The Third AFSS(1998), 506－511

EFFICIENCY TEST IN POSSIBILISTIC MULTIOBJECTIVE LINEAR PROGRAMMING

Masaaki IDA

Faculty of Engineering, Kyoto University,
Yoshida-Honmachi, Sakyo, Kyoto, 606-8501, Japan
Tel: +81-75-753-5257, Fax: +81-75-771-7286
E-mail: ida@prec.kyoto-u.ac.jp

Abstract

In this paper we consider multiobjective linear programming problems with coefficients of the objective functions specified by possibility distributions. Possibly and necessarily efficient solution sets are defined as fuzzy solution sets whose membership grades represent possibility and necessity degrees to which a feasible solution is efficient. Considering efficiency condition and its dual condition in ordinary multiobjective linear programming problem, we propose efficiency test methods based on an extreme ray generation method. Since the poposed methods can be put in the part of a bi-section method, we can develop calculation mehtods of the degree of possible and necessary efficiency for feasible solutions.

Keywords: Multiobjective programming, Possibilistic coefficient, Efficiency test, Extreme ray.

1. Introduction

In practical application it is usually difficult to determine the coefficients of the objective functions of multiobjective linear programming problems because they are specified subjectively by decision makers or analysts. we consider multiobjective linear programming problems with coefficients of the objective functions specified by possibility distributions [7]. Solutions for the problem are defined as two kinds of fuzzy sets, i.e., possibly efficient solution set and necessarily efficient solution set. Therefore, we need to calculate the degrees of possible or necessary efficiency for feasible solutions. Efficient solution generation methods formerly developed for usual multiobjective linear programming problems require to solve linear programming problems in efficiency test procedure [3],[9],[10]. Thus, in case of higher dimensional problems, they need excessive computational demand.

Considering efficiency condition and its dual condition in ordinary multiobjective linear programming problems, we propose effective efficiency test methods based on an extreme ray generation method. Since the poposed methods can be put in the part of a bi-section method, we can develop calculation mehtods of the degree of possible and necessary efficiency for feasible solutions. Finally, a numerical example is shown to demonstrate our methods.

2. Possibilistic multiobjective linear programming

2.1 Possibilistic objective function

A multiobjective linear programming problem is represented as

$$\text{Maximize } \{Cx \mid x \in F\},$$
$$F = \{x \mid Ax = b,\ x \geqq 0\}, \quad (1)$$

where C and A are $p \times n$ and $m \times n$ matrices. x and b are n and m column vectors [3],[9],[10].

Different from usual linear programming with single objective function, an optimal solution maximizing all objective functions simultaneously does not necessarily exist in a multiobjective linear programming problem.

For the problem efficient solution set EF is defined by

$$EF = \{x \in F \mid \text{ there is no } x' \in F \text{ such that } Cx' \geq Cx\}, \quad (2)$$

where $x \geq y$ means $x_i \geqq y_i$ for all i with at least one strict inequality.

In many practical cases, the coefficients of the objective functions can not be specified exactly, because of decision maker's subjectivity and the other outer factors. However, it is often the case that the coefficients can be specifed by possible ranges represented by fuzzy sets or intervals.

When the coefficients is specified by a fuzzy set regarded as a possibility distribution, the problem is called a "possibilistic multiobjective linear programming problem" [7]. A possibilistic multiobjective linear programming problem is represented as

$$\text{Maximize } \{\Phi x \mid x \in F\}, \quad (3)$$

where Φ is $p \times n$ matrix whose components θ_{ij}'s are mutually independent possibilistic variables restricted by L-R fuzzy numbers $\Gamma_{ij} = (c_{ij}^L, c_{ij}^R, d_{ij}^L, d_{ij}^R)$ with reference functions L_{ij} and R_{ij}. The membership function of Γ_{ij} is $\mu_{\Gamma_{ij}}(r)$.

