

INTERVAL AND FUZZY MULTI-OBJECTIVE NON-LINEAR PROGRAMMING PROBLEM WITH NECESSARILY EFFICIENCY POINTS

M. Kiruthiga,
Assistant Professor,
Department of Mathematics,
Maharaja Arts and Science College,
Coimbatore, Tamilnadu, India.

Abstract: This paper presents the method for solving nonlinear programming problems with fuzzy numbers and multiple objectives. We transform these problems to multiobjective problem with fuzzy coefficient and crisp constraint by using α -cut and fuzzy ranking. Then, we define the necessarily efficient points for a new problem, then solve that problem by applying it and find all the necessarily efficient points.

Keywords: Multi-objective fuzzy programming, necessarily efficiency points, interval programming, and fuzzy ranking.

I. INTRODUCTION

In most real-world situations, the model coefficients are not exactly known because relevant data is in existent or scarce, difficult to find, the system is subject to change etc. Therefore, mathematical programming models for decision support must take explicitly into account, besides multiple and conflicting objective functions, the treatment of the intrinsic uncertainty associated with the model coefficients.

Interval programming is one of the approaches to trackle uncertainty in mathematical programming models. In fuzzy programming problems the constraints and goals are viewed as fuzzy sets and assumed that their membership functions are known. But, in reality, to decision maker (DM) it is not always easy to specify the membership function. Atleast in some of the cases, use an interval co-efficients may serve the purpose better. Though by using α - cuts, fuzzy numbers can be transformed into interval numbers.[9,12] Inuiguchi and Sakawa (1993); Steuer(1981); Wang, M.L., Wang, H.F., 2001b,[13,14] solved two different approaches to deal with an interval objective function. Bitran (1980)[1]; IdaM(2000); IdaM(1999), Ishibuchi H, SakawaM (1995), developed the concept of efficiency used in traditional MOLP to the interval objective function case.

In this context, there are two types of efficient solutions to the interval MOLP is used called "Necessarily Efficient" if it is efficient for any given objective function coefficient vectors within their admissible range of variation(Fig1). The approach presented in Ida (1999) proposed the enumeration of all possibly efficient solutions and all necessarily efficient solutions, the computational burden required is considerable.

In this paper, we consider the multi-objective non-linear programming with fuzzy coefficient. By using α - cuts and Ramik and Raminek ranking functions, we transform the

problem into a multiobjective non-linear programming with interval objective functions and crisp constraints then define necessarily efficient points and solve these points for problem with proposed algorithms. A simulation example is given and conclusion are presented.

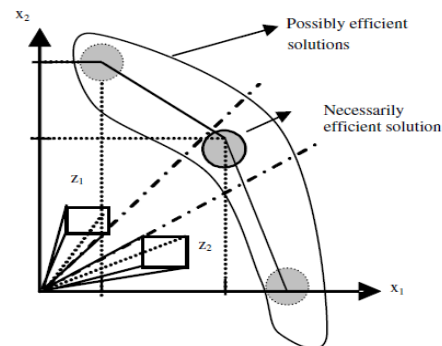


Figure 1: Necessarily and possibly efficient solutions

II. DEFINITIONS:

a) α - Level Set:

The α level set of a fuzzy set $\tilde{A}$ is defined as an ordinary set $\tilde{A}_\alpha$ for which the degree of its membership function exceeds the level α :

$$\tilde{A}_\alpha = \{x \mid \mu_{\tilde{A}}(x) \geq \alpha, \alpha \in [0, 1]\}$$

Observe that the α level set $\tilde{A}_\alpha$ can be defined by characteristic function

$$C_{\tilde{A}_\alpha} = \begin{cases} 1 & \mu_{\tilde{A}}(x) \geq \alpha \\ 0 & \mu_{\tilde{A}}(x) < \alpha \end{cases}$$

Actually, α – level set is an ordinary set whose elements belong to the corresponding fuzzy set to a certain degree α .

b) Fuzzy Number :

Fuzzy number A, it is significant to note that the α – level set A of a fuzzy number A can be represented by the closed interval which depends on interval value of α

Namely, $\tilde{A}_\alpha = \{x \mid \mu_{\tilde{A}} \geq \alpha\}$, $[A_\alpha^L, A_\alpha^U]$ where A_α^L or A_α^U represent the left or right extreme point of the α – level set $\tilde{A}_\alpha$ respectively.

