$\max_{x^1} \ f_1(x^1, x^2) = c^{11}x^1 + c^{12}x^2 \tag{2a}$$

subject to

$$\max_{x^2} \; f_2(x^1, x^2) = c^{21}x^1 + c^{22}x^2 \tag{2b}$$

subject to

$$A^1x^1 + A^2x^2 \leq b^2 \tag{2c}$$

$$x^1 = \{x^1_1, x^1_2, \ldots, x^1_{n1}\} \tag{2d}$$

$$x^2 = \{x^2_1, x^2_2, \ldots, x^2_{n2}\} \tag{2e}$$

$$x^1_j \in \{0, 1\}, \quad j = 1, 2, \ldots, n1 \tag{2f}$$

$$x^2_i \in \{0, 1\}, \quad i = 1, 2, \ldots, n2. \tag{2g}$$

Note that the leader's variables are chosen first and thus become a constant in the followers objective function and do not affect its optimization. In 1990, Wen and Yang [3] developed a branch and bound algorithm to solve the mixed integer binary bi-level linear programming problem. In their formulation, the followers variables were not required to be binary but were continuous variables only restricted to be non-negative. Their algorithm used upper and lower bounds to prune off the tree but did not use upper bounds to decide which were the the most efficient branches to chose in the tree. The algorithm proposed in this paper generalizes many of the ideas originally proposed by Wen and Yang and will impose a preferential choice on which branch to take in the tree based on calculating upper bounds for the leader's objective function.

Wen and Yang utilized a particular notation for the value of the leader's variables at any level in the branch and bound tree. This notation will also be used in this paper.

This notation is defined as follows:

k: the order number of the current node in the branch-and-bound tree:

$J^0_k = \{j | x^1_j$ is a free binary variable, $j = 1, 2, \ldots, n1\}$;

$J^+_k = \{j | x^1_j$ is fixed at 1, $j = 1, 2, \ldots, n1\}$;

$J_k^- = \{j | x_j^1 \text{ is fixed at } 0,\ j = 1, 2, \ldots, n1\}$;

This allows for the formulation of the binary problem (TP_f) in terms of fixing of some of the leader's variables as follows:

$$(TP_f): \quad \max_{x^1} \; f_1 = \sum_{j \in J_k^0} c_j^{11} x_j^1 + \sum_{j \in J_k^+} c_j^{11} + \sum_{i=1}^{n2} c_i^{12} x_i^2 \tag{3a}$$

subject to

$$\max_{x^2} \; f_2 = \sum_{j \in J_k^0} c_j^{21} x_j^1 + \sum_{j \in J_k^+} c_j^{21} + \sum_{i=1}^{n2} c_i^{22} x_i^2 \tag{3b}$$

subject to

$$\sum_{j \in J_k^0} a_j^1 x_j^1 + \sum_{i=1}^{n2} a_i^2 x_i^2 \leq b - \sum_{j \in J_k^+} a_j^1 \tag{3c}$$

$$x_j^1 \in \{0, 1\}, \quad j \in J_k^0 \tag{3d}$$

$$x_i^2 \in \{0, 1\}, \quad i = 1, 2, \ldots, n2 \tag{3e}$$

where a_j^i is the j^{th} column of the matrix, A^i.

Relaxing the TP_f by removing the follower's objective function creates a problem denoted as P_f. It appears, after minor rearrangement, as follows:

$$(P_f): \quad \max_{x^1} \; g = \sum_{j \in J_k^0} c_j^{11} x_j^1 + \sum_{j \in J_k^+} c_j^{11} + \sum_{i=1}^{n2} c_i^{12} x_i^2 \tag{4a}$$

subject to

$$\sum_{i=1}^{n2} a_i^2 x_i^2 \leq b - \sum_{j \in J_k^+} a_j^1 - \sum_{j \in J_k^0} a_j^1 x_j^1 \tag{4b}$$

$$x_j^1 \in \{0, 1\}, \quad j \in J_k^0 \tag{4c}$$

$$x_i^2 \in \{0, 1\}, \quad i = 1, 2, \ldots, n2 \tag{4d}$$

3 BOUNDING THEOREM AND LEMMA

In their paper Wen and Yang proved the following:

Lemma 1 *[3]*

Given two linear programming problems:

$$\begin{aligned}(P): \quad \max \quad & Z = \sum_{j=1}^{n} c_j x_j \\ st: \quad & \sum_{j=1}^{n} a_j x_j \leq b \\ & x_j \geq 0, \quad j = 1, 2, \ldots, n\end{aligned}$$

and

$$\begin{aligned}(P^1): \quad \max \quad & Z^1 = \sum_{j=1}^{n} c_j x_j \\ st: \quad & \sum_{j=1}^{n} a_j x_j \leq b + \theta \\ & x_j \geq 0, \quad j = 1, 2, \ldots, n\end{aligned}$$

where θ is a $m \times 1$ parameter vector.

