An integer linear programming problem with multi-criteria and multi-constraint levels: a branch-and-partition algorithm

Jun Li\textsuperscript{a} and Yong Shi\textsuperscript{b}\textsuperscript{*}

\textsuperscript{a}Department of Business Management, Daqing Petroleum Institute, Anda, Heilongjiang 151400, China
\textsuperscript{b}College of Information Science and Technology, University of Nebraska at Omaha, Omaha, NE 68182, USA

Corresponding author: e-mail address: yshi@unomaha.edu

Received 7 October 1999; received in revised form 18 May 2000; accepted 4 September 2000

Abstract

In this paper, we propose a branch-and-partition algorithm to solve the integer linear programming problem with multi-criteria and multi-constraint levels (MC\textsuperscript{2}-ILP). The procedure begins with the relaxation problem that is formed by ignoring the integer restrictions. In this branch-and-partition procedure, an MC\textsuperscript{2} linear programming problem is adopted by adding a restriction according to a basic decision variable that is not integer. Then the MC\textsuperscript{2}-simplex method is applied to locate the set of all potential solutions over possible changes of the objective coefficient parameter and the constraint parameter for a regular MC\textsuperscript{2} linear programming problem. We use parameter partition to divide the \((\lambda, \gamma)\) space for integer solutions of MC\textsuperscript{2} problem. The branch-and-partition procedure terminates when every potential basis for the relaxation problem is a potential basis for the MC\textsuperscript{2}-ILP problem. A numerical example is used to demonstrate the proposed algorithm in solving the MC\textsuperscript{2}-ILP problems. The comparison study and discussion on the applicability of the proposed method are also provided.

Keywords: MC\textsuperscript{2}-simplex method, MC\textsuperscript{2}-ILP problems, algorithm, integer programming

1. Introduction

Integer linear programming with multi-criteria and multi-constraint levels (MC\textsuperscript{2}-ILP) problems can arise quite naturally in many real-world applications. For example, an airline might use an integer linear programming to determine what and how many aircraft to buy, but the problem must consider multiple criteria, such as capability, economy, safety, and technology. In addition, resource availability levels of the problem can be differently represented by a group of decision-makers of the airline. For the last four decades, traditional integer programming (single objective and single constraint level) and multi-objective integer programming problems have been extensively studied by a number of scholars (see e.g. Dakin, 1965; Garfinkel and Nemhauser, 1972; Klein and Hannan, 1982; Lawler and Wood, 1966; Taha, 1975; Villareal and Karwan, 1981; Zionts, 1974). Various algorithms using both of the branch-and-bound and cut plane techniques have been developed to solve these problems. However,
because most of these mathematical models have only a single resource availability level, they may have limitations in dealing with real-world problems. It has been well recognized that many real-world problems should have multiple criteria (MC). To extend the framework of MC linear programming, Seiford and Yu (1979) and Yu (1985) formulated a model of multi-criteria and multi-constraint levels (MC^2) linear programming. This model is supported by both the linear system structure and real application. Shi and Lee (1997) initiated a formulation and the branch-and-bound algorithm for solving an MC^2 binary integer linear program (MC^2-BILP). The main aim of this paper is to first extend the result of Shi and Lee (1997) for proposing a mathematical formulation of MC^2-ILP with the framework of the MC^2 linear programming problem, and then to develop a branch-and-partition procedure to solve such integer programming problems.

In the next section, the notation and definitions of the MC^2 linear programming problem and the MC^2-simplex method are briefly reviewed. Then, we introduce the concepts of MC^2-integer potential solution and MC^2-integer potential basis for describing an MC^2-ILP problem. To solve such MC^2-ILP problems, we develop a branch-and-partition procedure, in which the relaxation of each sub-problem in the branches is an MC^2 linear programming problem. Then a numerical example is used to illustrate the branch-and-partition procedure for solving the MC^2-ILP problem. We also compare the proposed method for the MC^2-ILP problem with the algorithm for the MC^2-ILP problem (Shi and Lee, 1997). The last section is devoted to some discussion of the applicability of the method as well as the conclusions.