III. MULTI OBJECTIVE NON-LINEAR PROGRAMMING WITH FUZZY COEFFICIENTS

The multi-objective non - linear programming with fuzzy coefficient can be formulated as follows[3,4]:

$$\begin{aligned} &\text{Maximize } \sum_j \tilde{c}_{rj} x_j^{aj} \quad r = 1, 2, 3, \dots, q \\ &\text{s.t: } \sum_j \tilde{a}_{ij} x_j \leq \tilde{b}_i \quad i = 1, 2, 3, \dots, m \\ &x_j \geq 0 \end{aligned}$$

,Where $\tilde{c}_{rj}$, $\tilde{a}_{ij}$ and $\tilde{b}_i$ are

fuzzy numbers.

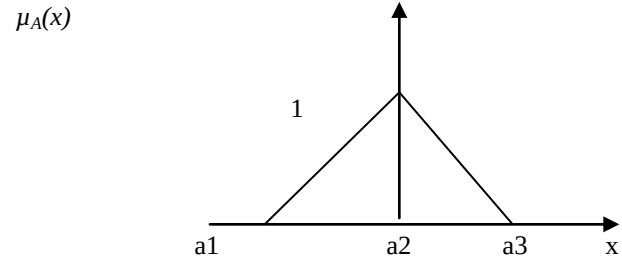
To find the necessarily efficiency points for problem 1, we transform in to a problem with interval objective function and crisp constraints. So we use α – cuts to transform coefficients to objective functions to interval, as indicated follow, Raminek ranking to obtain crisp constraints from our fuzzy constraints.

IV. TRIANGULAR FUZZY NUMBER

It is a fuzzy number represented with three points as follows: $A = (a_1, a_2, a_3)$

This representation is interpreted as membership functions

$$\mu_A(x) = \begin{cases} 0, & x < a_1 \\ \frac{x-a_1}{a_2-a_1}, & a_1 \leq x \leq a_2 \\ \frac{a_3-x}{a_3-a_2}, & a_2 \leq x \leq a_3 \\ 0, & x > a_3 \end{cases}$$



Now if you get crisp interval by α – cut operation, interval A_α shall be obtained as follows $\forall \alpha \in [0, 1]$. from

$$\frac{a_1^{(\alpha)} - a_1}{a_2 - a_1} = \alpha, \frac{a_3 - a_3^{(\alpha)}}{a_3 - a_2} = \alpha$$

We get

$$\begin{aligned} a_1^{(\alpha)} &= (a_2 - a_1)\alpha + a_1 \\ a_3^{(\alpha)} &= -(a_3 - a_2)\alpha + a_3 \end{aligned}$$

Thus

$$\begin{aligned} A_\alpha &= [a_1^{(\alpha)}, a_3^{(\alpha)}] \\ &= [(a_2 - a_1)\alpha + a_1, -(a_3 - a_2)\alpha + a_3] \end{aligned}$$

Especially if $\tilde{A} = (m, \gamma, \beta)$ is a triangular fuzzy number then

$$\begin{aligned} \tilde{A}_\alpha &= [\gamma(\alpha - 1) + m, m - \beta(\alpha - 1)] \quad \text{and} \quad \text{if} \\ \tilde{A} &= (m^L, m^R, \gamma, \beta) \text{ is trapezoidal fuzzy number then} \\ \tilde{A}_\alpha &= [\gamma(\alpha - 1) + m^L, m^R - \beta(\alpha - 1)]. \end{aligned}$$

