On necessarily efficient solutions in interval multiobjective linear programming

Milan Hladík

Charles University, Faculty of Mathematics and Physics,
Department of Applied Mathematics,
Malostranské nám. 25, 118 00, Prague, Czech Republic,
e-mail: milan.hladik@matfyz.cz.

ABSTRACT. We investigate multiobjective linear programming problems with objective coefficients varying inside given intervals. A feasible solution x^* is called necessarily efficient if it is efficient for all realizations of the interval objective function coefficients. Testing necessarily efficiency may be computationally expensive. Thus we propose one sufficient and also one necessary condition for necessarily efficiency that can significantly speed up decision algorithms. These conditions do not require the feasible solution x^* to be non-degenerate. We demonstrate usage of both conditions on illustrative examples and show how strong they are.

KEYWORDS. Multiobjective linear programming, efficient solution, interval matrix, interval analysis.

1 INTRODUCTION

In many real-life situations we come across problems with imprecise input values. Imprecisions are dealt with by various ways. One of them is interval based approach in which we model imprecise quantities by intervals, and suppose that the quantities may vary independently and simultaneously within their intervals.

In this paper, we investigate multiobjective linear programming (MOLP) problems in which objective function coefficients perturb within prescribed intervals. Interval MOLP was investigated by many authors using different approaches; an overview on interval MOLP was given by [Oliveira and Antunes, 2007]. Solving interval MOLP via preference ordering between intervals for was considered e.g. in [Chanas and Kuchta, 1996, Sengupta and Pal, 2009]. More attention was paid to possible and necessary efficiency. A solution is possibly efficient if it is efficient for at least one realization of interval objective function coefficients. Some fundamentals were stated by [Inuiguchi and Sakawa, 1996], and generation of all possibly efficient solution was investigated by [Wang and Wang, 2001a, Wang and Wang, 2001b] and [Ida, 1996]. More general approach involving uncertainties not only in the objective function coefficients but also in the constraints was considered e.g. in [Ida, 2007, Oliveira and Antunes, 2009, Urli and Nadeau, 1992].

Notion of necessarily efficiency is probably the most important concept of solution to interval MOLP since it ensures that a feasible point considered is efficient for all realizations of interval data. Some basic properties and theoretical foundations for necessarily efficiency were discussed in [Bitran, 1980, Inuiguchi and Sakawa, 1996, Oliveira and Antunes, 2007]. A branch & bound implicit enumeration algorithm for testing necessarily efficiency of a non-degenerate basic solution was proposed by [Bitran, 1980], and later improved by [Ida, 1999]. An exponential enumeration method for an arbitrary feasible point in case of just one criterion was presented by [Inuiguchi and Sakawa, 1994], and the whole necessarily efficient solution set generation by [Ida, 1996]. An application to portfolio selection problem can be found in [Ida, 2003, Ida, 2004].

There are problems closely related to interval MOLP. For instance, [Wang and Wang, 1997, Wang and Wang, 2001b] reduced fuzzy MOLP problems to parametric interval MOLP ones. [Benjamin, 2004] employed interval multi-objective programming for solving real-valued multi-objective decision problems by a branch & bound method; an application in robot controlling is studied in [Benjamin, 2002].

This paper is a contribution to necessarily efficiency testing. In Section 3 we give a novel characterization of necessarily efficiency. Sice the present methods for testing necessarily efficiency are computationally expensive (exponential in the worst case) there is a need for their accelerating. By using effective sufficient or necessary conditions we may significantly improve average time complexity; compare the conditions by [Inuiguchi and Sakawa, 1994] for one-objective case. In Section 3.1 we introduce an extension of the sufficient condition by [Bitran, 1980] and in Section 3.2 we propose a new necessary condition.

Throughout the paper, $A_{i,*}$ denotes the i-th row of a matrix A, and e a vector of ones (with convenient dimension). A diagonal matrix with entries $z_1, \dots, z_n$ is written as $\operatorname{diag}(z)$.