2. MC^2 linear programming

In this section, the solution concepts and the notation of MC^2 linear programming adopted from Seiford and Yu (1979) and Yu (1985) are sketched to facilitate the discussion on the MC^2-ILP model. An MC^2 linear programming problem can be formulated as

\[
\text{Maximize } \lambda^T CX \\
\text{s.t. } \begin{cases} AX = D\gamma \\ X \geq 0, \end{cases}
\]

(1)

where \( C \in \mathbb{R}^{q \times n} \), \( A \in \mathbb{R}^{m \times n} \), and \( D \in \mathbb{R}^{m \times t} \) are matrices of \( q \times n \), \( m \times n \), and \( m \times t \), dimensions, respectively; \( X \in \mathbb{R}^n \) are decision variables, \( \lambda \in \mathbb{R}^q \) is called the criteria parameter and \( \gamma \in \mathbb{R}^t \) is called the constraint level parameter. Both \((\lambda, \gamma)\) are assumed non-negative and unknown.

**REMARK 1.** It is immediately obvious that the MC^2 model is a symmetric extension of the MC when the constraint level parameter \( \gamma \) is known. In fact, for a particular value of \( \gamma \), the MC^2 problem reduces to the MC problem. Then, an efficient solution of \((\text{MC}, \gamma^0)\) is also an efficient solution of MC^2 at \( \gamma^0 \).

To find all the potential solutions for problem (1), we need to identify the corresponding set of potential bases for Problem (1). In the following discussion, we assume that both parameters \((\lambda, \gamma)\) are normalized; i.e., \( \lambda \in \mathbb{R}^q \) with \( \lambda_k > 0 \) and \( \sum \lambda_k = 1 \); \( \gamma \in \mathbb{R}^t \) with \( \gamma_k > 0 \) and \( \sum \gamma_k = 1 \). For a given basis \( J \) with its basic variables \( X(J) \), we can define the associated basic matrix \( B_J \) as the sub-matrix of \( A \) in (1) with the columns index of \( J \) (i.e., column \( j \) of \( A \) is in \( B_J \) if and only if \( j \in J \)), and the associated
objective function coefficients $C_J$ as the sub-matrix of $C$ with columns index of $J$. Let $X(J')$ be the non-basic variable corresponding to given $X(J)$. Then, we rearrange the index, if necessary, and decompose $A$ into $[B_J, N]$, where $N$ is the sub-matrix of $A$ associated with $X(J')$; and $C$ into $[C_B, C_N]$, where $C_N$ is the sub-matrix of $C$ associated with $X(J')$. The initial simplex tableau of Problem (1) can be shown in Table 1, and Table 1 can be rewritten as Table 2, where $I_m$ is an $m \times m$ identity matrix. By dropping $(\lambda, \gamma)$ from Table 2, we can obtain an MC$^2$-simplex tableau with a basis $B_J$ as Table 3.

**DEFINITION 1.** Given a basis $J$ for Problem (1), define its corresponding

1. Primal parameter set by $\Gamma(J) = \{\gamma > 0; B_J^{-1}D\gamma \geq 0\}$; and
2. Dual parameter set by $\Lambda(J) = \{\lambda > 0; \lambda[C_B B_J^{-1}N - C_N] \geq 0\}$.

**THEOREM 1.** Given a basis for $J$ for Problem (1),

1. The resulting feasible solution $X(J, \gamma) = B_J^{-1}D\gamma \geq 0$ if and only if $\gamma \in \Gamma(J)$;
2. The solution $X(J, \gamma)$ is optimal if and only if $\gamma \in \Gamma(J)$ and $\lambda \in \Lambda(J)$.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>The initial simplex tableau of Problem (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X(J)$</td>
<td>$X(J')$</td>
</tr>
<tr>
<td>$B_J$</td>
<td>$N$</td>
</tr>
<tr>
<td>$-\lambda \lambda C_B$</td>
<td>$-\lambda \lambda C_N$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Alternated simplex tableau of Problem (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X(J)$</td>
<td>$X(J')$</td>
</tr>
<tr>
<td>$I_m$</td>
<td>$B_J^{-1}N$</td>
</tr>
<tr>
<td>0</td>
<td>$\lambda[C_B B_J^{-1}N - C_N]$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Alternated simplex tableau of Problem (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X(J)$</td>
<td>$X(J')$</td>
</tr>
<tr>
<td>$I_m$</td>
<td>$B_J^{-1}N$</td>
</tr>
<tr>
<td>0</td>
<td>$C_B B_J^{-1}N - C_N$</td>
</tr>
</tbody>
</table>
DEFINITION 2. Given a basis \( J \) for Problem (1),

1. \( J \) is said to be a primal potential basis if \( \Gamma(J) \neq \emptyset \);
2. \( J \) is said to be a dual potential basis if \( \Lambda(J) \neq \emptyset \);
3. \( J \) is said to be a potential basis if \( \Gamma(J) \times \Lambda(J) \neq \emptyset \).

