A METHOD FOR SOLVING 0-1 MULTIPLE OBJECTIVE LINEAR PROGRAMMING PROBLEM USING DEA

Golam Reza Jahanshahloo  Farhad Hosseinzadeh  Nagi Shoja  Ghasem Tohid
Teacher Training University  Islamic Azad University

(Received March 19, 2002; Revised July 19, 2002)

Abstract  In this paper, by using Data Envelopment Analysis (DEA) technique a method is proposed to find efficient solutions of 0-1 Multiple Objective Linear Programming (MOLP) problem. In this method from a feasible solution of 0-1 MOLP problem, a Decision Making Unit (DMU) without input vector is constructed in which output vector for DMU is the values of objective functions. The method consists of a two-stage algorithm. In the first stage, some efficient solutions are generated. In the second stage, the DMUs corresponding to the generated efficient solutions in the first stage together with the generated DMUs in the previous iterations are evaluated by using the additive model without input.

Keywords: DEA; 0-1 multiple objective linear programming; efficient solution

1. Introduction

Data Envelopment Analysis (DEA) is a mathematical programming technique, which is used to evaluate relative efficiency of Decision Making Units (DMUs) and has been proposed by Charnes et al. [4]. This technique has extended by Banker et al. (BCC model) [2]. The additive model, which is used in this paper, has proposed by Charnes et al. [5]. In the literature of DEA some articles can be found in which application of DEA in Multiple Objective Linear Programming (MOLP) and application of MOLP in DEA have been discussed (see [7,8]).

To solve 0-1 MOLP problem some methods by Bitran [3] and Deckro et al. [6] have been proposed. Bitran used relaxation technique to generate efficient solutions. He defined a relaxation problem and proved the efficient solutions of the relaxation problem that are feasible to original problem would also be efficient in the original problem. Deckro et al. reported computational results in terms of implicit enumeration compared to Bitran’s works. They claimed that their studies was compared favorably with Bitran’s results. The proposed method by Deckro et al. solves 0-1 MOLP problem through implicit enumeration. Liu et al. [8] proposed another method and used DEA technique to generate efficient solutions. They defined a DMU corresponding to each feasible solution of the problem and developed a two-stage algorithm to generate and evaluate DMUs. They used BCC model to evaluate the generated DMUs. Since in each iteration of their method three problems are solved in stage 1, it is not computationally efficient. In their method for the following reasons some efficient solutions are lost.

1) Existence of the convexity constraint in BCC model,

2) adding the constraints \[
\sum_{q=1}^{Q} \left( \sum_{r=1}^{s} u_{rj}^* c_{rq} - \sum_{i=1}^{m} v_{ij}^* a_{iq} \right) w_{qd} > u_{oj}^*, \quad \forall j \in G_k' \]

in step 1.1 of algorithm,
3) omitting the DMUs with negative inputs or outputs.
As well, their method may find some efficient solutions, which are not indeed the desired solution according to the definition in literature.
The difficulties mentioned above are illustrated by numerical examples at the end of this paper. Since there is a close relation between DEA and MOLP, we use this relation for solving 0-1 MOLP problem. The proposed method in this paper, in comparison with the proposed method by Liu et al., is more computationally efficient and removes some difficulties of this method.

In the next section, 0-1 Multiple Objective Programming (MOP) and DEA are introduced. In section 3, a method for finding efficient solutions of 0-1 MOLP problem by using DEA and reference hyperplane is proposed. Section 4 illustrates the procedure with some numerical examples. In the last section, a conclusion and some remarks are put forward.

2. 0-1 MOP and DEA
DEA and MOP are introduced briefly in the following subsections.

2.1. 0-1 multiple objective programming
A multiple objective programming problem is defined in the following form:

\[
\begin{align*}
\max & \quad (f_1(W), f_2(W), \ldots, f_k(W)) \\
\min & \quad (g_1(W), g_2(W), \ldots, g_t(W)) \\
\text{s.t.} & \quad W \in \Omega 
\end{align*}
\]

where \(f_1, f_2, \ldots, f_k\) and \(g_1, g_2, \ldots, g_t\) are objective functions and \(\Omega\) is a feasible region. If all objective functions are linear and \(\Omega\) is a convex polyhedron, then problem (1) is called a multiple objective linear programming problem. In order to solve problem (1), there are some different methods in the literature.

**Definition 1:** \(\overline{W} \in \Omega\) is said to be an efficient solution of problem (1) if and only if there does not exist a point \(W^0 \in \Omega\), such that:

\[(f_1(W^0), \ldots, f_k(W^0), -g_1(W^0), \ldots, -g_t(W^0)) \geq (f_1(\overline{W}), \ldots, f_k(\overline{W}), -g_1(\overline{W}), \ldots, -g_t(\overline{W}))\]

and inequality holds strictly for at least one index.
If in problem (1) all variables are restricted to be zero - one and all objective functions and constraints are linear then, problem (1) is called 0-1 MOLP problem and defined as follows:

\[
\begin{align*}
\max & \quad (f_1(W), f_2(W), \ldots, f_k(W)) \\
\min & \quad (g_1(W), g_2(W), \ldots, g_t(W)) \\
\text{s.t.} & \quad W \in \Omega \\
& \quad w_j \in \{0, 1\}, \quad j = 1, 2, \ldots, n
\end{align*}
\]

where \(W = (w_1, w_2, \ldots, w_n)^T\).