2 PRELIMINARIES

A multiobjective linear programming (MOLP) problem reads

$$\max_{x \in \mathcal{M}} Cx, \tag{1}$$

where the feasible set $\mathcal{M} := \{x \in \mathbb{R}^n \mid Ax \leq b\}$, $C \in \mathbb{R}^{s \times n}$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. A feasible solution x^* to (1) is called *efficient* if there is no $x \in \mathcal{M}$ such that $Cx \geq Cx^*$ with at least one strict inequality; we denote it briefly $Cx \gneqq Cx^*$.

Efficiency of points may be characterized by tangent and normal cones [Nožička et al., 1988, Rockafellar and Wets, 2004]. The tangent cone of $\mathcal{M}$ at the point x^* is defined

$$\mathcal{T}(x^*) := \{x \in \mathbb{R}^n \mid A_P x \leq 0\},$$

where $P := \{i \mid A_{i*}x^* = b_i\}$ and A_P denotes the submatrix of A consisting of the rows indexed by P. The normal (polar) cone [Nožička et al., 1988, Rockafellar and Wets, 2004] of $\mathcal{M}$ at the point x is defined as

$$\begin{aligned}\mathcal{N}(x^*) :&= \{x \in \mathbb{R}^n \mid x^T y \leq 0 \ \forall y \in \mathcal{T}(x^*)\} \\ &= \{A_P^T u \in \mathbb{R}^n \mid u \in \mathbb{R}^{|P|},\ u \geq 0\}.\end{aligned}$$

The extremal directions of the cone $\mathcal{N}(x^*)$ are constituted by the rows of the matrix A_P. Since $\mathcal{N}(x^*)$ is a convex polyhedral cone, it can by described by means of linear inequalities

$$\mathcal{N}(x^*) = \{x \in \mathbb{R}^n \mid Dx \leq 0\},$$

where $D \in \mathbb{R}^{r \times n}$ is an appropriate matrix. To determine such a description is an expensive task in general [Padberg, 1999]. One way is to compute all extremal directions h_i, $i \in I$, of $\mathcal{T}(x^*)$ to obtain the desired description

$$\mathcal{N}(x^*) = \{x \in \mathbb{R}^n \mid h_i^T x \leq 0 \ \forall i \in I\}.$$

For this task we can utilize the algorithm by Chernikova [Chernikova, 1965, Kuz, 1966] or the double description algorithm [Padberg, 1999]. Notice that extreme direction generation was also used e.g. by [Ida, 2005].

As long as x^* is a non-degenerate basic solution corresponding to a basis $B \subseteq \{1, \ldots, m\}$ then the normal cone reads

$$\mathcal{N}(x^*) = \{x \in \mathbb{R}^n \mid (A_B^T)^{-1}x \geq 0\},$$

that is, D is effectively computable and $D = -(A_B^T)^{-1}$.

Normal and tangent cones relate to efficiency in the following way. A point $x^* \in \mathcal{M}$ is efficient if and only if there is some positive combination of objectives lying inside $\mathcal{N}(x^*)$. In other words, if and only if $DC^T\lambda \leq 0$ for some $\lambda \in \mathbb{R}^s$, $\lambda > 0$. A point $x^* \in \mathcal{M}$ is not efficient if and only if there is $y \in \mathcal{T}(x^*)$ such that $Cy \gneqq 0$.

In this paper, we suppose that the objective functions coefficients are not known precisely. We are given only some lower and upper bounds as follows $\underline{c}_{ij} \leq c_{ij} \leq \overline{c}_{ij}$, $i = 1, \ldots, s$, $j = 1, \ldots, n$. Define an interval matrix

$$\boldsymbol{C} := [\underline{C}, \overline{C}] = \{C \in \mathbb{R}^{s \times n} \mid \underline{c}_{ij} \leq c_{ij} \leq \overline{c}_{ij},\ i = 1, \ldots, s,\ j = 1, \ldots, n\}.$$

The corresponding midpoint matrix and radius matrix are denoted respectively by $C^c := \frac{1}{2}.(\overline{C} + \underline{C})$ and $C^\Delta := \frac{1}{2}.(\overline{C} - \underline{C})$. By an interval MOLP problem we understood a family of problems

$$\max_{x \in \mathcal{M}} Cx, \quad \text{where} \quad C \in \boldsymbol{C}. \tag{2}$$

A feasible solution x^* is called *necessarily efficient* if it is efficient to (1) for every $C \in \boldsymbol{C}$.