By using the MC²-simplex method discussed above, one can locate a set of potential solutions. A computer-software package for solving Problem (1) was developed by Hao and Shi (1996). The following example shows how to use the MC²-simplex method to identify the potential solutions.

Example 1. Consider the following MC² linear programming problem:

\[
\text{Max } z = (\lambda_1, \lambda_2) \begin{pmatrix} 2 & 5 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}
\]

subject to:

\[
\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 4 & 3 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix}
\]

where \( x_j \geq 0 \), \( j = 1, 2 \).

Let \( s_1 \) and \( s_2 \) be slack variables for restrictions 1 and 2, respectively. Using computer software of the MC²-simplex method, we obtain the set of all potential solutions, denoted by \( J_1 \) and \( J_2 \) as in Table 4.

In Table 4, \( J_1 \) has \((x_1, x_2)\) as the basic variables and \( J_1 \) is a potential solution whenever \( 0 < \gamma_1 \leq 1 \) and \( 0 < \lambda_1 \leq 0.857 \). Similarly, \( J_2 \) has \((s_2, s_2)\) as the basic variables and \( J_2 \) is a potential solution whenever \( 0 < \gamma_1 \leq 1 \) and \( 0.857 \leq \lambda_1 \leq 1 \). The fourth column of Table 4 displays the objective payoffs. Note that they are functions of \((\lambda, \gamma)\).

3. MC² integer linear programming

Many real-world decision problems can be classified as integer programming problems under a multi-criteria and multi-constraint levels environment. It thus becomes necessary and important to develop an MC²-ILP model. In order to do this, we recall that given a basis \( J \), its basic solution \( x(J, \gamma) = B_J^{-1}D\gamma \) is a function of parameter \( \gamma \). If the \( j^{th} \) component of \( x(J, \gamma) \) is a decision variable, then it is called a basic decision variable, denoted by \( x_j(J, \gamma) \); otherwise, it is a basic slack variable, denoted by \( s_j(J, \gamma) \). We also recall that both parameters \( \lambda, \gamma \) are normalized; i.e., \( \lambda \in \mathbb{R}^d \) with \( \lambda_k > 0 \)

<table>
<thead>
<tr>
<th>Potential bases</th>
<th>( \Gamma(J_i) )</th>
<th>( \Lambda(J_i) )</th>
<th>Payoff ( V(J_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X(J_1) = (x_1, x_2) )</td>
<td>( 0 \leq \gamma_1 \leq 1 )</td>
<td>( 0 \leq \lambda_1 \leq 0.857 )</td>
<td>( (\lambda_1, \lambda_2) \begin{pmatrix} 10 &amp; 6.33 \ 4 &amp; 10 \end{pmatrix} \begin{pmatrix} \gamma_1 \ \gamma_2 \end{pmatrix} )</td>
</tr>
<tr>
<td>( X(J_2) = (s_2, s_2) )</td>
<td>( 0 \leq \gamma_1 \leq 1 )</td>
<td>( 0.857 \leq \lambda_1 \leq 1 )</td>
<td>( (\lambda_1, \lambda_2) \begin{pmatrix} 10 &amp; 7.50 \ 4 &amp; 3 \end{pmatrix} \begin{pmatrix} \gamma_1 \ \gamma_2 \end{pmatrix} )</td>
</tr>
</tbody>
</table>
and $\sum \lambda_k = 1; \gamma \in R^t$ with $\gamma_k > 0$ and $\sum \gamma_k = 1$. We can present a mathematical model of an MC$^2$-ILP problem as follows:

Maximize $\lambda^t CX$

s.t. $AX = D\gamma$

$X$ are non-negative integers,

(2)

where $C \in R^{q \times n}$, $A \in R^{m \times n}$, and $D \in R^{m \times t}$ are matrices of $q \times n$, $m \times n$, and $m \times t$, dimensions, respectively; $X \in R^n$ are decision variables, $\lambda \in R^q$ is called the criteria parameter and $\gamma \in R^t$ is called the constraint level parameter. Both ($\lambda$, $\gamma$) are assumed non-negative and unknown.

