Computers Opns Res. Vol. 17, No. 2, pp. 133–142, 1990
Printed in Great Britain.

0305–0548/90 $3.00 + 0.00

ALGORITHMS FOR SOLVING THE MIXED INTEGER TWO-LEVEL LINEAR PROGRAMMING PROBLEM

U. P. WEN*,† and Y. H. YANG‡
Department of Industrial Engineering, National Tsing Hua University, Kuang Fu Road, Hsinchu, Taiwan, Republic of China

(*Received September* 1988; *revised July* 1989)

Scope and Purpose—Decentralized planning has long been recognized as an important decision making problem. Multilevel programming models partition control over decision variables. In such a hierarchical structure system, the high-level decision making situations often require the inclusion of zero–one variables which represent 'yes–no' decisions. In this paper, we combine the zero–one decision variables controlled by the high-level decision maker with the real-valued decision variables controlled by the low-level decision maker into the two-level programming model and formulate them as a mixed integer two-level linear programming problem. Both the exact and heuristic solution procedures are available.

Abstract—Several algorithms have been developed to solve the two-level linear programming problem during the past years. In this paper, we will formulate the mixed integer two-level linear programming problem and develop both the exact and heuristic solution procedures based on the branch-and-bound technique for solving the problem. Computational experience and comparisons will be presented.

1. INTRODUCTION

Many resource planning problems require compromises among the objectives of several interacting planners in a hierarchical system. Such a decision making problem can be found under the heading of multiple criteria decision making. Much of the research work in this important area has occurred since the 1970s [1].

A number of techniques that have proved successful in modeling the individual preferences and interactions that prevail in such systems include multiobjective mathematical programming [2] and goal programming [3]. These approaches explicitly consider all competing goals and then seek a compromise or efficient solution. The shortcoming, however, is that the decision is assumed to be made by one person or a homogeneous group. Multilevel programming finds its rationale in the need for a more accurate way of representing the situation in which a sequence of decisions are to be made by a group of independent decision makers. A special case, the two-level linear form, is widely discussed in the literature [4–9].

In such a problem, the decision variables are partitioned between the high-level and low-level decision makers, each of whom optimizes his objective function. Furthermore, the decision variables in the previous two-level linear model are restricted to continuous variables.

However, in many decentralized decision making systems, the high-level decision makers face a 'yes–no' decision problem. Therefore, we should combine the integer linear version of the program with binary restrictions on the high-level decision variables into the continuous decision variables controlled by the low-level as a mixed integer two-level linear programming problem (MITLP). In this paper, we will develop an exact algorithm and a heuristic solution procedure to solve the MITLP. In the next section we formulate the MITLP and give the notation. Design of bounds are introduced in Section 3. In Section 4, two algorithms are proposed. Computational results are provided in Section 5. Section 6 contains some concluding remarks.

*Address correspondence to Dr U. P. Wen, Department of Management Science and Statistics, College of Business & Management, University of Maryland, College Park, MD 20742, U.S.A.
†U. P. Wen is Associate Professor of Industrial Engineering, National Tsing Hua University, Hsinchu, Taiwan, R.O.C. He received his PhD in industrial engineering from State University of New York at Buffalo. His primary research interests are in the area of multicriteria decision making, automatic warehousing systems and applied operations research. Dr Wen is currently a Visiting Scholar at the University of Maryland (1989–1990).
‡Y. H. Yang earned his MS in industrial engineering from National Tsing Hua University, Taiwan, R.O.C. Presently, he is working in the Military Service.

2. PROBLEM FORMULATION

The mixed integer two-level programming problem can be formulated as follows:

$$
\begin{aligned}
\text{(MITLP):}\quad \max\ & \{f_1(x^1, x^2) = c^{11}x^1 + c^{12}x^2 : (x^1)\} \\
\text{st:}\quad \max\ & \{f_2(x^1, x^2) = c^{22}x^2 : (x^2 \mid \hat{x}^1)\} \\
\text{st:}\quad & A^1x^1 + A^2x^2 \leqslant b \\
& x^1 = (x_1^1, x_2^1, \ldots, x_{n1}^1), \\
& x_j^1 \in \{0, 1\}, \quad j = 1, 2, \ldots, n1, \\
& x^2 \geqslant 0
\end{aligned}
$$

where $\max\{f_1(x^1, x^2) = c^{11}x^1 + c^{12}x^2 : (x^1)\}$ denotes the maximum of f_1 over x^1, x^2 but only x^1 can be controlled at the high level problem.
$\max\{f_2(x^1, x^2) = c^{22}x^2 : (x^2 \mid \hat{x}^1)\}$ denotes the maximum of f_2 over x^2 for a fixed value of $\hat{x}^1$.