3 NECESSARILY EFFICIENCY

Lemma 1. *The inequality $Cx \gneqq 0$ is true for some $C \in \boldsymbol{C}$ if and only if $C^c x + C^\Delta |x| \gneqq 0$.*

Proof. It is a slight modification of Gerlach theorem [Fiedler et al., 2006, Gerlach, 1981]. If $Cx \gneqq 0$, then

$$0 \lneqq C^c x + (C - C^c)x \leq C^c x + |C - C^c||x| \leq C^c x + C^\Delta |x|.$$

Conversely, suppose that $C^c x + C^\Delta |x| \gneqq 0$. Define $z = sgn(x)$. Then $|x| = \operatorname{diag}(z)x$ and

$$0 \lneqq C^c x + C^\Delta \operatorname{diag}(z)x = (C^c + C^\Delta \operatorname{diag}(z))x.$$

Since $C := C^c + C^\Delta \operatorname{diag}(z) \in \boldsymbol{C}$, the proof is completed. ☐

Remind that the tangent cone to $\mathcal{M}$ at the point x^* is described by the inequality system $A_P x \leq 0$. Below, we present a characterization of necessarily efficiency.

Theorem 1. *The vector x^* is necessarily efficient if and only if the system*

$$C^c x + C^\Delta |x| \gneqq 0,\ A_P x \leq 0,\ e^T |x| = 1 \tag{3}$$

has no solution.

Proof. The vector x^* is efficient to (1) with fixed $C \in \boldsymbol{C}$ iff

$$Cx \gneqq 0,\ A_P x \leq 0 \tag{4}$$

has no solution [Ehrgott, 2005]. Thus x^* is necessarily efficient iff there is no $C \in \boldsymbol{C}$ such that (4) is solvable. By Lemma 1, this is true iff

$$C^c x + C^\Delta |x| \gneqq 0,\ A_P x \leq 0$$

is not solvable. Using L^1-norm to normalize x we obtain the final form of (3). ☐

Observe that Theorem 1 gives rise to a simple but expensive algorithm for testing necessarily efficiency. Decomposing (3) according to signs of particular x_i-s we can provide the testing by solving up to 2^n linear programs. System (3) is solvable iff the systems

$$C^c x + C^\Delta |x| - y \geq 0,\ A_P x \leq 0,\ y \geq 0,\ e^T y \geq 1$$

is solvable, or equivalently, iff the linear system

$$C^c x + C^\Delta \mathrm{diag}(s) x - y \geq 0,\ A_P x \leq 0,\ \mathrm{diag}(s) x \geq 0,\ y \geq 0,\ e^T y \geq 1 \tag{5}$$

is solvable for all $s \in \{\pm 1\}^n$. Herein, $|x_i|$ was linearized by $s_i x_i$, where s_i is a sign of x_i. The number may be sometimes decreased when we employ sign restriction on variables x_i-s. Suppose that the system $A_P x \leq 0$ contains some non-negativity constraints $x_i \geq 0$, $i \in I$ for certain $I \subseteq \{1, \ldots, n\}$; non-positive variables are handled in a similar manner. Then we fix $s_i := 1$ for each $i \in I$, and hence it suffices to check solvability of (5) for all $s \in \{\pm 1\}^n$ such that $s_i = 1$, $i \in I$ and $s_i = \pm 1$, $i \notin I$. We reduced the number of possibilities to $2^{n-|I|}$, which can still be very high.

Notice that there are another algorithms of exponential time complexity, e.g. that one introduced by [Inuiguchi and Sakawa, 1994] for one-criterion case, or that by [Ida, 1996].

3.1 Necessary efficiency: a sufficient condition

Testing necessarily efficiency is a bit costly. That is why exploiting necessary or sufficient conditions may speed up significantly the decision process. We present a sufficient condition first, which improves that one by [Hladík, 2008] and extends the Bitran's condition [Bitran, 1980] for any feasible solution.