2.2. Data envelopment analysis
Consider \(n\) decision making units \(\text{DMU}_j\) \((j = 1, 2, \ldots, n)\) where each DMU consumes a \(m\)-vector input to produce a \(s\)-vector output. Suppose that \(X_j = (x_{1j}, x_{2j}, \ldots, x_{mj})^T\) and \(Y_j = (y_{1j}, y_{2j}, \ldots, y_{sj})^T\) are the vectors of inputs and outputs values respectively for DMU\(_j\), in which it has been assumed that \(X_j \geq 0 \& X_j \neq 0\) and \(Y_j \geq 0 \& Y_j \neq 0\). Consider the set \(S\) and its convex hull as follows:
Solving 0-1 MOLP Problem by Using DEA

\[ S = \left\{ \left( \frac{Y_j}{X_j} \right) \mid j = 1, 2, \ldots, n \right\} \]

\[ C(S) = \left\{ \left( \frac{Y}{X} \right) = \sum_{j=1}^{n} \lambda_j \left( \frac{Y_j}{X_j} \right), \sum_{j=1}^{n} \lambda_j = 1, \lambda_j \geq 0, j = 1, \ldots, n \right\}. \]

Suppose that \( \left( \frac{Y_p}{X_p} \right) \) corresponds to DMU\(_p\). If a vector belonging to \( C(S) \) can be found such that

\[ \left( \frac{Y}{X} \right) \geq \left( \frac{Y_p}{X_p} \right) \& \left( \frac{Y}{X} \right) \neq \left( \frac{Y_p}{X_p} \right) \]

then DMU\(_p\) is called inefficient, otherwise it is called efficient.

For evaluating relative efficiency of DMU\(_p\), additive model is used which is as follows:

\[
\begin{align*}
\text{Min } h_p &= -\sum_{i=1}^{m} s_i^- - \sum_{r=1}^{s} s_r^+ \\
\text{s.t.} \quad &\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{rp}, \quad r = 1, \ldots, s \\
&-\sum_{j=1}^{n} \lambda_j x_{ij} = -x_{ip}, \quad i = 1, \ldots, m \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \geq 0, \quad s_i^- \geq 0, \quad s_r^+ \geq 0, \quad j = 1, \ldots, n, \quad i = 1, \ldots, m, \quad r = 1, \ldots, s
\end{align*}
\]

where \( s_i^- \) and \( s_r^+ \) are the slack variables of the corresponding constraints of \( i^{th} \) input and \( r^{th} \) output, respectively. The dual of model (3) is:

\[
\begin{align*}
\text{Max } g_p &= \sum_{r=1}^{s} u_{rp} y_{rp} - \sum_{i=1}^{m} v_{ip} x_{ip} + u_{op} \\
\text{s.t.} \quad &\sum_{r=1}^{s} u_{rp} y_{rj} - \sum_{i=1}^{m} v_{ip} x_{ij} + u_{op} \leq 0, \quad j = 1, \ldots, n \\
&v_{ip} \leq -1, \quad i = 1, \ldots, m \\
&-u_{rp} \leq -1, \quad r = 1, \ldots, s
\end{align*}
\]

We know that DMU\(_p\) is efficient in the additive model if and only if \( h_p^* = g_p^* = 0 \) (see [5]).

**Lemma 1:** The additive model is translation invariant (see [1]).

As additive model is translation invariant, it can be used for evaluating relative efficiency of DMUs with zero or negative component of input/output vector. The additive model also can be used for evaluating relative efficiency of DMUs without input. The models (5) and (6) which are the multiplier and envelopment sides of additive model without input respectively, formulated as:

\[
\begin{align*}
\text{Max } Q_p &= \sum_{r=1}^{s} u_{rp} y_{rp} + u_{op} \\
\text{s.t.} \quad &\sum_{r=1}^{s} u_{rp} y_{rj} + u_{op} \leq 0, \quad j = 1, \ldots, n \\
&u_{rp} \geq 1, \quad r = 1, \ldots, s
\end{align*}
\]
Lemma 3: Theorem 1: Lemma 2:

Proof: If DMU exists a feasible solution for model (6), say \((s^+_r)\). Since \((s^+_r)\) is evaluated.

Consider the model (5) which is the dual of envelopment side of additive model without input is feasible and bounded. 

The proof is straightforward.

Consider the following problem:

\[ \text{Max} \quad \{C_1W, C_2W, \ldots, C_sW\} \]

\[ \text{s.t.} \quad A_iW \leq b_i, \quad i = 1, 2, \ldots, m \]

\[ w_j \in \{0, 1\}, \quad j = 1, 2, \ldots, n \] \quad (7)

where \(A_i = (a_{i1}, a_{i2}, \ldots, a_{in})\) and \(C_r = (c_{r1}, c_{r2}, \ldots, c_{rn})\) are the coefficients vector for the \(i^{th}\) constraint and the \(r^{th}\) objective function, respectively.

Corresponding to each feasible solution \(W_d\) in (7), the vector \(Y_d\) is defined as \(Y_d = (y_{1d}, y_{2d}, \ldots, y_{sd})^T\) where \(Y_d \in R^s \quad \& \quad y_{rd} = C_r W_d = \sum_{j=1}^{n} C_{rj} w_{jd}, \quad r = 1, \ldots, s\).

In order to use DEA technique for finding efficient solutions of the problem (7), corresponding to each vector \(Y_d\), we consider a DMU with output vector of \(Y_d\) and without input vector. By using additive model without input, the relative efficiency of constructed DMUs is evaluated.

Theorem 2: If DMU \(d\) is efficient in model (5) then \(W_d\) is an efficient solution for the problem (7).