V. RANKING FUZZY NUMBERS

Dubios and prade proposed a method of ranking fuzzy numbers as follow:

Definition let $\tilde{M}$ and $\tilde{N}$ be fuzzy numbers, then we have

$$\tilde{M} \geq \tilde{N} \Leftrightarrow \tilde{M} \square \tilde{N} = \tilde{M}$$

By usind the definition Ramik, and Raminek suggested the following lemma:

Lemma5.1 if $\tilde{M}$ and $\tilde{N}$ be a fuzzy numbers, $\tilde{M} \square \tilde{N} = \tilde{M}$ if and only if for every $h \in [0, 1]$ we have

$$\inf \{s : \mu_{\tilde{M}}(s) \geq h\} \geq \inf \{t : \mu_{\tilde{N}}(t) \geq h\}$$

$$\sup \{s : \mu_{\tilde{M}}(s) \geq h\} \geq \sup \{t : \mu_{\tilde{N}}(t) \geq h\}$$

Especially if $\tilde{M} = (m^L, m^R, \alpha, \beta)_{LR}$ and $\tilde{N} = (n^L, n^R, \gamma, \delta)_{LR}$ be two trapezoidal fuzzy numbers, the above relation is true if and only if

$$m^L - L^*(h) \alpha^L \geq n^L - L^*(h) \beta^L \quad \forall h \in [0, 1]$$

$$m^R + R^*(h) \alpha^R \geq n^R + R^*(h) \beta^R \quad \forall h \in [0, 1]$$

where

$$L^*(h) = \sup\{z : L(z) \geq h\}$$

$$L'^*(h) = \sup\{z : L'(z) \geq h\}$$

$$R^*(h) = \sup\{z : T(z) \geq h\}$$

$$R'^*(h) = \sup\{z : R'(z) \geq h\}$$

From the definition of Ramik and Raminek, if $\tilde{M} = (m, \alpha, \beta)_{LR}$ and $\tilde{N} = (n, \gamma, \delta)_{LR}$, be triangular fuzzy numbers, then we have

$$\tilde{M} \leq \tilde{N} \leftrightarrow \begin{cases} m \leq n \\ m - \alpha \leq n - \gamma \\ m + \beta \leq n + \delta \end{cases}$$

if $\tilde{M} = (m^L, m^R, \alpha, \beta)_{LR}$ and $\tilde{N} = (n^L, n^R, \gamma, \delta)_{LR}$ be two fuzzy numbers, similarly we can compare them as follow

$$\tilde{M} \leq \tilde{N} \leftrightarrow \begin{cases} m^L \leq n^L \\ m^R \leq n^R \\ m^L - \alpha \leq n^L - \gamma \\ m^R + \beta \leq n^R + \delta \end{cases}$$

By using above definition and with respect to the kind of fuzzy numbers of the problem, we can transform the fuzzy constraints to crisp ones.

In this way, if $\tilde{c}_{kj} = (c_{kj}, \alpha_{kj}, \beta_{kj})$ and $\tilde{a}_{ij} = (a_{ij}, \gamma_{ij}, \delta_{ij})$ and $\tilde{b}_i = (b_i, n_i, v_i)$ be triangular fuzzy numbers, problem (1) is equivalent to

$$\begin{aligned} & \text{maximize} \sum_{j=1}^n [\alpha_{kj}(\alpha - 1) + c_{kj}, c_{kj} - \beta_{kj}(\alpha - 1)]x_j \\ & k = 1, \dots, p \quad \text{s.t} \\ & \sum_{j=1}^n a_{ij}x_j \leq b_i \quad i = 1, \dots, m \\ & \sum_{j=1}^n (a_{ij} - \gamma_{ij})x_j \leq b_i - \eta_i \quad i = 1, \dots, m \\ & \sum_{j=1}^n (a_{ij} + \beta_{ij})x_j \leq b_i + v_i \quad i = 1, \dots, m \\ & x_j \geq 0. \end{aligned} \quad (2)$$