Theorem 2 (sufficient condition)**.** *Define the matrix $M \in \mathbb{R}^{r \times s}$ componentwise as*

$$m_{ij} := \sum_{k=1}^{n} d_{ik} c_{kj}(d_{ik}),$$

where

$$c_{kj}(d_{ik}) := \begin{cases} \overline{c}_{kj} & \text{if } d_{ik} \geq 0, \\ \underline{c}_{kj} & \text{if } d_{ik} < 0. \end{cases}$$

If the linear system

$$M\lambda \leq 0,\ \lambda \geq e \tag{6}$$

is solvable then x^* is necessarily efficient.

Proof. Let λ be a solution to (6) and $C \in \boldsymbol{C}$. It suffices to show that $DC^T\lambda \leq 0$ holds true. Since $d_{ik} c_{kj} \leq d_{ik} c_{kj}(d_{ik})$ we get

$$\sum_{k=1}^{n} d_{ik} c_{kj} \leq \sum_{k=1}^{n} d_{ik} c_{kj}(d_{ik}).$$

Therefore $DC^T \leq M$, whence $DC^T\lambda \leq M\lambda \leq 0$ follows. □

Note that (6) can be equivalently formulated as $\overline{D\boldsymbol{C}^T}\lambda \leq 0,\ \lambda \geq e$ by using interval arithmetic [Alefeld and Herzberger, 1983].

Generally, the proposed sufficient condition is not the necessary one. The reason is that M is entrywise the best upper bound for DC^T, $C \in \boldsymbol{C}$, but the particular maximizers are attained for different matrices $C \in \boldsymbol{C}$.

To use the sufficient condition presented in Theorem 2 we have to check solvability of a linear system of inequalities. This is an easy task for a linear programming solver. Moreover, we accelerate the decision process when we check a promising candidate for a solution to (6). For instance, such a candidate may be a vector of weights proving efficiency of x^* for some $C \in \boldsymbol{C}$ (typically the midpoint matrix).

As long as x^* is non-degenerate, the proposed sufficient condition is very cheap; it requires just to solve one linear program to check solvability of a linear system (6). If it is not the case, calculation of D might be computationally expensive. We can overcome this drawback by computing only a subset of $\mathcal{N}(x^*)$ that correspond to any feasible basis. Particularly, take any feasible basis $B \subseteq \{1, \dots, m\}$ corresponding to x^* and put $D := -(A_B^T)^{-1}$. Then the inequality system $Dx \leq 0$ determines a part of the normal cone $\mathcal{N}(x^*)$. The method remains still valid, but the sufficient condition will be weaker. Nevertheless, it can happen that x^* is necessarily efficient even though it is confirmed for no subpart corresponding to any feasible basis.

Notice that for a non-degenerate basic solution x^* our condition coincides with the stopping criterion in the branch & bound method used by [Bitran, 1980]. Thus our approach generalizes Bitran's results to possibly degenerate point. In this manner we can extend the Bitran's implicit enumeration method to an arbitrary feasible point.

Example 1. Let us consider an example by [Inuiguchi and Sakawa, 1996] with two objectives:

$$\boldsymbol{C} = \begin{pmatrix} [2,3] & [1.5, 2.5] \\ [3,4] & [0.5, 0.8] \end{pmatrix}, \quad A = \begin{pmatrix} 3 & 4 \\ 3 & 1 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 42 \\ 24 \\ 9 \\ 0 \\ 0 \end{pmatrix}.$$

We want to check whether a feasible solution $x^* = (6,6)^T$ is necessarily efficient. Since x^* is a non-degenerate basic solution corresponding to the basis $B = \{1, 2\}$ we compute the normal cone at x^* as follows

$$\mathcal{N}(x^*) = \{x \in \mathbb{R}^n \mid -(A_B^T)^{-1}x \leq 0\} = \{x \in \mathbb{R}^2 \mid x_1 - 3x_2 \leq 0,\ -4x_1 + 3x_2 \leq 0\}.$$

Now, the linear system (6) reads

$$-1.5\lambda_1 + 2.5\lambda_2 \leq 0,\ -0.5\lambda_1 - 9.6\lambda_2 \leq 0,\ \lambda_1, \lambda_2 \geq 1.$$

Obviously, this system has a solution, e.g. take $\lambda_1 = 2,\ \lambda_2 = 1$. Thus $(6,6)^T$ is necessarily efficient.