Since θ_{ij}'s are mutually independent possibilistic variables, we can define a fuzzy set Γ which restrices the $p \times n$ matrix Φ as follows:

$$\mu_\Gamma(C) = \min_{i,j} \mu_{\Gamma_{ij}}(c_{ij}) \quad (4)$$

where C is a $p \times n$ matrix whose (i,j)-component is c_{ij}.

2.2 Possible and necessary efficiency

Let $P(x)$ be a set of $p \times n$ matrices with which x is efficient, i.e.,

$$P(x) = \{C \mid \text{there is no } x' \in F \text{ such that } Cx' \geq Cx\} \quad (5)$$

Inuiguchi and Sakawa [7] introduced the concepts of possibly and necessarily efficient solutions into the possibilistic muliobjective linear programming problem. In the case of possibilistic multiobjective linear programming, possibly and necessarily efficient solution sets ΠS and NS become fuzzy sets. Their membership functions, $\mu_{\Pi S}$ and μ_{NS}, can be defined as follows:

$$\mu_{\Pi S}(x) = \sup_{c}\{\mu_{\Gamma}(C) \mid C \in P(x)\}, \quad (6)$$

$$\mu_{NS}(x) = \inf_{c}\{1 - \mu_{\Gamma}(C) \mid C \notin P(x)\}. \quad (7)$$

These equations can be transformed into more operational forms [7]:

$$\mu_{\Pi S}(x) = \sup_{h}\{h \mid [\Gamma]_h \cap P(x) \neq \emptyset\}, \quad (8)$$

$$\mu_{NS}(x) = \sup_{h}\{h \mid [\Gamma]_{1-h} \subseteq P(x)\}. \quad (9)$$

If $\mu_{\Gamma_{ij}}$ is upper semi-continuous, the h-level set $[\Gamma]_h$ can be represented as follows:

$$[\Gamma]_h = \{C = (c_{ij}) \mid c_{ij}^L - L_{ij}^*(h)d_{ij}^L \leqq c_{ij} \leqq c_{ij}^R + R_{ij}^*(h)d_{ij}^R\}, \quad (10)$$

where L_{ij}^* and R_{ij}^* are pseudo-inverses of the reference functions L_{ij} and R_{ij} [7].

2.3 Interval objective function

If we fix the grade of membership h, all possible ranges of c_{ij} in equation (10) can be represented by colosed intervals denoted by $[l_{ij}, u_{ij}]$, i.e., the set Γ becomes the following crisp set:

$$\Gamma = \{C = (c_{ij}) \mid l_{ij} \leqq c_{ij} \leqq u_{ij}, \; i = 1, \ldots, p, \; j = 1, \ldots, n\}. \quad (11)$$

Hence the problem (3) becomes a multiobjective linear programming with interval objective functions, or called an "interval multiobjective linear programming problem". An interval multiobjective linear programming problem is a special case of possibilistic multiobjective linear programming.

Accordingly, if we can effectively check efficiency for feasible solutions in interval multiobjective linear programming problems, then the nonlinear programming problems, (8) and (9), can be solved by a bi-section method with respect to h and possible and necessary efficiency tests. Namely, we fix the degree h, and check the possible and necessary efficiency conditions, i.e., $[\Gamma]_h \cap P(x) \neq \emptyset$ and $[\Gamma]_{1-h} \subseteq P(x)$, until the grade h satisfies certain termination condition in the bi-section method. Consequently, we obtain the possible and necessary degrees for efficiency on the feasible solutions, i.e., $\mu_{\Pi S}(x)$ and $\mu_{NS}(x)$.

3. Interval multiobjective linear programming problem

3.1 Possible and necessary efficinecy for interval objective functions

In this section, we consider interval multiobjective linear programming problems and their efficiency condition.