Similarly if $\tilde{c}_{kj} = (c_{kj}^L, c_{kj}^R, \alpha_{kj}, \beta_{kj})$, $\tilde{a} = (a_{ij}^L, a_{ij}^R, \gamma_{ij}, \delta_{ij})$ and $\tilde{b}_i = (b_i^L, b_i^R, \eta_i, v_i)$ be trapezoidal fuzzy numbers then, problem (1) is equivalent to

$$\begin{aligned} & \text{Maximize} \sum_{j=1}^n [\alpha_{kj}(\alpha - 1) + c_{kj}^L, c_{kj}^R - \beta_{kj}(\alpha - 1)]x_j \\ & k = 1, \dots, p \text{ s.t} \\ & \sum_{j=1}^n a_{ij}^L x_j \leq b_i^L \quad i = 1, \dots, m \\ & \sum_{j=1}^n a_{ij}^R x_j \leq b_i^R \quad i = 1, \dots, m \\ & \sum_{j=1}^n (a_{ij}^L - \gamma_{ij})x_j \leq b_i^L - \eta_i \quad i = 1, \dots, m \end{aligned} \quad (3)$$

$$\sum_{j=1}^n (a_{ij}^R + \beta_{ij})x_j \leq b_i^R + v_i \quad i = 1, \dots, m \quad x_j \geq 0.$$

VI. NECESSARILY EFFICIENT :

Lemma 6.1.[10] *The inequality $Cx \geq 0$ is true for some $C \in C$ if and only if $C^c x + C^A |x| \geq 0$.*

Proof:

It is a slight modification of Gerlach theorem].

If $Cx \geq 0$, then $0 \leq C^c x + (C - C^c)x \leq C^c x + |C - C^c||x| \leq C^c x + C^A |x|$

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

$$0 \leq C^c x + C^A \text{diag}(z)x = (C^c + C^A \text{diag}(z))x.$$

Since $C := C^c x + C^A \text{diag}(z)x \in C$, the proof is completed.

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

Theorem 6.1. The vector x^* is necessarily efficient if and only if the system

$$C^c x + C^A |x| \geq 0, Apx \leq 0, e^T |x| = 1 \quad (4)$$

has no solution.

Proof.

The vector x^* is efficient to (1) with fixed $C \in C$ iff

$$Cx \geq 0, Apx \leq 0 \quad (5)$$

has no solution. Thus x^* is necessarily efficient iff there is no $C \in C$ such that (5) is solvable. By Lemma 1, this is true iff

$$C^c x + C^A |x| \geq 0, Apx \leq 0$$

is not solvable. Using L^1 - norm to normalize x we obtain the final form of (4).

Observe that Theorem 1 gives rise to a simple but expensive algorithm for testing necessarily efficiency. Decomposing (4) according to signs of particular x_i -s we can provide the testing by solving up to 2^n linear programs. System (4) is solvable iff the systems

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

is solvable, or equivalently, iff the linear system

$$C^c x + C^A \text{diag}(s)x - y \geq 0, Apx \leq 0, \text{diag}(s)x \geq 0, y \geq 0, e^T y \geq 1 \quad (6)$$

is solvable for all $s \in \{\pm 1\}^n$ here in, $|x_i|$ 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 $Apx \leq 0$ contains some non-negativity constraints $x_i \geq 0$, $i \in I$ for certain $I \subseteq \{1, \dots, n\}$; 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 (6) 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.