Then, if

Z^* is the optimal objective value of P,

Y^* is a $1 \times m$ vector, denoting the dual optimal solution of P,

Z^{1*} is the optimal objective value of P^1,

then $Z^{1*} \leq Z^* + Y^*\theta$.

Theorem 1 *Consider the following problem denoted problem B:*

$$\begin{aligned}(B): \quad \max \quad & Z_B = \sum_{i=1}^{n2} c_i^{12} x_i^2 \\ st: \quad & \sum_{i=1}^{n2} a_i^2 x_i^2 \leq b\end{aligned}$$

$$x_i^2 \geq 0, \quad x_i^2 \leq 1 \quad i = 1, 2, \ldots, n2$$

Let Z_B^* be the optimal objective function value for problem B above. Also let Y^* be the optimal dual solution of problem B. Then an upper bound, Z^U, is established for the leader's objective function value in problem TP_f where:

$$Z^U = Z_B^* + \sum_{j \in J_k^+} (c_j^{11} - Y_B^* a_j^1) + \sum_{j \in J_k^0} \max\{(c_j^{11} - Y_B^* a_j^1), 0\} \tag{5}$$

That is $f_1^* \leq Z^U$.

Proof: Relax problem P_f by replacing the constraint

$x_i^2 \in \{0, 1\}, \; i = 1, 2, \ldots, n2$ with $x_i^2 \leq 1, \; x_i^2 \geq 0, \; i = 1, 2, \ldots, n2.$

This relaxation produces a problem we denote as LP_f.

$$(LP_f) : \max \quad g = \sum_{i=1}^{n2} c_i^{12} x_i^2 + \overbrace{\sum_{j \in J_k^0} c_j^{11} x_j^1 + \sum_{j \in J_k^+} c_j^{11}}^{K}$$

$$st: \quad \sum_{i=1}^{n2} a_i^2 x_i^2 \leq b - \overbrace{\sum_{j \in J_k^+} a_j^1 - \sum_{j \in J_k^0} a_j^1 x_j^1}^{\theta}$$

$$x_j^1 \in \{0, 1\}, \quad j \in J_k^0$$

$$x_i^2 \leq 1, \quad x_i^2 \geq 0 \quad i = 1, 2, \ldots, n2.$$

Let g^* be the optimal objective function of the above LP_f with optimal values $\{x_j^{1*}\}$ of the variables $\{x_j^1\}$. Then we obtain the following linear programming problem.

$$(LP_f^{'}) : \max \quad g = \sum_{i=1}^{n2} c_i^{12} x_i^2 + K^{'}$$

$$st: \quad \sum_{i=1}^{n2} a_i^2 x_i^2 \leq b - \theta^{'}$$

$$x_i^2 \leq 1, \quad x_i^2 \geq 0 \quad i = 1, 2, \ldots, n2$$

where $K^{'}$ is a constant determined by evaluating $\sum_{j\in J_k^0} c_j^{11}x_j^1 + \sum_{j\in J_k^+} c_j^{11}$ using the values $\{x_j^{1*}\}$ and $\theta^{'}$ is similarly a constant calculated from $\sum_{j\in J_k^+} a_j^1 + \sum_{j\in J_k^0} a_j^1 x_j^1$ once again using $\{x_j^{1*}\}$.

Then by applying Lemma 1 to $LP_f{}'$ and B:

$$
\begin{aligned}
g^* - K^{'} &\le Z_B^* - Y_B^*\theta^{'} \\
g^* &\le Z_B * + \sum_{j\in J_k^0} c_j^{11}x_j^{1*} + \sum_{j\in J_k^+} c_j^{11} - Y_B^*(\sum_{j\in J_k^+} a_j^1 + \sum_{j\in J_k^0} a_j^1 x_j^{1*}) \\
&= Z_B^* + \sum_{j\in J_k^+}(c_j^{11} - Y_B^* a_j^1) + \sum_{j\in J_k^0}(c_j^{11} - Y_B^* a_j^1)x_j^{1*} \\
&\le Z_B^* + \sum_{j\in J_k^+}(c_j^{11} - Y_B^* a_j^1) + \sum_{j\in J_k^0}\max\{c_j^{11} - Y_B^* a_j^1, 0\}
\end{aligned}
$$

Hence $g^* \le Z^U$.

But since problem P_f is less constrained than problem TP_f, $f_1^* \le g^*$ and so also $f_1^* \le Z^U$.