and x^1 is an $n1 \times 1$ vector of decision variables controlled by the high-level decision maker;
x^2 is an $n2 \times 1$ vector of decision variables controlled by the low-level decision maker;
A^1 is an $m \times n1$ matrix of technological coefficients for the high-level decision variables;
A^2 is an $m \times n2$ matrix of technological coefficients for the low-level decision variables;
c^{11} is a $1 \times n1$ vector of profit coefficients of the high-level decision variables in the high-level objective function f_1;
c^{12} is a $1 \times n2$ vector of profit coefficients of the low-level decision variables in the high-level objective function f_1;
c^{22} is a $1 \times n1$ vector of profit coefficients of the low-level decision variables in the low-level objective function f_2;
b is an $m \times 1$ vector of resource capacity of the system.

For the convenience of analysis, we will refer to the following notations throughout the development, which are

$$S = \{(x^1, x^2) \mid A^1x^1 + A^2x^2 \leqslant b,\ x_j^1 \in \{0, 1\},\ j = 1, 2, \ldots, n1;\ x^2 \geqslant 0\}$$

and

$$W_{f_2}(S) = \{(\hat{x}^1, \hat{x}^2) \in S \mid f_2(\hat{x}^1, \hat{x}^2) = \max\{f_2(x^1, x^2) : (x^2 \mid \hat{x}^1)\},$$

where $W_{f_2}(S)$ represents the feasible region of the high-level decision maker. Without loss of generality, we will assume A^2 and $b \geqslant 0$, S is nonempty and the optimal solution of MITLP is nondegenerate. Furthermore, we use the following definitions to describe the feasibility and optimality.

Definition 1

A point $(\bar{x}^1, \bar{x}^2)$ is said to be feasible to the MITLP if $(\bar{x}^1, \bar{x}^2) \in W_{f_2}(S)$.

Definition 2

A point (x^{1*}, x^{2*}) is said to be an optimal solution to the MITLP if

(a) (x^{1*}, x^{2*}) is feasible; and
(b) for all feasible points $(\bar{x}^1, \bar{x}^2) \in S$, $c^{11}x^{1*} + c^{12}x^{2*} \geqslant c^{11}\bar{x}^1 + c^{12}\bar{x}^2$.

Most of the existing solution techniques for the two-level linear programming problem replace the low-level problem by its Kuhn–Tucker conditions [4, 5]. However, limited computational experienced is reported, and it is anticipated that only small problems could be solved with reasonable computational time. Therefore, we will concentrate on the original form of the MITLP in this paper.

3. DESIGN OF BOUNDS

Many branch-and-bound procedures for integer programming employ linear programming to obtain bound information [10–12].

For the maximization form of MITLP, the lower bound on the optimal value of the high-level objective function can be determined as the greatest value of the high-level objective function found so far. On the other hand, it is trivial that an upper bound is the optimal objective value of MITLP by neglecting the low-level objective function. However, the resulting problem, a mixed zero–one linear case, is still difficult to solve. This leads us to search for a better bound. In order to develop the bounding function, we adopt the following notations:

k: the order number of generated node in a branch-and-bound tree;
$J_k^0 = \{j \mid x_j^1$ is free binary variable, $j = 1, 2, \ldots, n1\}$;
$J_k^+ = \{j \mid x_j^1$ is fixed at 1, $j = 1, 2, \ldots, n1\}$;
$J_k^- = \{j \mid x_j^1$ is fixed at 0, $j = 1, 2, \ldots, n1\}$;

and have the following lemma:

Lemma 1

Given two linear programming problems

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

and

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

where

a_j is the jth column vector of the $m \times n$ matrix, A;
θ is an $m \times 1$ 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; and
Y^{1*} is a $1 \times m$ vector, denoting the dual optimal solution of P^1,
then $Z^{1*} \leqslant Z^* + Y^*\theta$.