3.2 Necessary efficiency: a necessary condition

In the following we are concerned with a necessary condition for necessarily efficiency of x^*. Necessary conditions were not thoroughly studied even though its importance was observed already by [Bitran, 1980].

Let us describe the idea behind our approach. Suppose that x^* is efficient to (1) for some $C^1 \in \boldsymbol{C}$, but is not efficient to (1) for certain $C^2 \in \boldsymbol{C}$. Then we can expect that there is some $k \in P$ such that the vector $A_{k,*}$ (i.e., the extremal direction of $\mathcal{N}(x^*)$) lies in the convex cone generated by the rows of C^1 (or row of another matrix in $\boldsymbol{C}$), but does not lie in the convex cone generated by the rows of C^2. To find out the matrix certifying necessarily non-efficiency of x^* we can proceed in the following manner. Define the direction $d := A_{k,*} - \sum_{i \in P, i \neq k} A_{i,*}$. Now, determine the matrix $C^0 \in \boldsymbol{C}$ that its rows are the farest one in the direction of d, that is, $C^0_{i,*} = \operatorname{argmax}_{C_{i,*} \in \boldsymbol{C}_{i,*}} C_{i,*}d$, $i = 1, \ldots, s$. It is easy to see that $c^0_{ij} = \overline{c}_{ij}$ if $d_j \geq 0$ and $c^0_{ij} = \underline{c}_{ij}$ otherwise. Because of the construction of C^0 we may hope that the convex cone generated by the rows of C^0 contains no point from the convex hull of $A_{i,*}$, $i \in P$. It would mean that x^* is not efficient to (1) for the objective matrix C^0.

The formal formulation is given in Theorem 3.

Theorem 3 (necessary condition)**.** *Let $k \in P$. Put $d := A_{k,*} - \sum_{i \in P, i \neq k} A_{i,*}$ and define $C^0 \in \boldsymbol{C}$ column-wise as follows*

$$C^0_{*,j} = \begin{cases} \overline{C}_{*,j} & \text{if } d_j \geq 0, \\ \underline{C}_{*,j} & \text{otherwise}, \end{cases}$$

$j = 1, \ldots, n$. If the linear system

$$C^0 x - y \geq 0, \;\; A_P x \leq 0, \;\; y \geq 0, \;\; e^T y = 1. \tag{7}$$

is solvable then x^* is not necessarily efficient.

Proof. From definition, C^0 comes always from $\boldsymbol{C}$. The point x^* is efficient to (1) for the objective matrix C^0 if and only if the linear system (7) is not solvable. If it turns out that x^* is not efficient for a particular $C^0 \in \boldsymbol{C}$ then it cannot be necessarily efficient. □

We employ the necessary condition in this way: Solve the linear programs (7) for particular $k \in P$ until we find that x^* is not necessarily efficient, or process every $k \in P$ without conclusion. It requires to solve the total number of at most m linear programs, usually about n or even less.

Our method runs in polynomial time provided that we solve linear programs by an appropriate interior point method.

Remark 1. Another approach to a necessary condition is as follows. We may try to find out or generate a point that—in the case x^* is not necessarily efficient—dominates x^* for some realization of interval data. This candidate may be e.g. a vector in direction to a neighboring vertex to x^*. That is, if x^0 is a neighbor to x^* then check $C^c(x^0 - x^*) + C^\Delta |x^0 - x^*| \gneqq 0$. This is particularly promising direction if x^0 was previously recognized as necessarily efficient solution.