Proof: If DMU \(d\) is efficient in (5) then it will be efficient in (6). By contradiction, suppose that \(W_d\) is not an efficient solution of (7). So, there exists \(W_{\beta}\) such that

\[ C_{r}(W_{\beta}) \geq C_{r}(W_d), \quad r = 1, \ldots, s \]

and inequality holds strictly for at least one index. That is, there will be at least one index, say \(l\), such that \(C_l(W_{\beta}) > C_l(W_d)\). Hence, \(y_{ld} > y_{ld}'\). From \(\bar{y} = e_l = (0, 0, \ldots, 0, 1, 0, \ldots, 0, 0)\) and \(C_l(W_{\beta}) > C_l(W_d)\), we will have \(\sum_{j=1}^{n} \bar{y} y_j > y_{ld}\). Hence, \(s^+_l > 0\). This means that there exists a feasible solution for model (6), say \((\bar{X}, S^+)\), such that \(Q_d = -s^+_l < 0\) and this is a contradiction.

3.1. The reference hyperplane

Let \( g_r = C_rW_r^* \), \( r = 1, 2, \ldots, s \) where \( W_r^* \) is the optimal solution of the following problem:

\[
\begin{align*}
\max & \quad C_rW_r \\
\text{s.t.} & \quad A_iW \leq b_i, \quad i = 1, 2, \ldots, m \\
& \quad w_j \in \{0, 1\}, \quad j = 1, 2, \ldots, n.
\end{align*}
\]

For \( r = 1, 2, \ldots, s \), (8) denotes \( s \) problem. To state the following theorem, we consider the \( l^{th} \) problem from problems (8) \( (r = l) \).

**Theorem 3:** If \( O_l = \{W_{l1}^*, W_{l2}^*, \ldots, W_{lf}^*\} \) is the set of optimal solutions of \( l^{th} \) problem from the problems (8), then at least one of these is an efficient solution of the problem (7).

Proof: First, suppose that \( O_l \) is singleton; namely, \( O_l = \{W_l^*\} \). We prove that \( W_l^* \) is an efficient solution of the problem (7). By contradiction, suppose that \( W_l^* \) is not an efficient solution. So, there will be a feasible solution such as \( W^o \) such that

\[
C_rW^o \geq C_rW_l^*, \quad r = 1, 2, \ldots, s \quad \& \quad \exists k \ (1 \leq k \leq s) ; \quad C_kW^o > C_kW_l^*.
\]

Hence, \( C_lW^o \geq C_lW_l^* \) and this contradicts \( C_lW^o < C_lW_l^* \).

Now, suppose \( O_l = \{W_{l1}^*, W_{l2}^*, \ldots, W_{lf}^*\} \) where \( f > 1 \) and \( \Omega \) is the feasible region of the problem (7). We prove that for each \( W^o \in \Omega - O_l \) there is not \( W^o_{ql} \in O_l \) such that:

\[
C_rW^o \geq C_rW_{ql}^*, \quad r = 1, 2, \ldots, s \quad \& \quad \exists k ; \quad C_kW^o > C_kW_{ql}^*.
\]

By contradiction, suppose that there are \( W^p \in \Omega - O_l \) and \( W^o_{ql} \in O_l \) such that:

\[
C_rW^p \geq C_rW_{ql}^*, \quad r = 1, 2, \ldots, s \quad \& \quad \exists k ; \quad C_kW^p < C_kW_{ql}^*\]

if \( k = l \) then \( C_lW^p > C_lW_{ql}^* \) which is a contradiction. If \( k \neq l \) then \( C_lW^p = C_lW_{ql}^* \) so, \( W^p \in O_l \) and this contradicts \( W^p \in \Omega - O_l \).

Let \( K = \{Y_{l1}, Y_{l2}, \ldots, Y_{lf}\} \) where \( Y_{ql} = (C_1W_{ql}^*, C_2W_{ql}^*, \ldots, C_sW_{ql}^*)^T(q = 1, 2, \ldots, f) \). It is evident that there is a member from \( K \), say \( Y_{ql} \), such that \( Y_{ql} \not\geq Y_{ql}^* \) \((q = 1, 2, \ldots, f, q \neq i) \).

Otherwise, for each member of \( K \), say \( Y_{jl} \), there should exist a member of \( K \), say \( Y_{jl} \), so that \( Y_{jl} \leq Y_{jl}^* \) & \( Y_{jl} \neq Y_{jl}^* \). Without loss of generality, suppose that

\[
\begin{align*}
Y_{l1} & \leq Y_{l2} \quad \& \quad Y_{l1} \neq Y_{l2}, \\
Y_{l2} & \leq Y_{l3} \quad \& \quad Y_{l2} \neq Y_{l3}, \\
\vdots \\
Y_{(f-1)l} & \leq Y_{fl} \quad \& \quad Y_{(f-1)l} \neq Y_{fl}, \\
Y_{fl} & \leq Y_{kl} \quad \& \quad Y_{fl} \neq Y_{kl}, \quad 1 \leq k < f
\end{align*}
\]

which is a contradiction. Therefore, \( W^o_{ql} \) is an efficient solution of the problem (7). \( \Box \)

**Definition 2:** The vector \( g \), which is defined as

\[
g = (g_1, g_2, \ldots, g_j, \ldots, g_s)^T = (C_1W_1^*, C_2W_2^*, \ldots, C_jW_j^*, \ldots, C_sW_s^*)^T,
\]

is called the ideal vector.