Proof. Since Y^* is the dual optimal solution of P and is also a dual feasible solution of P^1, we must have

$$Z^{1*} = Y^{1*} \cdot (b + \theta) \leqslant Y^* \cdot (b + \theta) = Y^*b + Y^*\theta = Z^* + Y^*\theta. \qquad \square$$

For the MITLP, if part of the high-level decision variables are fixed, then the resulting problem is as follows:

$$\begin{aligned} (\mathrm{MITLP}_f): \quad \max \quad & f_1 = \sum_{j \in J_k^0} c_j^{11} x_j^1 + \sum_{j \in J_k^+} c_j^{11} + \sum_{j=1}^{n2} c_j^{12} x_j^2 \\ \text{st:} \quad \max \quad & f_2 = \sum_{j=1}^{n2} c_j^{22} x_j^2 \\ \text{st:} \quad & \sum_{j \in J_k^0} a_j^1 x_j^1 + \sum_{j=1}^{n2} a_j^2 x_j^2 \leqslant b - \sum_{j \in J_k^+} \mathrm{a}_j^1 \\ & x_j^1 \in \{0, 1\}, \quad j \in J_k^0 \\ & x_j^2 \geqslant 0, \quad j = 1, 2, \ldots, n2, \end{aligned}$$

where a_j^1 is the jth column vector of A^1,
a_j^2 is the jth column vector of A^2.

On the other hand, if we neglect the low-level objective function, f_2 of MITLP$_f$, the resulting problem becomes the following mixed integer linear programming problem:

$$\begin{aligned}(\text{MILP}_f):\quad \max\quad & g = \sum_{j\in J_k^0} c_j^{11}x_j^1 + \sum_{j\in J_k^+} c_j^{11} + \sum_{j=1}^{n2} c_j^{12}x_j^2 \\ \text{st:}\quad & \sum_{j\in J_k^0} a_j^1 x_j^1 + \sum_{j=1}^{n2} a_j^2 x_j^2 \leqslant b - \sum_{j\in J_k^+} a_j^1 \\ & x_j^1 \in \{0,1\}, \quad j\in J_k^0 \\ & x_j^2 \geqslant 0, \quad j = 1,2,\ldots,n2.\end{aligned}$$

Let f_1^* be the high-level optimal objective value of MITLP$_f$, g^* be the optimal objective value of MILP$_f$, and we have the following results:

Lemma 2

The high-level optimal objective value of MITLP$_f$ is less than or equal to the optimal objective value of MILP$_f$.

Proof. The solution space of MITLP$_f$ is contained in that of MILP$_f$ and the high-level objective function of MITLP$_f$ is the same as that of MILP$_f$. Therefore, $f_1^* \leqslant g^*$. □

The following theorem will be contributive to the design of the bounding function.

Theorem 1

Consider the following problem:

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

Then, if Z_B^* is the optimal objective value of (B) and Y_B^* is the dual optimal solution of (B), then the high-level optimal objective value of MITLP$_f$ is less than or equal to the value Z^U, where

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

i.e. Z^U is an upper bound of (MITLP$_f$).

Proof. Rearrange (MILP$_f$) as follows:

$$\begin{aligned}\max\quad & g = \sum_{j=1}^{n2} c_j^{12}x_j^2 + \sum_{j\in J_k^0} c_j^{11}x_j^1 + \sum_{j\in J_k^+} c_j^{11} \\ \text{st:}\quad & \sum_{j=1}^{n2} a_j^2 x_j^2 \leqslant b - \sum_{j\in J_k^+} a_j^1 - \sum_{j\in J_k^0} a_j^1 x_j^1 \\ & x_j^1 \in \{0,1\}, \quad j\in J_k^0, \\ & x_j^2 \geqslant 0, \quad j = 1,2,\ldots,n2.\end{aligned}$$

Then, the following two inequalities hold from previous results.

$$g^* \leqslant Z_B^* + \sum_{j \in J_k^0} c_j^{11} x_j^1 + \sum_{j \in J_k^+} c_j^{11} - Y_B^* \left(\sum_{j \in J_k^+} a_j^1 + \sum_{j \in J_k^0} a_j^1 x_j^1 \right) \quad \text{(1) [by Lemma 1]}$$

$$= Z_B^* + \sum_{j \in J_k^+} (c_j^{11} - Y_B^* a_j^1) + \sum_{j \in J_k^0} (c_j^{11} - Y_B^* a_j^1) x_j^1$$

$$\leqslant Z_B^* + \sum_{j \in J_k^+} (c_j^{11} - Y_B^* a_j^1) + \sum_{j \in J_k^0} \max\{c_j^{11} - Y_B^* a_j^1, 0\}$$