We note that Model (2) is defined on general integer, which contains binary (or 0-1) integers in Shi and Lee (1997)'s MC$^2$-BILP formulation. We also note that in the branch-and-bound method for solving integer linear programming problems (Dakin, 1965; Hu, 1969; Lawler and Wood, 1966; Mitten, 1970), once the branching is made by dividing the original feasible set into smaller subsets, each subproblem needs to be solved to obtain the ‘bound’. Such a bound can be determined by the optimal objective value of the sub-problem. The branch-and-bound algorithm of Shi and Lee (1997) for solving MC$^2$-BILP is an application of LP branch-and-bound method, in which the bound is represented by the expected objective payoff over a certain ($\lambda$, $\gamma$) range. Instead of using a ‘bound’ to compute the subproblems, this paper proposes an algorithm that first branches decision variables $x_i(J, \gamma)$ according to the parameter $\gamma$ (since $x_i(J, \gamma)$ is a function of $\gamma$) on the known $\lambda$ range, and then searches for an integer potential solution on each ($\lambda$, $\gamma$) partition. Because we identify a set of integer potential solutions over the ‘partitions’ of the ($\lambda$, $\gamma$) space through a sequence without calculating a ‘bound’, this method is called a ‘branch-and-partition’ algorithm so as to distinguish from the known ‘branch-and-bound’ method. In addition, the set of integer potential solutions identified by this method is not unique since the searching sequence can vary.

**Definition 3.** A given potential solution $x(J)$ is said to be an ‘MC$^2$-integer potential solution’ if and only if all decision variables $x_i(J, \gamma)$ in $x(J)$ are integer. If $x(J)$ is an ‘MC$^2$-integer potential solution’, then the corresponding $J$ is called ‘MC$^2$-integer potential basis’.

**Definition 4.** A problem corresponding to Model (1) is said to be the relaxation problem of Model (2).

**Theorem 2.** The set of all feasible solutions for an MC$^2$-ILP problem is a sub-set of all the feasible solutions for its relaxation problem. Furthermore, if every potential basis for the relaxation problem is an MC$^2$-integer potential basis, then the set of all MC$^2$-integer potential bases for the relaxation problem is also the set of all MC$^2$-integer potential bases for the MC$^2$-ILP problem.

Based on the above results, we propose the following algorithm to solve the MC$^2$-ILP problem. Its flow chart can be expressed as Fig. 1.

*A branch-and-partition algorithm*

**Step 1.** Ignoring the integer restrictions, solve the relaxation problem as an MC$^2$ linear programming in order to obtain its potential basic solutions.
STEP 2. Select the first potential basic solution $T_1$ in nature order to solve the equations of all the basic decision variables for $x_j(T, \gamma) \in [I]$ (where $[I]$ is an integer set) in $\Gamma(T_1) \times \Lambda(T_1)$. We can obtain a solution set $\Omega(\gamma)$ of the equations for all $x_j(T, \gamma)$. Set $\Gamma(T_1) = \Gamma(T_1) \setminus \Omega(\gamma)$.

STEP 3. Select a basic decision variable $x_j$ as the branching variable; take the solutions of the equation

Fig. 1. Flow chart of branch-and-partition algorithm.
\( x_j(T, \gamma) \in [I] \) in \( \Gamma(T_1) \) as the interface to divide \( \Gamma(T_1) \). The purpose of doing this step is to make sure \( x_j \) has one integer value at most in any sub-set of \( \Gamma(T_1) \).

**STEP 4.** Select a sub-set \( \Gamma_1(T_1) \) to find the branching points \( b_1 \) and \( b_2 \). Here \( b_1 \) is the largest integer not exceeding any value of \( x_j(T_1, \gamma) \) in \( \Gamma_1(T_1) \), and \( b_2 \) is the next largest integer from its sibling \( b_1 \).