**Lemma 4:** The vector \( g = (g_1, g_2, \ldots, g_s)^T \) dominates all DMUs \((\neq g)\) which correspond to the feasible solutions of the problem (7).

Proof: Suppose \( \text{DMU}_o \) is a decision making unit corresponding to \( W^o \) where,

\[
Y^o = (C_1W^o, C_2W^o, \ldots, C_sW^o)^T \quad \& \quad Y^o \neq g
\]

and also suppose that \( W_r^*(r = 1, 2, \ldots, s) \) is an optimal solution of \( r^{th} \) problem from the problems (8). So, \( C_rW^o \leq C_rW_r^* \) \((r = 1, 2, \ldots, s) \). In other words, \( g \geq Y^o \). If \( \exists k(1 \leq k \leq s) ; \quad g_k > g_k^* \) then \( g \) dominates the vector \( Y^o \). Otherwise, \( C_rW^o = C_rW_r^* \) \((r = 1, 2, \ldots, s) \)
and \( W^o \) is an optimal solution of all the problems (8). Therefore, \( Y^o = g \) which contradicts \( Y^o \neq g \).

**Definition 3:** The sets \( G \) and \( G' \) are defined as follows:

\[
G = \{ Y_d \mid Y_d = (C_1W_d, C_2W_d, \cdots, C_sW_d)^T, \ A_iW_d \leq b_i, \ i = 1, 2, \cdots, m, \ w_{jd} \in \{0, 1\} \}
\]

and \( G' = G \cup \{ g \} \).

**Definition 4:** The hyperplane \( H = \{ Y \mid 1^T(Y - g) = 0 \} \) where \( 1^T = (1, 1, \ldots, 1) \) is called reference hyperplane. \( H \) is a supporting hyperplane on the convex hull of \( G' \) at \( g \) because 1) \( 1^T(Y - g) = 1^T(g - g) = 0 \) and 2) if \( Y = (y_1, y_2, \cdots, y_s) \in G' \), then according to Lemma (4), we will have \( y_r \leq g_r, r = 1, 2, \cdots, s \). So, \( y_1 + y_2 + \cdots + y_s \leq g_1 + g_2 + \cdots + g_s \); that is, \( 1^TY \leq 1^Tg \). Instead of using gradient \( 1 \), a vector, say \( a = ([g_1], [g_2], \cdots, [g_s])^T \), may be used to find the supporting hyperplane for \( G' \) at \( g \), but from the computational point of view, this is a difficult task.

### 3.2. Distance function

For each point \( Y^o \in G \), the distance function is defined as follows:

\[
D(Y^o) = \frac{|a^TY^o - a^Tg|}{\sqrt{a^Ta}}.
\]

Note that for each \( Y^o \in G \), \( a^TY^o \leq a^Tg \) so, \( D(Y^o) = \frac{1}{\sqrt{a^Ta}}(a^Tg - a^TY^o) \).

For illustration, consider the following problem:

\[
\begin{align*}
\text{Max} & \quad 4w_1 - 3w_2 + 5w_3 \\
\text{Max} & \quad 2w_1 + 7w_2 - w_3 \\
\text{s.t.} & \quad w_1 + 2w_2 + w_3 \leq 7 \\
& \quad 3w_1 + w_2 + 2w_3 \leq 6 \\
& \quad w_1, w_2, w_3 \in \{0, 1\}.
\end{align*}
\]

The feasible solutions of the above problem and its corresponding output vectors are shown in Table 1.

<table>
<thead>
<tr>
<th>( DMU )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W_i )</td>
<td>(0,0,0)</td>
<td>(1,0,0)</td>
<td>(0,1,0)</td>
<td>(0,0,1)</td>
<td>(1,1,0)</td>
<td>(1,0,1)</td>
<td>(0,1,1)</td>
<td>(1,1,1)</td>
</tr>
<tr>
<td>Output</td>
<td>(0,0)</td>
<td>(4,2)</td>
<td>(-3,7)</td>
<td>(5,-1)</td>
<td>(1,9)</td>
<td>(9,1)</td>
<td>(2,6)</td>
<td>(6,8)</td>
</tr>
</tbody>
</table>

It is evident that \( W_5 = (1, 1, 0) \) and \( W_6 = (1, 0, 1) \) are the optimal solutions of (8) for \( r = 1, 2 \), respectively. For this problem, the reference hyperplane, the ideal vector and the distance function are \( y_1 + y_2 = 18 \), \( g = (C_1W_1^*, C_2W_2^*)^T = (9, 9)^T \) and \( D(Y^o) = \frac{1}{\sqrt{2}}(18 - y_1^* - y_2^*) \), respectively which have been depicted in Figure 1.