VII. NECESSARILY EFFICIENT POINTS

Theorem: 7.1 [necessary Condition]

Let $k \in p$ put $d = Ak, *$ - $\Sigma i \in p, i \neq k$ $Ai, *$ and define $i \in c$ column_wise as follows

$$C_{*,j}^0 = \begin{cases} \bar{C}_{*,j} & \text{if } d_j \geq 0, \\ C_{*,j} & \text{otherwise,} \end{cases} \quad (7)$$

$i = 1, \dots, n$, if the linear system $C^0 x^a - y \geq 0, A_{px} \leq 0, y \geq 0$ is solvable then x^* is not necessarily $e_y^T = 1$ efficient.

Proof:

from definition, C^0 comes always from C . the print x^* is efficient to $\text{Max } cx^a$ for the objective matrix C^0 if and only if $x \in m$

the linear system (7) is not solvable. If it turns out that x^* is not efficient for a particular $C^0 \in C$ then it cannot be necessarily efficient.

Consider the following MONLP problem with interval objective functions:

$$\text{Max } z(x) = Cx^a$$

$$\text{s.t. } Ax \leq b$$

$$x \geq 0$$

$$C \in \Phi$$

Where Φ is a set of $p \times n$ matrices, with elements $C_{kj} \in [C_{kj}^L, C_{kj}^U]$ for $k = 1, \dots, p, j = 1, \dots, n$, A is a $m \times n$ matrix and b is a $m \times 1$ vector.

A solution is necessarily efficient to problem (4) if and only if it is efficient for any $C \in \Phi$. The necessarily efficient solution set (N_E) is obtained by $N_E = \bigcap_{C \in \Phi} X_E(C)$

$X_E(C)$ is the efficient solution set for each $C \in \Phi$.

Bitran (1980) [1] proposed an implicit enumeration algorithm that uses a sub problem to test efficiency of a given basic solution and a branch and bound algorithm to solve sub problem. But his method may result in a high computational burden if the solution being analyzed is not necessarily efficient.

Ida (1999) suggested an extension of the implicit enumeration algorithm the proposed method uses two efficiency test. One of them checks necessary efficiency and the other checks non-necessary efficiency.

Let $R^{(g, w_1, \dots, w_g)} = C_B^{(g, w_1, \dots, w_g)} B^{-1} N - C_N^U$, where the columns of the interval matrix, $C_B^{(g, w_1, \dots, w_g)}$ are defined as

$$C_B^{(g, w_1, \dots, w_g)} = \begin{cases} C_{B,j}^U & j \leq g, w_j = U \\ C_{B,j}^L & j \leq g, w_j = L \end{cases}$$

$$[C_{B,j}^L, C_{B,j}^U] \quad g \leq j \leq m$$

In this context, g is the tree level, m is the number of basic variables with interval coefficients, w_j is $L(U)$ if the column elements of the matrix within the tree level j are in the lower bounds $C_{B,j}^L$ (upper bounds $C_{B,j}^U$) of the intervals and $j = 1, \dots, g$.

Let $R^{L(g, w_1, \dots, w_g)}$ and $R^{U(g, w_1, \dots, w_g)}$ be composed of the lower and upper bounds of each element belonging to the interval matrix $R^{(g, w_1, \dots, w_g)}$, respectively. The operations “sc” is defined as

$$\text{sc}(R^{L(g, w_1, \dots, w_g)}) = \{ R^{L(g+1, w_1, \dots, w_g, L)}, R^{L(g+1, w_1, \dots, w_g, U)} \}$$

$$\text{and } \text{sc}(R^{U(g, w_1, \dots, w_g)}) = \{ R^{U(g+1, w_1, \dots, w_g, L)}, R^{U(g+1, w_1, \dots, w_g, U)} \}$$

VIII. NECESSARY

EFFICIENCY

PROGRAMMING TECHNIQUE

The necessary efficiency test is obtained in the following way[11]:

Procedure – 1

Step1. Let $S^L = \{R^{L(0)}\}$

Step2. Select one element $R^{L(g, w_1, \dots, w_g)}$ from S^L and check whether it is efficient,

- If it is efficient then remove the element from S^L .
- Otherwise, add $\text{sc}(R^{L(g, w_1, \dots, w_g)})$ to S^L . If $R^{(g, w_1, \dots, w_g)} = (R^{(m, w_1, \dots, w_m)})$, then R is not necessarily efficient.