Example 2. Consider the interval MOLP problem

$$\max \; \boldsymbol{C}x \;\; \text{subject to} \;\; Ax \leq b, \; x \geq 0$$

with data from [Ida, 1999, Oliveira and Antunes, 2007]

$$\boldsymbol{C} = \begin{pmatrix} [1,2] & [2,3] & [-2,-1] & [3,4] & [2,3] & [0,1] & [1,2] \\ [-1,0] & [1,2] & [1,2] & [2,3] & [3,4] & [1,2] & [0,1] \\ [3,4] & [0,1] & [1,2] & [1,2] & [0,1] & [-2,-1] & [-2,-1] \end{pmatrix},$$

$$A = \begin{pmatrix} 1 & 2 & 1 & 1 & 2 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 & 0 & 1 \\ -1 & 0 & 1 & 0 & 2 & 0 & -2 \\ 0 & 1 & 2 & -1 & 1 & -2 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 16 \\ 16 \\ 16 \\ 16 \end{pmatrix}.$$

We want to test necessarily efficiency of the point $x^* = \left(0, 0, \frac{32}{3}, \frac{16}{3}, 0, 0, 0\right)^T$ along Theorem 3. Determine $P = \{1, 4, 5, 6, 9, 10, 11\}$, where indices greater then four stand for non-negativity constraints. Put $k := 1$. Then

$$d := A_{1,*} - \sum_{i \in P,\, i \neq 1} A_{i,*} = (2, 2, -1, 2, 2, 4, 4).$$

The corresponding objective function matrix reads

$$C^0 = \begin{pmatrix} 2 & 3 & -2 & 4 & 3 & 1 & 2 \\ 0 & 2 & 1 & 3 & 4 & 2 & 1 \\ 4 & 1 & 1 & 2 & 1 & -1 & -1 \end{pmatrix}$$

and the linear system (7) has a solution $x = (0.0417, 0, -0.2085, 0.1667, 0, 0, 0)^T$ and $y = (0.4167, 0.2917, 0.2916)^T$. Therefore x^* is not necessarily efficient.

Notice that for $k \in \{5, 6, 9, 10, 11\}$ we obtain the same result, only $k = 4$ does not certify necessarily non-efficiency of x^*.

Necessarily efficiency of x^* can be also disproved along Remark 1. The vertex x^* is adjacent to the vertex $x^0 = (0, 0, 0, 16, 0, 0, 0)^T$, and for some $C \in \boldsymbol{C}$ the vertex x^0 dominates to x^* since

$$C^c(x^0 - x^*) + C^\Delta |x^0 - x^*| = (64, \tfrac{64}{3}, \tfrac{32}{3})^T \gneqq 0.$$

Notice that x^0 is necessarily efficient, which may be shown by Theorem 2.

4 NUMERICAL EXPERIMENTS

We carried out some numerical experiments to tell us about the strength of sufficient and necessary conditions presented in previous sections. The computations were done in MATLAB 7.7.0.471 (R2008b). In accordance with [Bitran, 1980] we considered the interval MOLP in the form

$$\max_{x \in \mathcal{M}} Cx, \qquad C \in \boldsymbol{C},$$

where the feasible set was defined as $\mathcal{M} := \{x \in \mathbb{R}^n \mid Ax = b,\ x \geq 0\}$. The matrices $C^c, C^\Delta \in \mathbb{R}^{s \times 2n}$, $A \in \mathbb{R}^{n \times 2n}$ and $b \in \mathbb{R}^n$ were pseudorandomly generated in the following way. Entries of A were uniformly distributed in $[-10, 10]$ with exception of the first row; that one was composed of random numbers in $[0, 20]$ in order that the feasible set is bounded. The right-hand side was set as $b := Ae$. Entries of C^c were randomly chosen in $[-10, 10]$, too, and entries of C^Δ in $[0, R]$, where $R > 0$ was a parameter. The number of $(s+1)$ vertices for testing of necessarily efficiency were determined as optimal solutions to linear programming scalarizations $\max_{x \in \mathcal{M}} e^T C^c x$ and $\max_{x \in \mathcal{M}} e_k^T C^c x$, $k = 1, \ldots, s$, where e_k denotes the k-th Cartesian unit vector.