Let C^p be the $2^n p \times n$ matrix with respect to Γ defined as follows:

$$C^p = \begin{pmatrix} l_{11} & u_{11} & l_{11} & u_{11} & \ldots & u_{11} & l_{21} & \ldots & u_{p1} \\ l_{12} & l_{12} & u_{12} & u_{12} & \ldots & u_{12} & l_{22} & \ldots & u_{p2} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ l_{1n} & l_{1n} & l_{1n} & l_{1n} & \ldots & u_{1n} & l_{2n} & \ldots & u_{pn} \end{pmatrix}^T \quad (12)$$

We obtain the following theorem from the theorem in [4]:

Theorem 1 *Determining the possibly efficient solutions for an interval multiobjective linear programming problem is equivalent to solve*

$$Max \; \{ C^p x \mid x \in F \}. \quad (13)$$

On the other hand, for necessarily efficient solution, Bitran obtained the following result [1].

Denote by M the subset of matrices of Γ having all elements of each column at the upper bound or at the lower bound. Hence, if $C \in M$, for $j = 1, \ldots, n$ either $C_{\cdot j} = L_{\cdot j}$ or $C_{\cdot j} = U_{\cdot j}$, where $L = (l_{ij})$, $U = (u_{ij})$. The maximun number of elements in M is 2^n. Following theorem was deduced:

Theorem 2 *Determining the necessarily efficient solutions for an interval multiobjective linear programming problem is equivalent to solve*

$$Max \; \{ Cx \mid x \in F, \; C \in M \}. \quad (14)$$

The former problem (13) is a large problem as to the matrix C^p, and the latter problem (14) represents a finite but large family of multiobjective linear programming problems. Therefore, for both problems, it is required to develop an effective algorithm for efficiency test.

3.2 Efficiency condition

Now let $B = B_{m \times m}$ be a nonsingular submatrix (also called a basis) of A, and N be the remaining $m \times (n-m)$ submatrix. Thus we can write (renumber the index if necessary) $A = (B, N)$.

We call x is a feasible basis of F when $B^{-1}b \geqq 0$. Let (C_B, C_N) be the matrices associated with B and N.

We have the following theorem for efficiency [3],[9]:

Theorem 3 *A feasible basis x is efficient, if and only if the system*

$$(C_B B^{-1} N - C_N)\,\mu \leq 0, \quad \mu = (\mu_1, \ldots, \mu_{n-m})^T \geqq 0 \quad (15)$$

has no solution.

Therefore,

$$P(x) = \{C = (C_B, C_N) \in \Gamma \mid \text{there is no } \mu \geqq 0 \text{ such that } (C_B B^{-1} N - C_N)\, \mu \leq 0\} \quad (16)$$

We obtain the following theorem from Theorem 1 for possible efficiency:

Theorem 4 *A feasible basis x is possibly efficient if and only if the system*

$$(C_B^p B^{-1} N - C_N^p)\mu \leq 0$$
$$\mu = (\mu_1, \ldots, \mu_{n-m})^T \geqq 0, \quad (17)$$

has no solution.

where (C_B^p, C_N^p) are the matrices associated with B and N.

On the other hand, the following theorem from Theorem 2 for necessary efficiency was obtained [1]:

Theorem 5 *A feassible basis x is necessarily efficient if and only if, for all $(C_B, C_N) \in M$, each system*

$$(C_B B^{-1} N - U_N)\mu \leq 0$$
$$\mu = (\mu_1, \ldots, \mu_{m-n})^T \geqq 0, \quad (18)$$

has no solution.

3.3 Efficient condition test

We denote relative cost coefficient matrix for C at a feasible basis x by

$$R = C_B B^{-1} N - C_N \quad (19)$$

As we saw in the previous section, the most essential problem in Theorem 4 and 5 is how to check efficinecy effectively for large or many relative cost coefficient matrices.

Theorem 3 can be transformed into a linear programming form [3],[9]:

Corollary 1 *A feasible basis x is efficient, if and only if the optimal value of the problem*

$$max \{ e\, s \mid R\, \mu + I\, s = 0,\ \mu \geqq 0, s \geqq 0\} \quad (20)$$

is zero.

where $e = (1, \ldots, 1)$ and I is a $p \times p$ identity matrix.