Step3. If the set S^L is empty, then R is necessarily efficient.

Step4. Return to Step2.

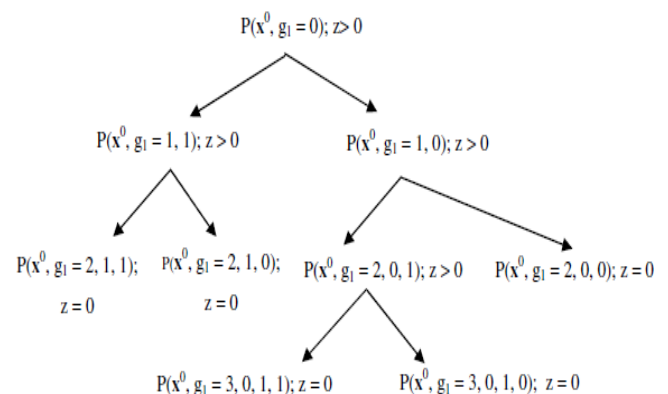


Figure 2: Example of tree generated by the implicit enumeration algorithm proposed by Bitran.

This procedure is based on the following theorem:

Theorem If all of the elements of S^L are efficient, then R is necessarily efficient. (Ida, 1999)

If the solution being analyzed is not necessarily efficient, his implicit enumeration algorithm may be out of acceptable computational limits due to the branching required. Therefore, Ida proposed another algorithm in the following way.

Procedure – 2

Step1. Let $S^U = \{R^{U(0)}\}$

Step2. Select one element $R^{U(g, w_1, \dots, w_g)}$ from S^U and check whether it is efficient,

- If it is not efficient, then R is not necessarily efficient.
- Otherwise, add $sc(R^{U(g, w_1, \dots, w_g)})$ to S^U . If $(R^{(g, w_1, \dots, w_g)}) = (R^{(m, w_1, \dots, w_m)})$. Do not add anything to S^U .

Step3. If the set S^U is empty, then R is necessarily efficient.

Step4. Return to Step2.

This procedure is based on the following theorem (Ida, 1999)[6]:

Theorem If there is an element of S^U that it is not efficient, then R is not efficient. The efficiency is based on *chernikova's* algorithm (*chernikova*, 1999) and proceeds as follows:

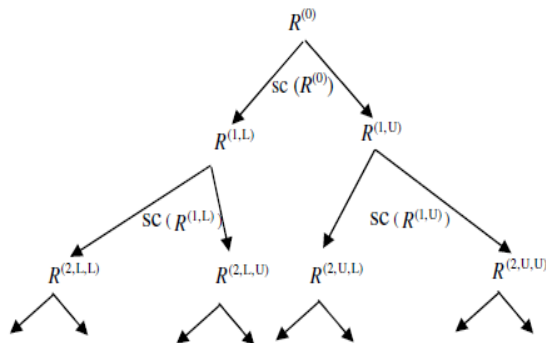


Figure 3: Illustration of the branching process with the operator “sc”

Procedure – 3

Step1. Compute R .

Step2. Analyze column and rows of R and proceed as follow:

- If there are any columns in R such that $R_j \geq 0$, then eliminate these columns.
- If there are any rows in R such that $R_L \leq 0$, then eliminate these rows.

Step3. Analyze columns and rows of R and proceed as follow:

- If there is a column in R such that $R_j \leq 0$, then R is not efficient.
- If there is a row in R such that $R_L > 0$, then R is efficient.
- If there is a row in R such that $R_L \geq 0$ and a row $i' \neq i$ such that $R_{ij} > 0$ ($R_{ij} = 0$), then R is efficient.

Step4. Calculate the summation of the columns (R_{Σ}) and rows (R_{Σ}) of R .