### 3.3. Efficient solutions generation

Let \( G'' \) be the set of DMUs corresponding to all optimal solutions of the problems (8). To find efficient solutions of the problem (7), two steps should be taken. First, the members of \( G'' \) are evaluated by using the model (5) and efficient DMUs are specified. Suppose that \( G_o \)
is the set of efficient DMUs in $G''$. Based on the theorem (2), each $W$ corresponding to a
member of $G_o$ is an efficient solution of the problem (7) and these are not the only efficient
solutions of it. Let $(U_p^*, u_{op}^*)$ denote an optimal solution of the problem which corresponds
to DMU$_p$ (DMU$_p \in G''$). If DMU$_p$ is efficient in the model (5), $U_p^* Y + u_{op}^* = 0$ is the
supporting hyperplane on the production possibility set constructed by efficient DMUs (the
members of $G_o$) which is defined as follows:

$$T(G_o) = \{ Y \mid Y \leq \sum_{j=1}^{l} \lambda_j Y_j, \sum_{j=1}^{l} \lambda_j = 1, \lambda_j \geq 0, j = 1, 2, \ldots, l \}. $$

The supporting hyperplanes on $T(G_o)$ separate the set $G$ into two sets which are as follows:

$$G^H = \{ Y_d \mid U_p^* Y_d + u_{op}^* \leq 0, \quad p = 1, 2, \ldots, l \}, \quad G^E = G - G^H. $$

In the second step, in order to find other efficient solutions, we specify a point from $G^E$ that
has the minimal distance from the reference hyperplane. To do so, the following problem may be solved.

$$\text{Min } D(Y_d)$$

s.t. $U_p^* Y_d + u_{op}^* > -M t_p, \quad p = 1, 2, \ldots, l$

$t_1 + t_2 + \cdots + t_l \leq l - 1$

$Y_d \in G$

$t_p \in \{0, 1\}, \quad p = 1, 2, \ldots, l.$

(9)

where $D(Y_d)$ is the distance of the vector $Y_d$ from the reference hyperplane and $M$ is a
positive large number. Note that if $t_p = 1$, then constraint $U_p^* Y_d + u_{op}^* > -M$ is redundant.
Otherwise, this constraint is not redundant. The constraint $t_1 + t_2 + \cdots + t_l \leq l - 1$ implies
that at least one of the constraints $U_p^* Y_d + u_{op}^* > 0, p = 1, 2, \ldots, l$ is not redundant.

Based on the definition of the vector $Y_d = (y_{1d}, y_{2d}, \ldots, y_{sd})^T$, we will have:

$$y_{rd} = \sum_{j=1}^{n} C_{rj} w_{jd}, \quad r = 1, 2, \ldots, s,$$
Proof: Let $W$ be an optimal solution of the problem (10). By contradiction, suppose that $W^*$ is not efficient for the problem (7). Hence, the problem (7) has a feasible solution $W^o$ so that:

$$C_rW^o \geq C_rW^*_d, \quad r = 1, 2, \ldots, s \quad \text{and} \quad \exists k ; \quad C_kW^o > C_kW^*_d$$

By multiplying $u^*_{rp}$ in $C_rW^o \geq C_rW^*_d$ ($r = 1, 2, \ldots, s$) and summing them, we will have:

$$\sum_{r=1}^{n} \sum_{s} u^*_{rp}C_rW^o \geq \sum_{r=1}^{n} u^*_{rp}C_rW^*_d, \quad p = 1, 2, \ldots, l$$

$$\sum_{j=1}^{n} \sum_{s} u^*_{rp}C_jw^*_j \geq \sum_{j=1}^{n} \sum_{r=1}^{n} u^*_{rp}C_jw^*_jd, \quad p = 1, 2, \ldots, l$$

$$\sum_{j=1}^{n} u^*_{rp}C_jw^*_j > -u^*_{op} - t_pM, \quad p = 1, 2, \ldots, l.$$ 

Also, $W^o$ holds in the inequalities $\sum_{j=1}^{n} a_{ij}w^*_j \leq b_i, \quad i = 1, 2, \ldots, m$. Therefore, $W^o$ is a feasible solution of the problem (10). From (11), we have $\sum_{r=1}^{n} C_rW^o \geq \sum_{r=1}^{n} C_rW^*_d (Z_{W^o} > Z_{W^*_d})$ which is a contradiction.

After obtaining $W^*_d$, the corresponding DMU is constructed and the set of $G_1$ is defined as $G_1 = G_o \cup \{Y_d\}$. By evaluating the members of $G_1$ and continuing the above process, the efficient solutions of (7) are obtained. Since the number of the efficient solutions is finite and at least one efficient solution is found at each iteration, the algorithm is terminated in the finite number of the iteration. The aforementioned concepts have been illustrated in Figure 2. This Figure corresponds to a problem which has two objective functions.

In the Figure 2: 1) $L_1$ is the reference hyperplane, 2)$D(Y_d)$ is the distance $Y_d$ from $L_1$, 3) The points (1) and (3) are the members of $G_o$ and the point (2) is corresponding to an
efficient solution which is obtained in iteration 1 (that is \( G_1 = \{1, 2, 3\} \)), 4) \( L_2 \) and \( L_3 \) are the supporting hyperplanes on \( T(G_1) \) and 5) \( G^H \) and \( G^E \) are the set of the points • and ○, respectively.

![Figure 2: Distance \( D(Y_d) \) and sets \( G^H \) and \( G^E \)](image)

In order to develop the algorithm of finding efficient solutions, the sets \( G_k, G^H_k, G^E_k \) and \( T(G_k) \) are defined as follows:

- **\( G_k \):** \( G_k \) is a subset of \( G \) which denotes the set of the evaluated DMUs by the model (5) at the beginning of the \( k^{th} \) iteration.
- **\( G^H_k \):** \( G^H_k = \{Y_d \mid U^*_p Y_d + u^*_p \leq 0, \ p = 1, 2, \ldots, l, l + 1, \ldots, \eta\} \) where \( \eta \) is the number of the elements \( G_k \) and \( Y_d \) is corresponding to a feasible solution of (7), say \( W_d \).
- **\( G^E \):** \( G^E = G - G^H \).
- **\( T(G_k) \):** \( T(G_k) = \{Y \mid Y \leq \sum_{j=1}^{\eta} \lambda_j Y_j, \ \sum_{j=1}^{\eta} \lambda_j = 1, \lambda_j \geq 0, j = 1, 2, \ldots, \eta\} \).