$f_1^* \leqslant g^*$. (2) [by Lemma 2]

By (1) and (2), we conclude that Z^U is an upper bound of (MITLP$_f$). □

4. PROPOSED ALGORITHMS

Exact algorithm

Based on the Theorem 1, the upper bound of node k in the tree can be found by solving the problem (B). The branching procedure always chose the first free variable to branch upon by first setting the corresponding binary variable to zero and then setting it to one. There are two important reasons for this: one is to reduce the number of performing dual simplex pivot operations, the other is to save the memory storage space. Especially, the first reason will greatly affect the efficiency for solving the MITLP.

The exact algorithm can be described as follows:

Step 1a: (INITIALIZATION) Set $N = 0$, $k = 0$, $J_k^0 = \{1, 2, \ldots, n1\}$, $J_k^+ = J_k^- = \phi$, and solve the following problem:

$$\text{(F):} \quad \max \sum_{j=1}^{n2} c_j^{22} x_j^2$$

$$\text{st:} \quad \sum_{j=1}^{n2} a_j^2 x_j^2 \leqslant b$$

$$x_j^2 \geqslant 0, \quad j = 1, 2, \ldots, n2$$

with the optimal solution x^{2*} and objective value Z^*.

Step 1b: Solve the problem (B) with the optimal objective value Z_B^* and the dual optimal solution Y_B^*. Calculate $H(j) = c_j^{11} - Y_B^* a_j^1$, $j = 1, 2, \ldots, n1$.

Step 2: (BRANCHING) Set $N = N + 1$, $J_k^0 = J_k^0 \setminus \{N\}$, $J_k^- = J_k^- \cup \{N\}$, $k = k + 1$.

Step 3: (CALCULATING BOUND) $Z^U = Z_B^* + \sum_{j \in J_k^+} H(j) + \sum_{j \in J_k^0} \max\{H(j), 0\}$.

Step 4a: (FATHOMING) If $Z^U \leqslant Z^*$, go to Step 5a; otherwise, go to Step 4b.

Step 4b: Check if $N = n1 - 1$, set $N = N + 1$, $J_k^0 = J_k^0 \setminus \{N\}$, $J_k^- = J_k^- \cup \{N\}$, $k = k + 1$ and go to Step 6a; otherwise, go to Step 2.

Step 5a: (BACKTRACKING) If $N \in J_k^-$, set $J_k^+ = J_k^+ \cup \{N\}$, $J_k^- = J_k^- \setminus \{N\}$, $k = k + 1$ and go to Step 3; otherwise, go to Step 5b.

Step 5b: Set $J_k^+ = J_k^+ \setminus \{N\}$, $J_k^0 = J_k^0 \cup \{N\}$, $N = N - 1$. If $N = 0$ (no live node exists), go to Step 7; otherwise, go to Step 5a.

Step 6a: (CALCULATING FEASIBLE SOLUTION) Solve the following problem (L):

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

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

$$x_j^2 \geqslant 0, \quad j = 1, 2, \ldots, n2.$$

Let x^{2L} be the optimal solution of (L) and set

$$Z^L = \sum_{j=1}^{n2} c_j^{12} x_j^{2L} + \sum_{j \in J_k^+} c_j^{11}.$$

Step 6b: If $Z^L > Z^*$, update $(x^{1*}; x^{2*})$ and Z^* by $(x^{1L}; x^{2L})$ and Z^L, respectively. Go to Step 6c.

Step 6c: If $N \in J_k^-$, set $J_k^+ = J_k^+ \cup \{N\}$, $J_k^- = J_k^- \setminus \{N\}$, $k = k + 1$ and go to Step 6a; otherwise, go to Step 5b.

Step 7: (TERMINATION) Stop; $(x^{1*}; x^{2*})$ is the optimal solution with the high-level optimal objective value Z^*.

Note that Step 1a is to find an initial feasible solution of MITLP by setting all high-level decision variables to zero via the simplex method. If we relax the assumptions A^2 and $b \geqslant 0$, then Step 1a may not find a feasible solution. Therefore, we need to provide a phase I procedure to ensure that the branch could contain a feasible solution. Step 1b is designed to find the basic information about bounds. Branching occurs at Step 2 where J_k is updated. Step 3 is to calculate the upper bound. Step 4 is to check if the upper bound is less than or equal to the current optimal objective value. If true, we may terminate this node; otherwise, it is necessary to move forward to branch. Backtracking procedure is adopted by branching to the newest live node which takes place in Step 5. To find a potentially better feasible solution for MITLP is accomplished in Step 6. Finally, if there exists no live node, the solution procedure will terminate with the current optimal solution in Step 7.