(a) If $R_{\Sigma} \leq 0$, then R is not efficient.

(b) If $R_{\Sigma} > 0$, then R is efficient.

Step5. Let $D = -R$ and process the rows by using the extreme ray generation method (*chernikova*, 1965)[2]. Let, simultaneously, $D = R^T$ and process the row in parallel by using extreme ray generation method (D is defined in *chernikova* (1965) and Ida (2000b))[7,8].

Step6. Return to step2.

Since $R^{(g+1, \omega_1, \dots, \omega_g, \omega_{g+1})} \subset R^{(g, \omega_1, \dots, \omega_g)}$, if the efficiency condition $(C_N - C_B B_N^{-1}) \mu \geq 0, \mu \geq 0, A = (N, B)$

holds for the entire set of matrices contained in $R^{L(g, \omega_1, \dots, \omega_g)}$, then

$R^{L(g, \omega_1, \dots, \omega_g)}$ is efficient and $R^{(g, \omega_1, \dots, \omega_g)}$ is necessarily efficient. Namely, if $R^{L(0)}$ is efficient, then the matrices obtained through the branching process ($sc(R^{L(0)})$) are necessarily efficient.

Therefore, considering the operator “sc” sequentially as shown in Fig. 5, the entire set of matrices obtained after branching is efficient, and, hence, R is necessarily (Ida, 1999). Analogously,

if $R^{U(g, \omega_1, \dots, \omega_g)}$ is not efficient then $R^{(g+1, \omega_1, \dots, \omega_g, L)}$ and $R^{(g+1, \omega_1, \dots, \omega_g, U)}$ are not necessarily efficient (Ida, 1999).

Step1. Let $S^L = \{R^{L(0)}\}$

Step2. Select one element from S^L and check whether it is efficient,

(a) If it is efficient then remove the elem $R^{(g, \omega_1, \dots, \omega_g)}$ ent from S^L .

(b) Otherwise, add $sc(R^{L(g, \omega_1, \dots, \omega_g)})$ to S^L . If $(R^{(g, \omega_1, \dots, \omega_g)}) = (R^{(m, \omega_1, \dots, \omega_m)})$, then R is not necessarily efficient.

Step3. If the set S^L is empty, then R is necessarily efficient.

Step4. Return to Step2.

IX. NUMERICAL EXAMPLE:

Consider the following MONLP with fuzzy Coefficients[5]:

$$\text{Max} = \tilde{C}x^a$$

$$\text{s.t. } \bar{A}x \leq \bar{b}$$

$$x \geq 0$$

$$\text{where, } \tilde{C} = \begin{bmatrix} (5,2) & (10,0.6) \\ (1,0.3) & (3,2) \end{bmatrix}, \quad \bar{A} =$$

$$\begin{bmatrix} (1,0.6) & (3,0.5) \\ (2,1) & (3,0.5) \end{bmatrix}$$

Let $\alpha = 0.6$, then we have the following interval for objective coefficients[14]

$$\begin{aligned} (5, 2)_{0.6} &= [4.7, 5.3] \\ (10, 0.7)_{0.6} &= [6.78, 7.22] \\ (2, 0.3)_{0.6} &= [1.78, 2.13] \\ (2, 2)_{0.6} &= [2.5, 3.5] \end{aligned}$$