### 3.4. An algorithm for generating efficient solutions

**Stage 0: Initialization**

Step 0-1: Find the output vector of DMUs corresponding to all optimal solutions of (8) and put, \( G'' = \{Y_1, Y_2, \cdots, Y_k\} \).

Step 0-2: By evaluating the members of \( G'' \) by means of the model (5), determine the members of \( G_o \) and the supporting hyperplanes on \( T(G_o) \).

**Stage 1: The generation of an Efficient Solution**

Step 1-1: Solve the problem (10). If it has an optimal solution, go to stage 2; otherwise, stop.

**Stage 2: The evaluation of DMUs and Generation of the Supporting Hyperplane**

Step 2-1: Determine the vector \( Y_d = (C_1 W^*_d, C_2 W^*_d, \ldots, C_s W^*_d) \) where, \( W^*_d \) is an efficient solution of the problem (10), and put \( G_{k+1} = G_k \cup \{Y_d\} \).

Step 2-2: Evaluate the members of \( G_{k+1} \) by using model (5) and identify the supporting hyperplanes on \( T(G_{k+1}) \) and go to stage 1.

### 4. Numerical Examples

**Example 1:** Consider the following 0-1 MOLP problem:
Max $2w_1 - 5w_2 + w_3$
Max $-7w_1 + 4w_2 + w_3$
\(s.t.\) $2w_1 + 3w_2 - w_3 \leq 7$
$-3w_1 + w_2 + 5w_3 \leq 6$
\(w_1, w_2, w_3 \in \{0,1\}\).

Table 2 denotes the feasible solutions of the problem, the outputs vector and the inputs vector of DMU corresponding to these feasible solutions.

<table>
<thead>
<tr>
<th>DMU (j)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>(W_i)</td>
<td>(0,0)</td>
<td>(1,0)</td>
<td>(0,1)</td>
<td>(0,0,1)</td>
<td>(1,1,0)</td>
<td>(1,0,1)</td>
<td>(0,1,1)</td>
<td>(1,1,1)</td>
</tr>
<tr>
<td>Input</td>
<td>(0,0)</td>
<td>(2,-3)</td>
<td>(3,1)</td>
<td>(-1,5)</td>
<td>(5,-2)</td>
<td>(1,2)</td>
<td>(2,6)</td>
<td>(4,3)</td>
</tr>
<tr>
<td>Output</td>
<td>(0,0)</td>
<td>(2,-7)</td>
<td>(-5,4)</td>
<td>(1,1)</td>
<td>(-3,-3)</td>
<td>(3,-6)</td>
<td>(-4,5)</td>
<td>(-2,-2)</td>
</tr>
</tbody>
</table>

As it can be seen, all DMUs (except DMU \(1\)) have at least one negative input or output. Thus, the proposed method by Liu et al. cannot determine the efficient solutions of this example. But the suggested method in this paper obtains efficient solutions, which are \(W_3 = (0,1,0)\), \(W_4 = (0,0,1)\), \(W_6 = (1,0,1)\) and \(W_7 = (0,1,1)\).

Example 2: Consider the following 0-1 MOLP problem in which \(W = (1,1)\) is its efficient solution.

Max $4w_1 + w_2$
Max $6w_1 + w_2$
\(s.t.\) $3w_1 + 5w_2 \leq 9$
$4w_1 + 7w_2 \leq 12$
\(w_1, w_2 \in \{0,1\}\).

This example has been solved by the proposed method in [8] and the presented method in this paper. To solve the above example by Liu’s method, we choose feasible solutions \(W_1 = (1,0)\), \(W_2 = (0,1)\), \(W_3 = (1,1)\) and \(W_4 = (0,0)\) in step 0-1 of the presented algorithm in [8]. The results of the evaluation of DMUs corresponding to these feasible solutions by BCC model are represented in Table 3.

Based on the proved theorem in [8], \(W_1 = (1,0)\) and \(W_3 = (1,1)\) are the efficient solutions of the above problem. But according to the definition of the efficient solution in literature (in [8]), \(W_1\) is not efficient; that is, the Liu’s method obtains a solution which is not efficient. The proposed method in this paper introduces only \(W_3 = (1,1)\) as the efficient solution.

Example 3: Consider a 0-1 MOLP problem which the corresponding DMUs of its feasible solutions have been depicted in Figure 3.
To solve the corresponding problem of Figure 3 by Liu’s method, if the points 2, 4 and 6 are chosen in step 0-1, the constraints $L_3 : U_1^* Y - V_1^* X > u_{01}^*$ and $L_4 : U_2^* Y - V_2^* X > u_{02}^*$ will be added to the problem in step 1-1. Figure 3 denotes that if these constraints are imposed to the problem, the points 3, 5 and 7 will be omitted. Hence, some efficient solutions may be lost.