We will give a numerical example to demonstrate the exact algorithm as follows:

Example 1:

$$\begin{aligned}
\max \quad & \{20x_1^1 + 60x_2^1 + 30x_3^1 + 50x_4^1 + 15x_1^2 + 10x_2^2 + 7x_3^2 : (x_1^1, x_2^1, x_3^1, x_4^1)\} \\
\text{st:} \quad & \max \quad \{20x_1^2 + 60x_2^2 + 8x_3^2 : (x_1^2, x_2^2, x_3^2 \mid x_1^1, x_2^1, x_3^1, x_4^1)\} \\
& \text{st:} \quad 5x_1^1 + 10x_2^1 + 30x_3^1 + 5x_4^1 + 8x_1^2 + 2x_2^2 + 3x_3^2 \leqslant 230 \\
& \qquad 20x_1^1 + 5x_2^1 + 10x_3^1 + 10x_4^1 + 4x_1^2 + 3x_2^2 \leqslant 240 \\
& \qquad 5x_1^1 + 5x_2^1 + 10x_3^1 + 5x_4^1 + 2x_1^2 + x_3^2 \leqslant 90 \\
& \qquad x_j^1 \in \{0, 1\}, \quad j = 1, 2, 3, 4. \\
& \qquad x_j^2 \geqslant 0, \quad j = 1, 2, 3.
\end{aligned}$$

Note that $(x^{1*}; x^{2*}) = (0, 1, 0, 1; 0, 75, 21.667)$ is the optimal solution with the high-level optimal objective value $Z^* = 1011.667$, which occurs on node 10. Nodes 4, 5, 9 and 10, on which the high-level variables are all fixed, are obviously terminated; nodes 6, 11 and 12 are fathomed by $Z^U \leqslant Z^*$.

The number of nodes generated is 12, compared with that the the total node number of complete tree is $2^{4+1} - 1 = 31$; the number of needed enumeration is 4, but the total enumeration needs $2^4 = 16$ times. The corresponding branch-and-bound tree is shown in Fig. 1.

Heuristic algorithm

The branch-and-bound technique is often an effective tool for dealing with the problems in mixed zero–one form. Its effectiveness depends on the design of the bound and the selection of branching rules. However, the bound information in exact algorithm is sometimes not effective for solving the (MITLP). In the worst case, the exact algorithm needs almost total enumeration in order to solve the problem. Therefore, when the number of high-level zero–one decision variables grows linearly, the computational time grows exponentially in such a situation.

In this section, we will propose a heuristic solution procedure, which can provide satisfactory near-optimal solutions in a reasonably short computational time. The heuristic algorithm was developed based on a judgment index which will be described as follows:

(1) Define the estimated 'profit' for each high-level zero–one decision variable by neglecting