This linear programming problem has been used for efficiency test in former studies [3],[9], however, we can check efficiency in another forms for the problem. In this paper, we consider efficiency condition in detail, and propose various procedures for efficiency tests.

Next theorem is deduced directly from Theorem 3 and the Farkas lemma, that is a dual representation of efficiency condition:

Theorem 6 *A feasible basis x is efficient, if and only if the system*

$$R^T \nu \geqq 0,\ \nu = (\nu_1, \ldots, \nu_p)^T > 0 \quad (21)$$

has a solution.

We do not use the linear programmin problem in Corollary 1 in order to check efficiency, nevertheless, we examine existance of a solution, i.e., extreme ray that satisfies the inequality system in Theorem 3 and 6.

Hearafter, we say a matrix R is efficient on behalf of that x is efficient.

4. Extreme ray generation

Generally, algorithms for enumerating all extreme points of a polyhedron can be divided into two classes: pivoting methods and nonpivoting methods [8]. All of the nonpivoting methods can be viewd as variants of the Double Description Method.

In the case where the constraints $Ax \geqq 0$ subsume the constraints $x \geqq 0$, then the Double Description Method is identical to the procedure given by Chernikova [2]. That algorithm generates all the extreme rays of a convex polyhedral cone in the nonnegative orthant with vertex at the origin. We shall accordingly concern ourselves with finding all the extreme rays of the form:

$$K = \{w \mid Dw \geqq 0, w \geqq 0\}, \quad (22)$$

where D is $n_1 \times n_2$. Consider the matrix $(D^T, I)^T$ where I is $n_2 \times n_2$ identity matrix.

Chernikova's algorithm gives a series of transformation of this matrix that generates all the extreme rays [2],[8]. At any stage of the process we denote the old matrix by $Y = (U^T, L^T)^T$, and the new matrix being generated denoted by $\bar{Y}$ The matrices U and L will always have n_1 and n_2, respectively; however, they will in genereal not have n_2 columns. They will have more than n_2 columns in most cases, but if K lies in some subspace of R^{n_2} they may have fewer than n_2 columns. For $w \in R^{n_2}$, we use the sysmbol (w) to denote the ray $\{\lambda w, \lambda \geqq 0\}$. Let $Y_{i\cdot}$ and $Y_{\cdot j}$ denote the i-th row and j-th column of matrix Y, respectively. The Chernikova's algorithm is as follows:

(Extreme ray generation method) [2],[8]

1. (a) If any row of U has all components negative, then $w = 0$ is the only point in K.
   (b) If all the elements of U are nonnegative, then the columns of L are the edges of K, i.e., the ray $(L_{\cdot j})$ is an edge of K.
2. Choose the first row of U, say row r, with at least one negative elements.
3. (a) Let $R = \{ j \mid y_{rj} \geqq 0\}$ and $v = |R|$ (i.e., the number of elements of R). Then the first v columns of the new matrix, $\bar{Y}$, are $Y_{\cdot j}$ ($j \in R$).
   (b) If Y has only two columns and $y_{r1} y_{r2} < 0$, adjoin the column $|y_{r2}| Y_{\cdot 1} + |y_{r1}| Y_{\cdot 2}$ to the $\bar{Y}$ matrix. Go to step 5.
4. Let $S = \{(s,t) \mid y_{rs} y_{rt} < 0, s < t\}$, i.e., the set of all (unordered) pairs of columns of Y whose elements in row r have opposite signs. Let I_0 be the index set of all nonnegative rows of Y. For each $(s,t) \in S$, find all $i \in I_0$ such that $y_{is} = y_{it} = 0$ Call this set $I_1(s,t)$. We now use some of the elements of S to create additional columns for $\bar{Y}$:
   (a) If $I_1(s,t) = \emptyset$, then Y_s and Y_t do not contribute another column to the new matrix.
   (b) If $I_1 \neq \emptyset$, check to see if there is a u not equal to either s or t, such that $y_{iu} = 0$ for all $i \in I_1(s,t)$. If such a u exists, then $Y_{\cdot s}$ and