■

4 ALGORITHM

In the algorithm, N is the current level in the tree, k is the counter for evaluated nodes and

$$
T_j = \begin{cases} 0 & \text{if both branches from the current node at level } j \text{ have not been examined} \\ 1 & \text{if one branch from the current node at level } j \text{ has been examined} \\ 2 & \text{if both branches from the current node at level } j \text{ have been examined} \end{cases}
$$

Step 1 *Initialization*

$$N = 0, k = 0$$

$$J_k^0 = \{1, 2, \ldots, n1\}, J_k^+ = J_k^- = \emptyset$$

$$T_j = 0, \;\; j = 1, 2, \ldots, n1.$$

This indicates that all the leader's variables are free.

Solve problem F:

$$\begin{aligned}(F:)\ \max\ &\sum_{j=1}^{n2} c_j^{22} x_j^2\\ st:\ &\sum_{j=1}^{n2} a_j^2 x_j^2 \leq b\\ &x_j^2 \in \{0,1\}\ \ j=1,2,\ldots,n2\end{aligned}$$

Let the optimal solution be x^{2*} *and let* $x^{1*} = (\overbrace{0,0,0,\ldots,0}^{n1})$ *and let* Z^* *be the value of the leader's objective function* f_1^**, evaluated using the values* x^{1*} *and* x^{2*}*.*

Step 2 *Solve problem B:*

$$\begin{aligned}(B):\ \ \max\ \ & Z_B = \sum_{j=1}^{n2} c_j^{12} x_j^2\\ st:\ \ &\sum_{j=1}^{n2} a_j^2 x_j^2 \leq b\\ &x_j^2 \geq 0,\quad x_j^2 \leq 1\quad j=1,2,\ldots,n2\end{aligned}$$

This problem results in Z_B^**, the optimal objective function value and* Y_B^**, the optimal dual solution. Calculate* $H(j) = c_j^{11} - Y_B^* a_j^1,\ j=1,2,\ldots,n1.$

In order to maximize the efficiency of branching, we establish an upper bound on the path chosen in the tree. This is the function of step 3, the branching step. The value Z_N^U is determined by finding the largest of Z_N^{U+}, the upper bound on the objective function when setting $x_N^1 = 1$ and Z_N^{U-}, the upper bound on the objective function when setting $x_N^1 = 0$.

Step 3 *Branching*

From Theorem 1, $Z^U = Z_B^* + \sum_{j\in J_k^+} H(j) + \sum_{j\in J_k^0} \max\{H(j),0\}.$

Let $S = \sum_{j\in J_k^+} H(j)$ *and let* $W = \sum_{j\in J_k^0} \max\{H(j),0\}$

and $N = N+1 \quad k = k+1.$

Calculate the upper bound of the leader's objective function Z_N^U *for the branch down-tree for* $x_N^1 = 0$*. This will be denoted as* Z_N^{U-} *.*

Loop 1: *Set* $S = W = 0$

For $i = 1$ *to* $n1$

If $x_i^1 = 1$

$S = S + H(i)$

else

if $x_i^1 = 0$

$W = W + \max\{H(i), 0\}$

else

end for loop

end loop

$Z_N^{U-} = Z_B^* + S + W$

Calculate Z_N^{U+}*, the branch where* $x_N^1 = 1$*.*

This is achieved in a very similar manner to the calculation of Z_N^{U-}*. Set* $x_N^1 = 1$ *then execute Loop 1 and finally* $Z_N^{U+} = Z_B^* + S + W$*.*

If $Z_N^{U+} \geq Z_N^{U-}$ *then the upper bound* $Z^U = Z_N^{U+}$*,* $x_N^1 = 1$ *and* $T_N = T_N + 1$*, otherwise* $Z^U = Z_N^{U-}$*,* $x_N^1 = 0$ *and again* $T_N = T_N + 1$

Of course, the upper bounds need to be checked against the current best solution of the objective function, Z^*. If the upper bound on that particular branch is not greater than the current best solution then that branch is fathomed.

Step 4 *Fathoming Check*

If $Z^U \leq Z^*$ *then set* $T_N = 2$*, go to Step 6.*

else go to Step 5.

The next step checks if $N = n1$, that is if all the leader's variables have been assigned a value. Then the follower's problem, L, with the given values of the leader's variables and if feasible the solution, Z^L, is compared to the current best solution.

Step 5 *Calculate Feasible Solutions*

If $N \neq n1$ *then go to Step 3*

else solve problem L

where

$$\text{(L:)} \max \sum_{j=1}^{n2} c_j^{22} x_j^2 + \sum_{j \in J_k^+} c_j^{11}$$

$$\text{st:} \sum_{j=1}^{n2} a_j^2 x_j^2 \leq b - \sum_{j \in J_k^+} a_j^2$$

$$x_j^2 \in \{0, 1\} \; j = 1, 2, \ldots, n2.$$

Let the current values of the leader's variables be x^{1L} *and the solution of problem L be* x^{2L}
Let Z^L *be the leader's objective function value evaluated using* x^{2L} *and* x^{1L}.
If $Z^L > Z^*$ *AND problem L is feasible,*

then update Z^*, x^{2*} *and* x^{1*} *from* Z^L, x^{2L} *and* x^{1L} *respectively , go to Step 6;*

else go to Step 6.