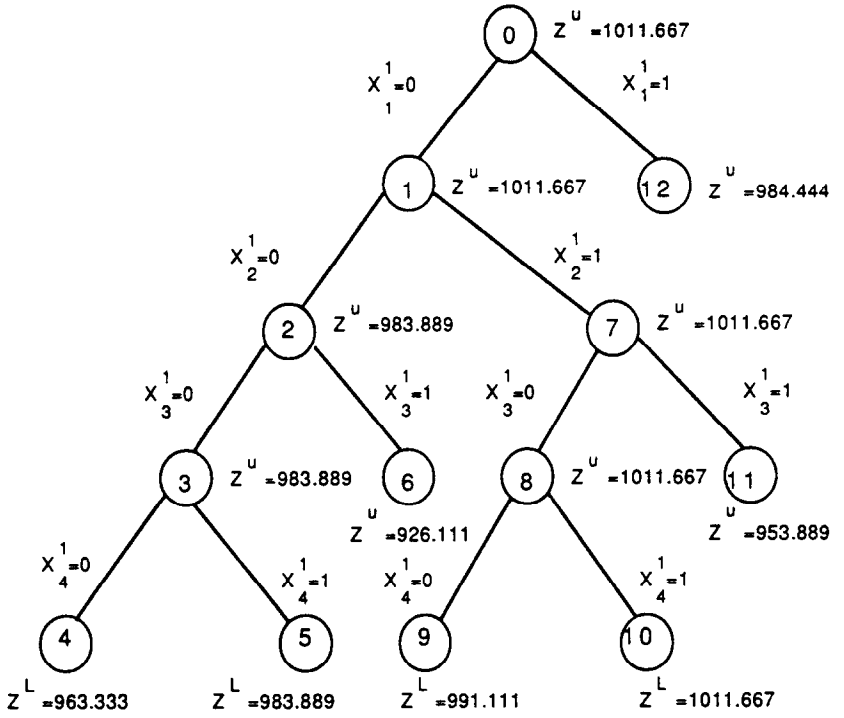

Fig. 1. Branch-and-bound tree for example 1.

the low-level objective function as:

$$H(j) = c_j^{11} - Y_B^* a_j^1, \quad j = 1, 2, \ldots, n1,$$

where Y_B^* is the dual optimal solution of (B).

(2) Define the estimated 'profit' for each high-level zero–one decision variables by neglecting the high-level objective function as:

$$L(j) = c_j^{11} - Y_F^* a_j^1, \quad j = 1, 2, \ldots, n1;$$

where Y_F^* is the dual optimal solution of (F).

(3) Define the judgment index as:

$$I(j) = \left(\frac{n1}{n1+n2}\right)H(j) + \left(\frac{n2}{n1+n2}\right)L(j), \quad j = 1, 2, \ldots, n1,$$

which indicates that the importance (or weight) of each high-level decision variable depends on its 'profit' and the number of variables controlled by itself. The higher the judgment index value of the high-level decision variable, $I(j)$, is, the higher priority we will assign that variable. In addition, the variable with the highest priority will be set to one.

For convenience, we will sort $I(j)$, $j = 1, 2, \ldots, n1$ in descending order and adopt the notation $I_N = \{j \mid$ the index correspondence to the first greatest value of $I(j)$, $j = 1, 2, \ldots, n1\}$. The solution can be described as follows:

Step 1a: (INITIALIZATION) Set $N = 0$, $x_j^1 = 0$, $j = 1, 2, \ldots, n1$ and solve the problem (B) with the dual optimal solution of Y_B^*. Calculate $H(j) = c_j^{11} - Y_B^* a_j^1$, $j = 1, 2, \ldots, n1$.

Step 1b: Solve the problem (F) with the optimal solution x_j^{2*} and objective value Z^*. Calculate $L(j) = c_j^{11} - Y_F^* a_j^1$, $j = 1, 2, \ldots, n1$, where Y_F^* is the dual optimal solution of (F).

Step 1c: Calculate $I(j) = \left(\frac{n1}{n1+n2}\right)H(j) + \left(\frac{n2}{n1+n2}\right)L(j)$, $j = 1, 2, \ldots, n1$, go to Step 2.

Step 2: (SORTING JUDGMENT INDEX) Sort $I(j)$, $j = 1, 2, \ldots, n1$ in descending order and go to Step 3.

Step 3: If $N \leqslant n1$, go to Step 4; otherwise, go to Step 6.

Step 4: Set $N = N + 1$ and solve the following problem (F^1) by sensitivity analysis:

$$(F^1): \quad \max \quad \sum_{j=1}^{n2} c_j^{22} x_j^2$$

$$\text{st:} \quad \sum_{j=1}^{n2} a_j x_j^2 \leqslant b - \sum_{j \in J_N} a_j^1$$

$$x_j^2 \geqslant 0, \quad j = 1, 2, \ldots, n2,$$

with the optimal solution x^{2L}. Set $Z^L = \sum_{j=1}^{n2} c_j^{12} x_j^{2L} + \sum_{j \in I_N} c_j^{11}$.

Step 5: If $Z^L > Z^*$, update $(x^{1*}; x^{2*})$ and Z^* by $(x^{1L}; x^{2L})$ and Z^L, respectively, where $x_j^{1L} = 1$, if $j \in I_N$; $x_j^{1L} = 0$, otherwise. Go to Step 3.

Step 6: (TERMINATION) Stop; $(x^{1*}; x^{2*})$ is the near-optimal solution with the high-level optimal objective value Z^*.