Y_t do not contribute another column to the new matirx. If no such u exists, then coose $\alpha_1, \alpha_2 > 0$ to satisfy $\alpha_1 y_{rs} + \alpha_2 y_{rt} = 0$. (one such choice is $\alpha_1 = |y_{rt}|, \alpha_2 = |y_{rs}|$.) Adjoin the column $\alpha_1 Y_s + \alpha_2 Y_t$ to the new matrix.

5. When all pairs in S have been examined, and the additional columns (if any) have been added, we say that row r has been processed. Now let Y denote the matrix $\bar{Y}$ produced in procesing row r and retrun to setp 1.

In the above algorithm, we have $U = D\ L$. Obviously maximun number of iteration of the algorithm, i.e., the number of row processes is n_1.

5. Efficiency test based on extreme ray generation

5.1 Simple efficiency tests

We apply the extreme ray generation method as described in previous section to Theorem 3 and 6. We propose following two efficiency tests.

(Efficiency test 1)

Let $D = -R$ in "Extreme ray generation method", we process each row of D, i.e., each row of $-R$. At any stage of the process, if an extreme ray $(L_{\cdot j})$ such that $U_{\cdot j} \geq 0$ is generated, then we find R is not efficient by Theorem 3. On the other hand, at the final stage, if there is no extreme ray that satisfies the condition in Theorem 3, then we find R is efficient.

(Efficiency test 2)

Let $D = R^T$ in "Extreme ray generation method", we process each row of D, i.e., each column of R. At the final stage of the process, if there is a common element of the cone generated by extreme rays obtained by the procedure and the strictly positive orthant, then we find R is efficient by Theorem 6. On the other hand, if there is no such element, then we find R is not efficient.

Above two tests are mutually independent, so two efficiency tests can be executed in parallel at the same time. Therefore we can test efficiency faster than the previous method in Corollary 1.

Previous efficiency test method was based on simplex method, then we must pivot until the optimal value is obtained. Namely, we search a extreme ray that satisfies the condition in Theorem 3 by depth first search.

On the other hand, our proposed methods in this section search the ray by width first search (i.e., row processes and column processes on R) until we find the extreme ray that satisfies the conditions in Theorem 3 and 6. Therefore, we can easily discriminate a feasible basis is efficient or not.

5.2 Consideration on R

In previous section, we proposed basic efficiency test methods based on extreme ray generation method. In this section, we consider the matrix R in detail in order to propose more effective efficiency test method.

At any stage of the extreme ray generation, we obtain the following theorem with respect to columns of R from Theorem 3:

Theorem 7 *If there is a column j such that $R_{\cdot j} \geq 0$, we eliminate j's column of R, and denote a new matrix R', then R is efficient is equivalent that R' is efficient.*

Similarlly we obtain the following from Theorem 6:

Theorem 8 *If there is a row i such that $R_{i\cdot} \leqq 0$, we eliminate i's row of R, and denote a new matrix R', then R is efficient is equivalent that R' is efficient.*

By using above theorems, the matrix R that we should check will be a reduction form.

We obtain the following theorem from Theorem 3:

Theorem 9 *If there is a column j such that $R_{\cdot j} \leq 0$, then R is not efficient.*

Simirally, from Theorem 6:

Theorem 10 *If there is a row i such that $R_{i\cdot} > 0$, then R is efficient.*

Above theorem directly deduces the following corollary:

Corollary 2 *If there is a row i such that $R_{i\cdot} \geq 0$ and $R_{ij} = 0$, and for the column j there is a row $i'(\neq i)$ such that $R_{i'j} > 0$, then R is efficient.*

By using above theorems and corollary, we can easily check effciency of R.