**STEP 5.** Construct the descendant MC\(^2\) linear programming problems and solve them with MC\(^2\)-simplex method. Each descendant has just one additional restriction \( x_j \leq b_1 \) or \( x_j \geq b_2 \).

**STEP 6.** Take the descendant MC\(^2\) linear programming problems as the original problems repeating the steps from Step 2 to Step 5 until both the descendant problems have MC\(^2\)-integer potential solutions for any \( \gamma \) in \( \Gamma_1(T_1) \).

**STEP 7.** Rearrange the MC\(^2\)-integer potential solutions by taking the solution that has maximum objective value in any sub-set of \( \Gamma_1(T_1) \).

**STEP 8.** Select another sub-set of \( \Gamma(T_1) \) repeating the steps from Step 4 to Step 7 until all the sub-sets of \( \Gamma(T_1) \) have been handled.

**STEP 9.** Select another potential solution of the relaxation problem repeating the steps from Step 2 to Step 8 until all the potential solutions have been selected.

**REMARK 2.** In general, the integer solutions of \( x_j(J, \gamma) \) in Step 3 result from \((t-1)\) dimensional equations about \( \gamma_1, \ldots, \gamma_t \) because of the normalization \( \sum \gamma_k = 1 \). In order to determine an interface \( \gamma^0 \) to partition \( \Gamma(T_1) \) for Step 4, we have to know \((t-2)\) components of \( \gamma^0 \) for solving the \((t-1)\)th component.

The following example is used to illustrate how the above algorithm can be employed to solve the MC\(^2\)-ILP problems.

**Example 2.** Based on the MC\(^2\) linear programming problem in Example 1, we construct an MC\(^2\)-ILP problem:

\[
\begin{align*}
\text{Max } & \quad z = (\lambda_1, \lambda_2) \begin{pmatrix} 2 & 5 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\
\text{s.t. } & \quad \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 4 & 3 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix}
\end{align*}
\]

where \( x_1, x_2 \) are non-negative integers.

**STEP 1.** Ignoring the integer restrictions, solve the relaxation problem as an MC\(^2\) linear programming. We can obtain the potential solutions that are shown in Tables 5 and 6.

We can express the parameter spaces corresponding to the potential solutions as Fig. 2.

**STEP 2.** Select the potential solution \( T_1 \) shown in Table 5:

For \( x_2: 1/3 + 5/3 \gamma_1 \in [I] \), then \( \gamma_1 = 2/5, 1 \).

For \( x_1: 7/3 - 7/3 \gamma_1 \in [I] \), then \( \gamma_1 = 1/7, 4/7, 1 \).

Because \( \Omega(\gamma) = \{ \gamma_1 = 1 \} \neq \emptyset \), we get an MC\(^2\)-integer potential solution at \( \gamma_1 = 1 \), that is \( X^* = (0, 2) \) at \( \gamma_1 = 1 \) and \( \lambda_1 \in [0, 6/7] \). Set \( \Gamma(T_1) = \Gamma(T_1) \setminus \Omega(\gamma) = \{ 0 \leq \gamma_1 < 1 \} \) and go to Step 3.

**STEP 3.** Select a basic decision variable \( x_2 \) as the branching variable and take \( \gamma_1 = 2/5 \) as the interface to divide \( \Gamma(T_1) \). It can be divided into \( \Gamma_1(T_1) = \{ 0 \leq \gamma_1 \leq 2/5 \} \) and \( \Gamma_2(T_1) = \{ 2/5 < \gamma_1 < 1 \} \).
STEP 4. Select a sub-set $\Gamma_1(T_1)$ to find the branching points $b_1$ and $b_2$, where $b_1 = 0$, $b_2 = 1$.

STEP 5. Construct the descendant $MC^2$ linear programming problems $MC^2$-LP1 and $MC^2$-LP2 (they are shown below). Solve $MC^2$-LP1 and $MC^2$-LP2 with $MC^2$-simplex method; we can obtain the results that are shown in Tables 7 and 8.