In each setting of s, n and R we carried out a sequence of 50 runs. The results are displayed in Table 1. Therein, "efficient" denotes the total number of necessarily efficient points and "sufficient" the number of them that were recognized by using the sufficient condition. Similarly, "non-efficient" denotes the number of necessarily non-efficient points and "necessary" the number of them that were recognized by using the necessary condition. Thus we have "efficient"+"non-efficient"$= 50(s+1)$.

Table 1: Performance of sufficient and necessary conditions.

n	s	R	sufficient	efficient	necessary	non-efficient
5	2	0.01	143	144	4	6
5	4	0.01	249	249	0	1
5	6	0.01	349	349	1	1
5	2	0.1	99	103	27	47
5	4	0.1	203	213	18	37
5	6	0.1	309	329	5	21
5	2	1	0	0	144	150
5	4	1	0	0	247	250
5	6	1	0	0	319	350
10	2	0.01	139	140	6	10
10	4	0.01	242	242	4	8
10	6	0.01	345	345	3	5
10	2	0.1	18	26	75	124
10	4	0.1	60	109	55	141
10	6	0.1	103	217	39	133
10	2	1	0	0	150	150
10	4	1	0	0	250	250
10	6	1	0	0	350	350
15	2	0.01	112	115	12	35
15	4	0.01	230	237	6	13
15	6	0.01	336	342	2	8
15	2	0.1	4	8	116	142
15	4	0.1	4	27	121	223
15	6	0.1	17	101	85	249
15	2	1	0	0	150	150
15	4	1	0	0	250	250
15	6	1	0	0	350	350

As long as the input intervals are tiny (R is small), the sufficient condition is very successful with the success rate almost 100%, but the necessary condition strikes approximately in 50%. Whenever the input intervals are wide enough then the situation is converse. In intermediate cases, both methods are successful in about 50%.

5 CONCLUSION

To accelerate methods for testing necessarily efficiency of x^* we proposed one sufficient condition and one necessary condition. Both methods work even when x^* is a degenerate solution. However, the former is provable effective provided x^* is non-degenerate, otherwise it may be costly. The latter is effective in any way.

The performed numerical experiments revealed that both methods are very effective, and the success rate is between 20% and 100%, depending mostly on the widths of input intervals.

REFERENCES

[Kuz, 1966] (1966). Algorithms for finding the general solution of a system of linear inequalities. *USSR Comput. Math. Math. Phys.*, 6(2):197–205.

[Alefeld and Herzberger, 1983] Alefeld, G. and Herzberger, J. (1983). *Introduction to interval computations.* Computer Science and Applied Mathematics. Academic Press, New York.

[Benjamin, 2002] Benjamin, M. R. (2002). *Interval Programming: A Multi-Objective Optimization Model for Autonomous Vehicle Control.* PhD thesis, Brown University, Providence, RI, USA.

[Benjamin, 2004] Benjamin, M. R. (2004). The interval programming model for multi-objective decision making. technical report AI Lab Technical Memo AIM-2004-021, Massachusetts Institute of Technology, Cambridge, MA, USA.

[Bitran, 1980] Bitran, G. R. (1980). Linear multiple objective problems with interval coefficients. *Manage. Sci.*, 26:694–706.

[Chanas and Kuchta, 1996] Chanas, S. and Kuchta, D. (1996). Multiobjective programming in optimization of interval objective functions – a generalized approach. *Eur. J. Oper. Res.*, 94(3):594–598.

[Chernikova, 1965] Chernikova, N. V. (1965). Algorithm for finding a general formula for the non-negative solutions of a system of linear inequalities. *USSR Comput. Math. Math. Phys.*, 5(2):228–233.

[Ehrgott, 2005] Ehrgott, M. (2005). *Multicriteria optimization. 2nd ed.* Springer, Berlin.

[Fiedler et al., 2006] Fiedler, M., Nedoma, J., Ramík, J., Rohn, J., and Zimmermann, K. (2006). *Linear optimization problems with inexact data.* Springer, New York.