At any stage of the extreme ray generation, we calculate sums of column and row vectors of R, i.e., $R_{\Sigma\cdot} = \sum_i R_{i\cdot}$ and $R_{\cdot\Sigma} = \sum_j R_{\cdot j}$. We obtain the following theorems as for those sums from Theorem 3 and 6:

Theorem 11 *If $R_{\cdot\Sigma} \leq 0$, then R is not efficient.*

Theorem 12 *If $R_{\Sigma\cdot} > 0$, then R is efficient.*

5.3 Effective efficiency test

In section 5.1, we propose the efficiency test mehods based on extreme ray generation method. In this section, we propose more effective efficiency test mehod by useing the characteristics of R obtained in previous section.

(Efficiency test 3)

1. Calculate R for a feasible basis x.
2. (a) If there are columns such that $R_{\cdot j} \geqq 0$, then elininate such counms (Theorem 7).
   (b) If there are rows such that $R_{i\cdot} \leqq 0$, then elininate such rows (Theorem 8).
3. (a) If there is a column such that $R_{\cdot j} \leq 0$, then R is not efficient (Theorem 9).
   (b) If there is a row such that $R_{i\cdot} > 0$, then R is efficient (Theorem 10).

(c) If there is a row such taht $R_{i\cdot} \geq 0$ and a row that is $R_{i'j} > 0$ for $j(R_{ij} = 0)$, then R is efficient (Corollary 2).

4. Calculate $R_{\cdot\Sigma}$ and $R_{\Sigma\cdot}$.
5. (a) If $R_{\cdot\Sigma} \leq 0$, then R is not efficient (Theorem 11).

   (b) If $R_{\Sigma\cdot} > 0$, then R is efficient (Theorem 12).
6. Let $D = -R$ and process the rows by using "Extreme ray generation method". At the same time with this, let $D = R^T$ and process the rows in parallel by using "Extreme ray generation method".
7. Go to step 2.

In case of higher dimensional problem, a matrix R becomes very large. Concerning the efficiency test based on Corollary 1, we must rewrite the large scale simplex tableau in each step of the procedure.

On the other hand, our new method proposed is an improved method of "Efficiency test 1 and 2", therefore in each process we shrink the matrix R and check efficiency by simple methods.

The maximum number of processes of our method is n-m times as for column process and p times as for row process, however, in actual execution, we can use matrix shrinking processes, simple efficiency tests and parallel execution, the number of processes is far shorter than the number.

In this way our mehod is very effective efficiency test mehtod.

6. A numerical example

Let us consider the following three objective problem in reference [10]:

$$C = \begin{pmatrix} 1 & 2 & -1 & 3 & 2 & 0 & 1 & 0\,0\,0\,0 \\ 0 & 1 & 1 & 2 & 3 & 1 & 0 & 0\,0\,0\,0 \\ 1 & 0 & 1 & -1 & 0 & -1 & -1 & 0\,0\,0\,0 \end{pmatrix},$$

$$A = \begin{pmatrix} 1 & 2 & 1 & 1 & 2 & 1 & 2 & 1\,0\,0\,0 \\ -2 & -1 & 0 & 1 & 2 & 0 & 1 & 0\,1\,0\,0 \\ -1 & 0 & 1 & 0 & 2 & 0 & -2 & 0\,0\,1\,0 \\ 0 & 1 & 2 & -1 & 1 & -2 & -1 & 0\,0\,0\,1 \end{pmatrix},$$

$$b = \begin{pmatrix} 16 \\ 16 \\ 16 \\ 16 \end{pmatrix}.$$

We consider the following two feasible bases.

- $(0, 0, \frac{32}{3}, \frac{16}{3}, 0, 0, 0, 0, \frac{32}{3}, \frac{32}{3}, 0)$:

The matrix R for this basis is processed as shown in Table 1. Namely, at first, we eliminate second and third column of the initial matrix R, and process the first row. After eliminating some columns, we process the second row. Finally, we find the R is efficient by (3.(b)) of "Efficiency test 3" (the third row of the bottom matrix).