Example 4: Consider the following 0-1 MOLP problem:

$$
\begin{align*}
& \text{Max} & & 4w_1 - 3w_2 + 5w_3 + w_4 \\
& \text{s.t.} & & 2w_1 + 7w_2 - w_3 - w_4 \leq 10 \\
& & & 4w_1 + 3w_2 + 5w_3 - 4w_4 \leq 12 \\
& & & -2w_1 + 4w_2 + 7w_3 + w_4 \leq 15 \\
& & & w_1, w_2, w_3, w_4 \in \{0, 1\}.
\end{align*}
$$

Stage 0, step 0-1: By solving $P_1$ and $P_2$, the set $G''$ is identified.

$$
\begin{align*}
P_1) \quad & \text{Max} & & 4w_1 - 3w_2 + 5w_3 + w_4 \\
& \text{s.t.} & & 2w_1 + 5w_2 - w_3 + 2w_4 \leq 10 \\
& & & 4w_1 + 3w_2 + 5w_3 - 4w_4 \leq 12 \\
& & & -2w_1 + 4w_2 + 7w_3 + w_4 \leq 15 \\
& & & w_1, w_2, w_3, w_4 \in \{0, 1\}.
\end{align*}
$$

$$
P_2) \quad & \text{Max} & & 2w_1 + 7w_2 - w_3 - w_4 \\
& \text{s.t.} & & 2w_1 + 5w_2 - w_3 + 2w_4 \leq 10 \\
& & & 4w_1 + 3w_2 + 5w_3 - 4w_4 \leq 12 \\
& & & -2w_1 + 4w_2 + 7w_3 + w_4 \leq 15 \\
& & & w_1, w_2, w_3, w_4 \in \{0, 1\}.
\end{align*}
$$

The optimal solutions of $P_1$ and $P_2$ are $W_1 = (1, 0, 1, 1)$ and $W_2 = (1, 1, 0, 0)$ respectively. Therefore, $G'' = \left\{ \begin{pmatrix} 10 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 9 \end{pmatrix} \right\}$.

Step 0-2: By evaluating the members of $G''$ by model (5), the members of $G_o$ and the supporting hyperplanes on the $T(G_o)$ are specified.

$$
\begin{align*}
& \text{Max} & & Q_1 = 10u_{11} + u_{01} \\
& \text{s.t.} & & 10u_{11} + u_{01} \leq 0 \\
& & & u_{11} + 9u_{21} + u_{01} \leq 0 \\
& & & u_{11}, u_{21} \geq 1
\end{align*}
$$

$$
\begin{align*}
& \text{Max} & & Q_2 = u_{12} + 9u_{22} + u_{02} \\
& \text{s.t.} & & 10u_{12} + u_{02} \leq 0 \\
& & & u_{12} + 9u_{22} + u_{02} \leq 0 \\
& & & u_{12}, u_{22} \geq 1
\end{align*}
$$

The results of evaluation have been presented in Table 4.

So, we have $G_o = \left\{ \begin{pmatrix} 10 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 9 \end{pmatrix} \right\}$ and $y_1 + y_2 - 10 = 0$, $y_1 + y_2 - 10 = 0$. The equation of the above hyperplane in terms of $w_j (j = 1, 2, 3, 4)$ is as follows:
Table 4: The results of evaluation

<table>
<thead>
<tr>
<th>variable</th>
<th>DMU₁</th>
<th>DMU₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u₁^* )</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( u₂^* )</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( u₀^* )</td>
<td>-10</td>
<td>-10</td>
</tr>
<tr>
<td>( Q^*_p )</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>result</td>
<td>efficient</td>
<td>efficient</td>
</tr>
</tbody>
</table>

\[ y₁ + y₂ - 10 = (4w₁ - 3w₂ + 5w₃ + w₄) + (2w₁ + 7w₂ - w₃ - w₄) - 10 = 0, \]

\[ 6w₁ + 4w₂ + 4w₃ - 10 = 0. \]

Iteration (1)

Stage 1, step 1-1: In this step, the following problem must be solved:

\[
\begin{align*}
\text{Max} & \quad 6w₁ + 4w₂ + 4w₃ \\
\text{s.t.} & \quad 6w₁ + 4w₂ + 4w₃ > 10 \\
& \quad 2w₁ + 5w₂ - w₃ + 2w₄ \leq 10 \\
& \quad 4w₁ + 3w₂ + 5w₃ - 4w₄ \leq 12 \\
& \quad -2w₁ + 4w₂ + 7w₃ + w₄ \leq 15 \\
& \quad w₁, w₂, w₃, w₄ \in \{0,1\}.
\end{align*}
\]

The alternative optimal solutions of the above problem are \( W₃ = (1,1,1,1) \) and \( W₄ = (1,1,1,0) \) which both are the efficient solutions of the problem.

Stage (2), step 2-1: \( Y₃ = (\frac{7}{7}) \) and \( Y₄ = (\frac{6}{8}) \) are the vectors corresponding to \( W₃ \) and \( W₄ \) respectively. Hence,

\[
G₁ = G₀ \cup \{Y₃, Y₄\} = \{ (\frac{10}{0}, \frac{1}{9}, \frac{7}{7}, \frac{6}{8}) \}.
\]

Step 2-2: The results of the evaluation of \( G₁ \)'s members have been presented in Table 5.