Note that in Step 1a, we calculate the estimated 'profit' of each high-level binary decision variable by neglecting the low-level objective function. Step 1b is designed to obtain an initial solution by fixing all high-level variables at zero, and the estimated 'profit' by neglecting the high-level objective function. We calculate the judgment index in Step 1c. Next, we sort the judgment index in descending order in Step 2. Step 3 is to check if the solution procedure will terminate. If it will not, then we go to Step 4 to assign 1 to the high-level decision variable corresponding to the highest judgment index in the unassigned candidates and calculate a potentially better solution by sensitivity analysis in Step 4. If infeasibility occurs, we must perform dual simplex pivot operations to retain feasibility. Step 5 is to check whether the solution obtained in Step 4 is better than the current optimal solution or not. Termination occurs in Step 6 via the near-optimal solution declaration.

In this heuristic algorithm, the number of iterations is equal to the number of the high-level decision variables plus one. That is, the proposed algorithm can be solved in linear computational time when the number of high-level zero–one decision variables grows linearly. The accuracy is demonstrated by computational experiments as shown in the next section.

5. COMPUTATIONAL RESULTS

Both the exact algorithm and heuristic solution for MITLP were coded in FORTRAN V. Various problems generated at random were solved on a CDC CYBER 840 computer system. The constraint matrix coefficients of tested problems are randomly generated within the range [0, 99]. The constraint matrix has a 75% density of nonzero elements and the right hand side vector is uniformly distributed between 1/4 and 3/4 row sum of the coefficients of system constraints. We consider 15 groups of problems which are the combinations of 5, 8, 10, 12, 15 high-level 0–1 variables and 5, 10, 15 low-level variables. The number of constraints is 10 in each problem and 10 problems were run for each group. Table 1 shows the average execution time of exact algorithm and heuristic

Table 1. Average execution results of algorithms

		*N*1				
*N*2		5	8	10	12	15
5	*E*	0.03	0.14	0.53	1.32	7.61
	H	0.02	0.02	0.02	0.03	0.03
10	*E*	0.05	0.23	0.83	3.02	26.16
	H	0.02	0.03	0.04	0.03	0.05
15	*E*	0.08	0.42	1.25	4.34	30.83
	H	0.03	0.03	0.04	0.04	0.05

*N*1: number of variables controlled by the high level; *N*2: number of variables controlled by the low level; *E*: average execution time in CPU sec for the exact algorithm; *H*: average execution time in CPU sec for the heuristic algorithm.

Table 2. Average enumeration ratio for the exact algorithm

S	$p=\int_{25}^{0}$	$\int_{50}^{26}$	$\int_{75}^{51}$	$\int_{99}^{76}$	100	Average enumeration ratio
5 × 5	$N=$ 3	2	1	0	4	59.3
5 × 10	1	2	4	1	2	58.4
5 × 15	0	0	1	2	7	95.0
8 × 5	5	2	2	0	1	35.2
8 × 10	4	2	1	0	3	47.6
8 × 15	0	2	1	1	6	84.1
10 × 5	6	0	2	2	0	34.3
10 × 10	6	1	0	3	0	38.1
10 × 15	2	3	0	3	2	60.7
12 × 5	8	0	2	0	0	18.5
12 × 10	5	3	0	0	2	34.7
12 × 15	3	1	1	1	4	60.8
15 × 5	7	3	0	0	0	14.8
15 × 10	5	2	1	2	0	34.7
15 × 15	5	0	1	3	1	45.5
Total	60	23	17	18	32	48.1
%	40	16	11	12	21	—

P: enumeration ratio represented by percentage; S: problem size $N1 \times N2$, where $N1$ is number of variables controlled by the high level; $N2$ is number of variables controlled by the low level; N: number of problems.

Table 3. Average accuracy for the heuristic algorithm

S	$p=\int_{84}^{75}$	$\int_{89}^{85}$	$\int_{94}^{90}$	$\int_{99}^{95}$	100	Average accuracy
5 × 5	$N=0$	0	1	0	9	99.2
5 × 10	0	0	1	3	6	97.9
5 × 15	0	0	0	2	8	99.3
8 × 5	0	1	1	3	5	96.8
8 × 10	0	0	0	3	7	99.5
8 × 15	1	0	1	2	6	97.2
10 × 5	0	0	0	3	7	99.5
10 × 10	0	1	0	5	4	98.3
10 × 15	0	1	2	5	2	95.8
12 × 5	0	0	0	8	2	98.7
12 × 10	0	0	1	7	2	98.0
12 × 15	2	1	1	2	4	93.5
15 × 5	2	1	2	4	1	92.0
15 × 10	1	0	0	7	2	96.4
15 × 15	0	0	2	4	4	98.4
Total	6	5	12	58	69	97.4
%	4	3.3	8	38.6	46.1	—