On Table 1, we saw efficiency test by row process in succession to first column elimination. Meanwhile, in case of column process of R, sequence of the column processes is shown in Table 2. Namely, we process the first column, and obtain the final matrix. we can find R is efficent by (3.(c)) of "Efficiency test 3" (the fourth row of the final matrix). In this case, we can check efficiency by only one column process.

On the other hand, in reference [10] for efficiency test on this feasible basis, we have to pivot two times in simplex algorithm, i.e., we must rewrite the large 4×10 simplex tableau twice.

(Initial matrix: $R = C_B B^{-1} N - C_N$)

		↓	↓					Σ
	$\frac{2}{3}$	0	0	$\frac{13}{3}$	$\frac{11}{3}$	$\frac{5}{3}$	$-\frac{4}{3}$	+
	$\frac{5}{3}$	2	0	$\frac{4}{3}$	$\frac{11}{3}$	$\frac{5}{3}$	$-\frac{1}{3}$	+
	$-\frac{4}{3}$	0	0	$-\frac{2}{3}$	$-\frac{1}{3}$	$-\frac{1}{3}$	$\frac{2}{3}$	−
Σ	+	+	0	+	+	+	−	

(Column elimination) ⇓

						Σ
→	$\frac{2}{3}$	$\frac{13}{3}$	$\frac{11}{3}$	$\frac{5}{3}$	$(-\frac{4}{3})$	+
	$\frac{5}{3}$	$\frac{4}{3}$	$\frac{11}{3}$	$\frac{5}{3}$	$-\frac{1}{3}$	+
	$-\frac{4}{3}$	$-\frac{2}{3}$	$-\frac{1}{3}$	$-\frac{1}{3}$	$\frac{2}{3}$	−
Σ	+	+	+	+	−	

(Process on the first row) ⇓

			↓	↓	↓	Σ
	$-\frac{4}{3}$	0	0	0	0	−
	$(-\frac{1}{3})$	$\frac{9}{3}$	$\frac{3}{3}$	$\frac{33}{3}$	$\frac{15}{3}$	+
	$\frac{2}{3}$	$-\frac{6}{3}$	$\frac{18}{3}$	$\frac{18}{3}$	$\frac{6}{3}$	+
Σ	−	+	+	+	+	

(Column elimination) ⇓

			Σ
	$-\frac{4}{3}$	0	−
→	$(-\frac{1}{3})$	$\frac{9}{3}$	+
	$\frac{2}{3}$	$-\frac{6}{3}$	−
Σ	−	+	

(Process on the second row) ⇓

			Σ
	$-\frac{4}{3}$	$-\frac{36}{3}$	−
	$-\frac{1}{3}$	0	−
	$\frac{2}{3}$	$\frac{12}{3}$	+
Σ	−	−	

Table 1: Efficiency test (row process)

- $(0, 0, 12, 0, 0, 4, 0, 0, 16, 4, 0)$:

The sequence of the processes on the matrix R is shown in Table 3. Namely, we eliminate the first row, and some columns. Finally we obtain a final matrix. By (5.(a)) of "Efficiency test 3" (the sum of columns), we find the matrix R is not efficient.

In reference [10], we must solve a linear programming problem for effciency test. On the other hand, by our method we need not process any row and column, only

	↓					Σ
	$(\frac{2}{3})$	$\frac{13}{3}$	$\frac{11}{3}$	$\frac{5}{3}$	$-\frac{4}{3}$	+
	$(\frac{5}{3})$	$\frac{4}{3}$	$\frac{11}{3}$	$\frac{5}{3}$	$-\frac{1}{3}$	+
	$-\frac{4}{3}$	$-\frac{2}{3}$	$-\frac{1}{3}$	$-\frac{1}{3}$	$\frac{2}{3}$	−
Σ	+	+	+	+	−	