STEP 6. Select Table 7 repeating the steps from Step 2 to Step 5, we can obtain the following integer potential solutions:

Table 5
The first potential solution of Example 2

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$s_1$</th>
<th>$s_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>1</td>
<td>2/3</td>
<td>$-1/3$</td>
</tr>
<tr>
<td>$x_1$</td>
<td>1</td>
<td>0</td>
<td>$-1/3$</td>
<td>2/3</td>
</tr>
</tbody>
</table>

$\sigma_j$

<table>
<thead>
<tr>
<th></th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>8/3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Where $\Gamma(T_1) \times \Lambda(T_1) = \{0 \leq \gamma_1 \leq 1\} \times \{0 \leq \lambda_1 \leq 6/7\}$.

Table 6
The second potential solution of Example 2

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$s_1$</th>
<th>$s_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_2$</td>
<td>1/2</td>
<td>1</td>
<td>1/2</td>
<td>0</td>
</tr>
<tr>
<td>$s_2$</td>
<td>3/2</td>
<td>0</td>
<td>$-1/2$</td>
<td>1</td>
</tr>
</tbody>
</table>

$\sigma_j$

<table>
<thead>
<tr>
<th></th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td>0</td>
<td>5/2</td>
</tr>
<tr>
<td>$-3$</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Where $\Gamma(T_2) \times \Lambda(T_2) = \{0 \leq \gamma_1 \leq 1\} \times \{6/7 < \lambda_1 \leq 1\}$.

Fig. 2. Parameter spaces corresponding to the potential solutions.

STEP 4. Select a sub-set $\Gamma_1(T_1)$ to find the branching points $b_1$ and $b_2$, where $b_1 = 0$, $b_2 = 1$.

STEP 5. Construct the descendant $MC^2$ linear programming problems $MC^2$-LP1 and $MC^2$-LP2 (they are shown below). Solve $MC^2$-LP1 and $MC^2$-LP2 with $MC^2$-simplex method; we can obtain the results that are shown in Tables 7 and 8.

STEP 6. Select Table 7 repeating the steps from Step 2 to Step 5, we can obtain the following integer potential solutions:
Select Table 8 repeating the steps from Step 2 to Step 5, we can obtain the following integer potential solutions:

\[ X^* = (2, 0) \text{ at } \gamma_1 \in [0, 1/3] \text{ and } \lambda_1 \in [0, 6/7]; \]

\[ X^* = (2, 0) \text{ at } \gamma_1 = 1/3 \text{ and } \lambda_1 \in [0, 6/7]; \]

\[ X^* = (1, 0) \text{ at } \gamma_1 \in (1/3, 2/5] \text{ and } \lambda_1 \in [0, 6/7]. \]

Select Table 8 repeating the steps from Step 2 to Step 5, we can obtain the following integer potential solutions:

\[ X^* = (1, 1) \text{ at } \gamma_1 = 0 \text{ and } \lambda_1 \in [0, 6/7]; \]

\[ X^* = (1, 1) \text{ at } \gamma_1 \in [0, 2/5] \text{ and } \lambda_1 \in [0, 6/7]; \]

STEP 7. Rearrange the integer potential solutions and the result is shown in Fig. 3.

STEP 8. Select another sub-set \( \Gamma_2(T_1) \), repeating the steps from Step 4 to Step 7. The result is shown in Fig. 4.

STEP 9. Select another potential solution of the original relaxation problem \( T_2 \) shown in Table 6 to repeat the steps from Step 2 to Step 8. The result is shown in Fig. 5.
Fig. 3. Parameter partition (1) for some integer potential solutions.

Fig. 4. Parameter partition (2) for some integer potential solutions.

Fig. 5. Parameter partition (3) for some integer potential solutions.
MC²-LP1: Max $z = (\lambda_1, \lambda_2) \begin{pmatrix} 2 & 5 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 4 & 3 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

$x_2 \leq 0$ (Additional restriction)

where $x_1, x_2 \geq 0$.

MC²-LP2: Max $z = (\lambda_1, \lambda_2) \begin{pmatrix} 2 & 5 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 4 & 3 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

$x_2 \geq 1$ (Additional restriction)

where $x_1, x_2 \geq 0$.

Synthesizing Fig. 3 to Fig. 5, we can obtain an integer potential solution spaces about $(\lambda_1, y_1)$ that are shown in Fig. 6 (The arrows give the integer potential solutions when $(\lambda_1, y_1)$ are located on the borders.)

Comparing the above algorithm with the one in Shi and Lee (1997), we summarize that:

(i) Since Model (2) of this paper is defined on general integer, the MC²-BILP model (M2) of Shi and Lee (1997) is a special form of Model (2). However, solving the MC²-BILP model may be more difficult than solving Model (2).