[Gerlach, 1981] Gerlach, W. (1981). Zur Lösung linearer Ungleichungssysteme bei Störung der rechten Seite und der Koeffizientenmatrix. *Math. Operationsforsch. Stat., Ser. Optimization*, 12:41–43.

[Hladík, 2008] Hladík, M. (2008). Computing the tolerances in multiobjective linear programming. *Optim. Methods Softw.*, 23(5):731–739.

[Ida, 1996] Ida, M. (1996). Generation of efficient solutions for multiobjective linear programming with interval coefficients. In *Proceedings of the SICE Annual Conference SICE'96*, pages 1041–1044, Tottori.

[Ida, 1999] Ida, M. (1999). Necessary efficient test in interval multiobjective linear programming. In *Proceedings of the Eighth International Fuzzy Systems Association World Congress*, pages 500–504, Taipei.

[Ida, 2003] Ida, M. (2003). Portfolio selection problem with interval coefficients. *Appl. Math. Lett.*, 16(5):709–713.

[Ida, 2004] Ida, M. (2004). Solutions for the portfolio selection problem with interval and fuzzy coefficients. *Reliab. Comput.*, 10(5):389–400.

[Ida, 2005] Ida, M. (2005). Efficient solution generation for multiple objective linear programming based on extreme ray generation method. *Eur. J. Oper. Res.*, 160(1):242–251.

[Ida, 2007] Ida, M. (2007). Robust basis of interval multiobjective linear and quadratic programming. In *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making, MCDM 2007*, pages 38–41, Honolulu.

[Inuiguchi and Sakawa, 1994] Inuiguchi, M. and Sakawa, M. (1994). Possible and necessary optimality tests in possibilistic linear programming problems. *Fuzzy Sets Syst.*, 67(1):29–46.

[Inuiguchi and Sakawa, 1996] Inuiguchi, M. and Sakawa, M. (1996). Possible and necessary efficiency in possibilistic multiobjective linear programming problems and possible efficiency test. *Fuzzy Sets Syst.*, 78(2):231–241.

[Nožička et al., 1988] Nožička, F., Grygarová, L., and Lommatzsch, K. (1988). *Geometrie konvexer Mengen und konvexe Analysis.* Akademie-Verlag, Berlin.

[Oliveira and Antunes, 2007] Oliveira, C. and Antunes, C. H. (2007). Multiple objective linear programming models with interval coefficients – an illustrated overview. *Eur. J. Oper. Res.*, 181(3):1434–1463.

[Oliveira and Antunes, 2009] Oliveira, C. and Antunes, C. H. (2009). An interactive method of tackling uncertainty in interval multiple objective linear programming. *J. Math. Sci.*, 161(6):854–866.

[Padberg, 1999] Padberg, M. (1999). *Linear optimization and extensions. 2nd ed.* Springer, Berlin.

[Rockafellar and Wets, 2004] Rockafellar, R. T. and Wets, R. J.-B. (2004). *Variational analysis. corr. 2nd print.* Springer, Berlin.

[Sengupta and Pal, 2009] Sengupta, A. and Pal, T. K. (2009). *Fuzzy preference ordering of interval numbers in decision problems.* Springer, Berlin.

[Urli and Nadeau, 1992] Urli, B. and Nadeau, R. (1992). An interactive method to multiobjective linear programming problems with interval coefficients. *INFOR*, 30(2):127–137.

[Wang and Wang, 1997] Wang, H.-F. and Wang, M.-L. (1997). A fuzzy multiobjective linear programming. *Fuzzy Sets Syst.*, 86(1):61–72.

[Wang and Wang, 2001a] Wang, H.-F. and Wang, M.-L. (2001a). Decision analysis of the interval-valued multiobjective linear programming problems. In Köksalan, M. and Zionts, S., editors, *Multiple criteria decision making in the new millennium.*, volume 507 of *Lect. Notes Econ. Math. Syst.*, pages 210–218. Springer, Berlin.

[Wang and Wang, 2001b] Wang, H.-F. and Wang, M.-L. (2001b). Interval analysis of a fuzzy multiobjective linear program. *Int. J. Fuzzy Syst.*, 3(4):558–568.