The algorithm now proceeds back up the tree, examining branches and their upper bounds. Each upper bound is compared to the current best solution to determine whether the branch can be fathomed or must be considered further. This is performed in the next step.

Step 6 *Backtracking*

$$\begin{aligned}
&\textit{If } T_N = 2 \textit{ then set}\\
&\quad T_N = 0,\ x_N^1 = 0, N = N - 1\\
&\quad \textit{If } N = 0 \textit{ then go to Step 7.}\\
&\quad \textit{else go to Step 6;}\\
&\textit{else } T_N = T_N + 1.\\
&\textit{If } x_N^1 = 0 \textit{ then } Z^U = Z^{U+},\ x_N^1 = 1,\ \textit{go to Step 4.}\\
&\textit{else } Z^U = Z_N^{U-},\ x_N^1 = 0 \textit{ go to Step 4.}
\end{aligned}$$

All that remains is to terminate the process at the point where all viable branches and leaves have been utilized.

Step 7 *Terminate*

Stop execution of the algorithm and output the solution.

5 COMPUTATIONAL RESULTS

To evaluate the results of the algorithm it was coded into a SAS program and bi-level problems were constructed randomly using the following guidelines.

The leader's objective function coefficients were chosen randomly between the limits of -30 to +30. The follower's objective function coefficients were chosen randomly between -12 and +12. The constraint matrix coefficients were chosen randomly between -18 and +18 and the b_j, or resource values were restricted to be within the range 0.5 to 0.75 of the sum of the a_j for the j^{th} constraint. This was to insure a high probability of feasibility.

In both table 1 and table 2 the column headers represent

evaluated nodes = number of nodes where an upper bound was established as a percentage of total nodes in the tree.

lpcalls = number of LP problems solved as a percentage of leaves in the tree.

kstar = the node number where the optimal solution was obtained as a percentage of nodes in the tree.

In Table 1, 10 randomly constructed problems were solved for each problem type.

n1	n2	evaluated nodes	total nodes	lpcalls	leaves	kstar
5	5	55%	62	39%	32	27%
5	8	72%	62	62%	32	36%
5	10	75%	62	73%	32	34%
8	5	37%	510	23%	256	13%
8	8	43%	510	31%	256	25%
8	10	64%	510	58%	256	30%
10	5	15%	2046	7%	1024	4%
10	10	51%	2046	41%	1024	11%

Table 1: Results of 10 samples for each $n1 \times n2$ problems

The results in Table 2 are from randomly constructing 100 problems for each problem type. This set of computations was mainly performed to check the statistical validity of the results in Table 1.

Several conclusions may be drawn from these computational results. In their paper Wen and Yang [3] suggested in their conclusions that there may well be a correlation between the effectiveness of their bounding function and the ratio of the number of leader's variables, $n1$, to the number of follower's variables, $n2$. The results seem to confirm this. It is apparent from the tables that when the numbers of both leader's and follower's variables are similar both the evaluated nodes percentage and the kstar percentage figures are larger. However if $n1 > n2$ then both these percentages, which measure the effectiveness of the bounding

n1	n2	evaluated nodes	total nodes	lpcalls	leaves	kstar
5	5	51%	62	36%	32	30%
5	8	69%	62	57%	32	35%
5	10	71%	62	64%	32	33%
8	5	23%	510	14%	256	12%
8	8	49%	510	37%	256	17%
8	10	57%	510	45%	256	20%
10	5	13%	2046	8%	1024	7%
10	10	52%	2046	42%	1024	21%

Table 2: Results of 100 samples for each $n1 \times n2$ problems

function at finding a tight upper bound for the optimal feasible solution, are significantly lower. In the case of the 10×5 problem these values drop to very low levels indicating excellent performance by both the bounding function and the algorithm in general. On the other hand it would seem that if $n2 > n1$ the effectiveness deteriorates giving the highest percentages.

Examining the performance of the algorithm and the use of the upper bounds at each level in the tree to choose branching means examining kstar. It would appear from the low values of kstar, all lower than 35%, that the addition of controlling the decision by utilizing the Z_N^U was an effective measure in tightening the bounds in the solution of the problem.

References

[1] Bard, J. and Falk, J, "The mixed interger linear bi-level programming problem," *Operations Research 38, 911-921, 1989.*

[2] Bialas, W. and Karwan, M., "Two-level linear programing," Mgmt Sci. 30, 1004-1020, 1984.

[3] Wen, U. and Yang, Y., "Algorithms for solving the mixed integer two-level programming problem," Computers Ops Res, 17, 133-142, 1990.