(ii) The algorithm of Shi and Lee (1997) requires the calculation of the expected objective payoff (varying with the probability distribution) over a certain $(\lambda, y)$ range for each relaxation MC².
problem to obtain a 'bound'. In contrast, the proposed algorithm of this paper only searches for an integer potential solution on each \((\lambda, \gamma)\) partition without computing the expected payoff. This can reduce the computation time. However, if the number of \((\lambda, \gamma)\) partitions increases, the degree of the complexity of solving Model (2) may increase.

(iii) Because the different nature between the MC\(^2\)-BILP problem and the MC\(^2\)-ILP problem, we cannot use the algorithm of Shi and Lee (1997) to substitute the proposed algorithm, and vice versa.

A computer program to solve the MC\(^2\)-BILP problem of Shi and Lee (1997) has been developed by Hao and Shi (1997). Instead of implementing the algorithm of Shi and Lee (1997) completely, the computer code adopted a numeration method to solve binary integer linear problems with two criteria and two constraint levels. In addition, Wei and Shi (2000) developed the first version of a computer program to solve the MC\(^2\)-ILP problem proposed in this paper. A number of MC\(^2\)-ILP problems, including Example 2 in the paper, have been tested. For any MC\(^2\)-ILP problem with fewer than 30 decision variables and two criteria and two constraint levels, the CPU time is less than two seconds. The development of a PC program that can solve a large-size MC\(^2\)-ILP problem is an ongoing project.

4. Concluding remarks

We have presented a mathematical formulation of integer linear programming with multi-criteria and multi-constraint levels (MC\(^2\)) within the framework of the well-known MC\(^2\) linear programming. We have proposed a branch-and-partition method to solve such MC\(^2\)-ILP problems, including the definitions about MC\(^2\)-integer potential solutions and MC\(^2\)-integer potential bases in the algorithm. We have also used a numerical example to demonstrate the branch-and-partition method in solving the MC\(^2\)-ILP problem. In this paper, when the criteria and constraint levels are two dimensions, we can easily obtain the integer potential solutions with the different values of criteria and constraint level parameters. When the criteria and constraint levels are more than two dimensions, we can obtain the integer potential solutions on different combination planes about criteria and constraint level parameters with normalization.

The prototype model of this paper can be applied to many management science or operations research problems whenever (i) the quantity of optimal solutions is required to be ‘integer’, but not ‘fraction’; and (ii) the nature of the problems involve conflicting objectives and multiple (or a group of) decision-makers. For instance, the order of a number of oil tankers for an international shipping company can be classified as an MC\(^2\)-ILP problem. First, the quantity of a certain type of oil tankers (variable) cannot be a ‘fraction’, and must be an integer. Second, the criteria of the problem may be technology, price, speed, and capacity, which are in conflict with each other. If the high-technology equipment is ordered for the oil tanker, the price will increase. The trade-off of these criteria (represented by \(\lambda\)) cannot be addressed in traditional integer programming because it has only a single objective. Third, since buying oil tankers is a big decision, a group of decision-makers, such as the Chief Executive Officer, Financial Vice President, Operational Vice President, and Marketing Vice President can be involved in the decision-making. How to cluster the trade-offs of decision makers’ opinions (represented by \(\gamma\)) on the company’s available resource for the purchase is not addressed by any known multi-objective integer programming.
Collectively, the MC²-ILP model proposed in this paper is the model that cannot only find the integer quantity, but also show the trade-offs between the criteria and resource availability. Therefore, the MC²-ILP model based on integer programming and MC² theory will have a great potential applicability in real-world problem solving.

References


Hao, X.R., Shi, Y. 1996. Large-scale MC² program: A C++ program run on PC or Unix. College of Information Science and Technology, University of Nebraska at Omaha.

Hao, X.R., Shi, Y. 1997. MC² Binary Integer Program (version 1.0): A C++ Program run on PC or Unix. College of Information Science and Technology, University of Nebraska-Omaha.


Wei, F., Shi, Y. 2000. MC² Integer Program (version 1.0): A C++ Program run on PC. College of Information Science and Technology, University of Nebraska-Omaha.