If we want to test the necessary efficiency of the extreme point $(\frac{10}{13}, 0, \frac{6}{13}, 0, \frac{12}{13})$, then

$$\text{We obtain } B^{-1} = \begin{bmatrix} 0 & 0.6692 & 0 \\ 1 & -1.5231 & 0 \\ 0 & -2.1621 & 1 \end{bmatrix}, N =$$

$$\begin{bmatrix} 2.9 & 0 \\ 2.3 & 1 \\ 3.6 & 0 \end{bmatrix}$$

$$\text{and } C_N^U = \begin{bmatrix} 10.24 & 0 \\ 3.95 & 0 \end{bmatrix}$$

$$\text{Thus, } R^{(0)} = C_B^{(0)} B^{-1} N - C_N^U R(0) =$$

$$\begin{bmatrix} [4.7, 5.3] & 0 & 0 \\ [1.78, 2.13] & 0 & 0 \end{bmatrix}$$

$$B^{-1}N - C_N^U$$

$$=$$

$$\begin{bmatrix} [-1.1627, -0.01623] & [3.5234, 4.16537] \\ [-0.67325, -0.23323] & [1.335087, 1.530835] \end{bmatrix}$$

Hence,

$$R^{U(0)} = \begin{bmatrix} -0.01623 & 4.16537 \\ -0.23323 & 1.530835 \end{bmatrix} \text{ and}$$

$$R^{L(0)} = \begin{bmatrix} -1.1627 & 3.5234 \\ -0.67325 & 1.335087 \end{bmatrix}$$

Since $R^{U(0)} < 0$, then R is not efficient. Therefore, by the two efficiency tests, it may be necessary to examine the efficiency of both $R^{(1,U)}$ and $R^{(1,L)}$.

X. CONCLUSION:

By using the necessary efficiency algorithm, we get a new solution to a nonlinear multi-objective programming problems with fuzzy co-efficient. The method to solve the necessarily efficiency points is indicated.

REFERENCES

- [1] Bitran G.R., linear multiple objective problems with interval coefficients, Management Science 26, pp. 694-706, **1980**.
- [2] Chemicova N.V., Algorithm for finding a general formula for nonnegative solutions of a system of linear inequalities, USSR. Computational Mathematics and Mathematical Physics 5, pp 228-233, **1965**.
- [3] P.Durga Prasad Dash and RajaniB.Dash, Solving Multi Objective Fuzzy Fractional Programming Problem Ultra Scientist Vol: 24(3)A, 429-434 (2012).
- [4] P.Durga Prasad Dash and RajaniB.Dash, Solving Fuzzy Multi Objective Non-linear Programming Problem Using Fuzzy Programming Technique, International Journal of Engineering Science and Innovative Technology, Vol: 2, Issue 5, September 2013.
- [5] Hassan Mishmast Nehi and Marzieh Alineghad., Solving Interval and Fuzzy Multi Objective Linear Programming Problemly Necessarily Efficiency Points ,International Mathematical Forum, 3, 2008, no. 3, 99 – 106.
- [6] Ida M., Necessary efficient test in interval multi objective linear programming, In: proceedings of 8th international fuzzy systems association world congress, pp.500- 504, **1999**.
- [7] Ida M., Efficient solution generation for multiple objective linear programming and uncertain coefficients, In: proceedings of 8th Bellman continuum, pp.132-136, **2000**.
- [8] Ida M., Interval multiobjective programming and mobile robot path planning, In: Mohammadian M. (Eds.), New Frontiers in computational Intelligence and its applications. ISO press, pp 313-322, **2000**.
- [9] Ishibuchi H., Tanaka H., Multiobjective programming in optimization of the interval objective Systems, Prentic –Hall, Englewood Cliffs, NJ, **1982**.
- [10] Milan Hlad'ik., On necessarily efficient solutions in interval multiobjective linear Programming, URPDM 2010.
- [11] Oliveira C., Antunes C.H., Multiple objective linear programming modelos with interval coefficients an illustrated overview, European journal of operational Research, pp.1-30, **2006**.
- [12] Sakawa M., Fuzzy sets and interactive multiobjective optimization, Plenum press, New York, **1993**.
- [13] Steuer P.E., Multiple criteria optimization: theory, computation and application, John wiley & Sons, New york, **1986**
- [14] Wang, M.L., Wang, H.F., 2001b. Interval analysis of a fuzzy multiobjective linear programming. International Journal of Fuzzy Systems, 34, 558–568.