Table 5: The results for example 4

<table>
<thead>
<tr>
<th>variable</th>
<th>DMU₁</th>
<th>DMU₂</th>
<th>DMU₃</th>
<th>DMU₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u₁^* )</td>
<td>2.333</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( u₂^* )</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( u₀^* )</td>
<td>-23.33</td>
<td>-46</td>
<td>-14</td>
<td>-14</td>
</tr>
<tr>
<td>( Q^*_p )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The equations of the supporting hyperplanes on \( T(G₁) \) are as follows:

\[
\begin{align*}
2.333y₁ + y₂ - 23.33 &= 0 \\
y₁ + 5y₂ - 46 &= 0 \\
y₁ + y₂ - 14 &= 0
\end{align*}
\]

and the equations of them in terms of \( w_j (j = 1, 2, 3, 4) \) are as follows:

\[
\begin{align*}
11.332w₁ + 0.001w₂ + 10.665w₃ + 1.333w₄ - 23.33 &= 0 \\
14w₁ + 32w₂ - 4w₄ - 46 &= 0 \\
6w₁ + 4w₂ + 4w₃ - 14 &= 0
\end{align*}
\]

Iteration (2)

Stage 1, step 1-1: In this step, the following problem must be solved:
Max \quad 6w_1 + 4w_2 + 4w_3 \\
\text{s.t.} \quad 6w_1 + 4w_2 + 4w_3 \quad > \quad 14 - t_1M \\
14w_1 + 32w_2 - w_4 \quad > \quad 46 - t_2M \\
11.333w_1 + 0.001w_2 + 10.665w_3 + 1.333w_4 \quad > \quad 23.33 - t_3M \\
2w_1 + 5w_2 - w_3 + 2w_4 \quad \leq \quad 10 \\
4w_1 + 3w_2 + 5w_3 - 4w_4 \quad \leq \quad 12 \\
-2w_1 + 4w_2 + 7w_3 + w_4 \quad \leq \quad 15 \\
t_1 + t_2 + t_3 \quad \leq \quad 2 \\
w_1, w_2, w_3, w_4, t_1, t_2, t_3 \in \{0, 1\}.

The above problem is infeasible, so the algorithm is terminated and the efficient solutions are as: \( W_1 = (1, 0, 1, 1), W_2 = (1, 1, 0, 0), W_3 = (1, 1, 1, 1), W_4 = (1, 1, 1, 0). \)

Example 5: The following 0-1 MOLP problem is an adaptation of an example from [8]:

Max \quad 3w_1 + 6w_2 + 5w_3 - 2w_4 + 3w_5 \\
Max \quad 6w_1 + 7w_2 + 4w_3 + 3w_4 - 8w_5 \\
Max \quad 5w_1 - 3w_2 + 8w_3 - 4w_4 + 3w_5 \\
\text{s.t.} \quad -2w_1 + 3w_2 + 8w_3 - w_4 + 5w_5 \quad \leq \quad 13 \\
\quad \quad \quad 6w_1 + 2w_2 + 4w_3 + 4w_4 - 3w_5 \quad \leq \quad 15 \\
\quad \quad \quad 4w_1 - 2w_2 + 6w_3 - 2w_4 + w_5 \quad \leq \quad 11 \\
\quad \quad \quad w_1, w_2, w_3, w_4, w_5 \in \{0, 1\}.

Stage 0, step 0-1: By solving the problems corresponding to objective functions of the example 5 (problems \( P_1, P_2 \) and \( P_3 \)), we will have \( G'' = \{(14, 17, 10)^T, (11, 2, 16)^T\} \).

Step 0-2: By evaluating the members of \( G'' \) by model (5), the set \( G_o \) and the supporting hyperplane on \( T(G_o) \) are determined as \( G_o = \{(14, 17, 10)^T, (11, 2, 16)^T\} \) and \( y_1 + y_2 + 3y_3 - 61 = 0 \).

The equation \( y_1 + y_2 + 3y_3 - 61 = 0 \) in terms of \( w_j (j = 1, 2, 3, 4, 5) \) is as
\[ 24w_1 + 4w_2 + 33w_3 - 11w_4 + 4w_5 - 61 = 0. \]

Iteration (1)
Stage 1, step 1-1: In this step, we solve the following problem:

Max \quad 14w_1 + 10w_2 + 17w_3 - 3w_4 - 2w_5 \\
\text{s.t.} \quad -2w_1 + 3w_2 + 8w_3 - w_4 + 5w_5 \quad \leq \quad 13 \\
\quad \quad \quad 6w_1 + 2w_2 + 4w_3 + 4w_4 - 3w_5 \quad \leq \quad 15 \\
\quad \quad \quad 4w_1 - 2w_2 + 6w_3 - 2w_4 + w_5 \quad \leq \quad 11 \\
\quad \quad \quad 24w_1 + 4w_2 + 33w_3 - 11w_4 + 4w_5 \quad > \quad 61 \\
\quad \quad \quad w_1, w_2, w_3, w_4, w_5 \in \{0, 1\}. \\

The above problem is infeasible. So, the algorithm is terminated. \( W_1 = (1, 1, 1, 0, 0) \) and \( W_2 = (1, 0, 1, 0, 1) \) are the efficient solutions of the example 5.

As it can be seen, the suggested method obtains the efficient solutions of the example 5 in one iteration while the Liu’s method obtains the same solutions in three iterations.

5. Conclusion

This paper presents a method for solving 0-1 MOLP problem. In the proposed method for finding the efficient solutions of 0-1 MOLP problem, full enumeration is not used. In each iteration of the suggested algorithm, at least one efficient solution is found. Since the
number of feasible solutions is finite, the algorithm is convergent. The examples 1, 2, 3 and 5 illustrate the advantage of our method in comparison with Liu's method. The existence of the convexity constraint in the additive model without input, which is used in this paper for evaluating constructed DMUs, may eliminate some efficient solutions of the problem. This deficiency should be studied in the future. A modified version of this algorithm can be used for solving integer MOLP problem.

References


Prof. Gholamreza Jahanshahloo
Department of mathematics
Islamic Azad university
Research & Science branch
Tehran-IRAN
jahanshahlo@yahoo.com