(Process on the first column) ⇓

						Σ
	$\frac{2}{3}$	$\frac{13}{3}$	$\frac{11}{3}$	$\frac{5}{3}$	$-\frac{4}{3}$	+
	$\frac{5}{3}$	$\frac{4}{3}$	$\frac{11}{3}$	$\frac{5}{3}$	$-\frac{1}{3}$	+
	0	$\frac{24}{3}$	$\frac{21}{3}$	$\frac{9}{3}$	$-\frac{6}{3}$	+
	0	$\frac{6}{3}$	$\frac{39}{3}$	$\frac{15}{3}$	$\frac{6}{3}$	+
Σ	+	+	+	+	−	

Table 2: Efficiency test (column process)

by eliminating the useless columns and rows, we can easily check efficiency.

(Initial matrix: $R = C_B B^{-1} N - C_N$)

								Σ
→	$-\frac{6}{4}$	$-\frac{13}{4}$	$-\frac{13}{4}$	$-\frac{13}{4}$	$-\frac{7}{4}$	$-\frac{2}{4}$	$-\frac{1}{4}$	−
	1	1	-1	-1	2	1	0	+
	-1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	+
Σ	−	−	−	−	−	−	0	

(Row elimination) ⇓

		↓			↓	↓	↓	Σ
	1	1	-1	-1	2	1	0	+
	-1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	+
Σ	0	+	−	−	+	+	+	

(Column elimination) ⇓

			Σ
	1	-1	0
	-1	$\frac{1}{2}$	−
Σ	0	−	

Table 3: Efficiency test

7. Conclusion

In this paper, we considered possibilistic multiobjective lienar programming problems, and proposed effective possible and necessary efficiency test mehods based on extreme ray generation mathod.

We can apply our proposed method into possibly and necessarily efficient solution generation mehods discussed in [5], and will be able to develop effective mehod for efficient solution generation of possibilistic multiobjective lienar programming problems. Moreover, applications to the actual problems such as multiobjective optimal control problems under uncertainty [6] are expected.

References

1. G. R. Bitran : Linear Multiple Objective Problems with Interval Coefficients, Management Science, 26, pp.694-706 (1980)
2. N. V. Chernikova : Algorithm for Finding a General Formula for the Nonnegative Solutions of a System of Linear Inequalities, U.S.S.R. computational Mathematics and Mathematical Physics, 5, pp.228-233 (1965)
3. J. P. Evans and R. E. Steuer : A Revised Simplex method for Linear Multiple Objective Programs, Mathematical Programming, 5, pp.54-72 (1973)
4. M. Ida and O. Katai : Discrimination methods of Efficient Solutions for Multiobjective Linear Programming Problems with Interval Coefficients, Trans. of the Soc. of Instrument and Control Engineers Japan, 29, pp.1247-1249 (1993)
5. M. Ida : Multiple Objective Linear Programming with Interval Coefficients and Its All Efficient Solutions, Proc. of IEEE CDC 1996, pp.2122-2123 (1996)
6. M. Ida : Necessarily Efficient Solutions of Multiple Objective Linear Optimal Control Problems, Proc. of IEEE SMC 1997, pp.1884-1889 (1997)
7. M. Inuiguchi and M. Sakawa : Possible and necessary efficiency in possibilistic multiobjective linear programming problems and possible efficiency test, Fuzzy Sets and Systems, 78, pp.231-241 (1996)
8. T. H. Matheiss and D. S. Rubin : A Survey and Comparison of Methods for finding All Verteces of Convex Polyhedral Sets, Mathematics of Operations Research, 5, pp.167-185 (1980)
9. P. L. Yu and M. Zeleny : The set of All Nondominated Solutions in Linear Cases and a Multicriteria simplex Method, Journal of Mathematical Analysis and Applications, 49, pp.430-468 (1975)
10. M. Zeleny : Linear Multiobjective Programming, Springer Verlag (1974)