P: percentage of accuracy; S: problem size $N1 \times N2$, where $N1$ is number of variables controlled by the high level; $N2$ is number of variables controlled by the low level; N: number of problems.

algorithm. When the number of high-level variables increases, the execution time of exact algorithm increases approximately exponentially; the fewer low-level variables there are, the slower the execution time grows. Furthermore, the execution time with a heuristic algorithm is much less than that with an exact algorithm.

The enumeration ratio, which is the number of needed enumeration divided by the number of total enumeration, for 150 problems by exact algorithm is listed in Table 2. If the number of high-level variables and the ratio of $N1$ to $N2$ are high, then the average enumeration ratio is relatively low (for example: the problem groups 12 × 5, 15 × 5). This indicates that the bounding function became more effective as the percentage of variables controlled by the high-level decision maker was increased. The percentage of problems with enumeration ratio under 25% is up to 40% and with total enumeration is 21%. Furthermore, the average enumeration ratio for all problems is 48.1% by the exact algorithm.

Table 3 shows the accuracy of the heuristic algorithm. The average optimality obtained is up to 97.4% and the percentage of the problems which obtain 100% optimality is up to 46.1%. This demonstrates that the heuristic algorithm provides a good approximation to find an optimal or near-optimal solution.

6. CONCLUSION

This paper has presented an exact algorithm and a heuristic solution procedure for solving the mixed integer two-level linear programming problem. The branch-and-bound technique is applied to tackle this class of problems. We designed a bounding function for the exact algorithm which is based on two concepts; one is to neglect the low-level objective function, the other is to consider how it affects the high-level objective value by completing the assignment of binary values to the free variables controlled by the high-level. The performance of the exact algorithm is apparently affected by the number of variables controlled by high-level decision maker.

We also proposed a heuristic algorithm for speeding convergence. Empirical results show that the heuristic algorithm can provide a satisfactory near-optimal or optimal solution in a reasonable amount of computational time. It encourages us to believe that the heuristic procedure is a proper approach to solve the mixed (or pure) integer two-level linear programming problem.

Acknowledgement—This research was supported by the National Science Council of the R.O.C., grant NSC-77-0415-E007-03.

REFERENCES

1. G. W. Evans, An overview of techniques for solving multiobjective mathematical programs. *Mgmt Sci.* **30**, 1268–1282 (1984).
2. G. Kiziltan and E. Yucaoglu, An algorithm for multiobjective zero–one linear programming. *Mgmt Sci.* **29**, 1444–1453 (1983).
3. E. P. Winkofsky, N. R. Baker and D. J. Sweeney, A decision process model of R & D resource allocation in hierarchical organizations. *Mgmt Sci.* **27**, 268–283 (1981).
4. J. Fortuny-Amat and B. McCarl, A representation and economic interpretation of two-level programming problem. *J. Opl Res. Soc.* **32**, 783–792 (1981).
5. J. F. Bard and J. E. Falk, An explicit solution to the multi-level programming problem. *Computers Opns Res.* **9**, 77–100 (1982).
6. W. V. Candler and R. J. Townsley, A linear two-level programming problem. *Computers Opns Res.* **9**, 59–76 (1982).
7. J. F. Bard, An algorithm for solving the general bilevel programming problem. *Mathemat. Opns Res.* **8**, 260–272 (1983).
8. J. F. Bard, Optimality conditions for the bilevel programming problem. *Naval Res. Log. Q.* **31**, 13–26 (1984).
9. W. F. Bialas and M. H. Karwan, Two-level linear programming. *Mgmt Sci.* **30**, 1004–1020 (1984).
10. G. A. Kochenberger and V. H. Richard, A simple, all primal branch and bound approach to pure and mixed integer binary programs. *Opns Res. Lett.* **1**, 182–185 (1982).
11. R. K. Martin, D. J. Sweeney and M. E. Doherty, The reduced cost branch and bound algorithm for mixed integer programming. *Computers Opns Res.* **12**, 139–149 (1985).
12. G. E. Fox and G. D. Scudder, A simple strategy for solving a class of 0–1 integer programming models. *Computers Opns Res.* **13**, 707–712 (1986).