# Software for optimization of linear objective function with fuzzy relational constraint

Ketty Peeva, Zlatko Zahariev and Ivajlo Atanasov

***Abstract*— The paper presents method, algorithm and software in MATLAB and Java for solving various optimization problems when the linear objective function is subject to fuzzy linear system of equations or fuzzy linear systems of inequalities as constraint.**

***Index Terms*— fuzzy linear systems of equations, fuzzy linear systems of inequalities, linear optimization**

## I. INTRODUCTION

This paper deals with optimization problem when the linear objective function with traditional addition and multiplication

$$Z = \sum_{j=1}^{n} c_j x_j, \; c_j \in R, 0 \le x_j \le 1, 1 \le j \le n, \tag{1}$$

is subject to fuzzy linear system of equations or inequalities as constraint

$A \bullet X = B$ (2) or $A \bullet X \ge B$, (3)

where $A = (a_{ij})_{m \times n}$ stands for the matrix of coefficients, $X = (x_j)_{n \times 1}$ stands for the matrix of unknowns, $B = (b_i)_{m \times 1}$ is the right-hand side of the system and for each $i$, $1 \le i \le m$ and for each $j$, $1 \le j \le n$, we have $a_{ij}$, $b_i$, $x_j \in [0,1]$. The max-min composition is written as $\bullet$ and $c = (c_1, ..., c_n)$ is the weight (cost) vector. The aim is to minimize or maximize (1) subject to constraint (2) or (3).

The linear optimization problem is investigated by many authors [4], [6], [8], [14], [15], [16]. Comparison between their and our results is given in Section VI of this paper.

In this paper we propose unified method, algorithm and software to find the optimal solution of (1) when the composition is max-min with straightforward search among the extremal solutions of constraint (2) or (3).

Terminology for computational complexity and algorithms is as in [5], for fuzzy sets and fuzzy relations is according to [3], [7], [12], [13], for algebra - as in [9].

Ketty Peeva is with the Department of Algebra and Geometry, Technical University of Sofia, 8, Kliment Ohridski St, Sofia 1000, Bulgaria (telephone: +359 2 9653351, e-mail: *kgp@tu-sofia.bg*).

Zlatko Zahariev is with the Department of Algebra and Geometry, Technical University of Sofia, 8, Kliment Ohridski St, Sofia 1000, Bulgaria (telephone: +359 2 9653427, e-mail: *zlatko@tu-sofia.bg*).

Ivajlo Atanasov is with the Department of Algebra and Geometry, Technical University of Sofia, 8, Kliment Ohridski St, Sofia 1000, Bulgaria (telephone: +359 2 9653379, e-mail: *ivcho_f22@yahoo.com*).

## II. BASIC NOTIONS

The basic notions are given according to [1], [7]. Partial order relation on a partially ordered set (poset) $P$ is denoted by the symbol $\le$. By a *greatest element* of $P$ we mean an element $b \in P$ such that $x \le b$ for all $x \in P$. The *least element* of $P$ is defined dually. The (unique) least and greatest elements of $P$, when they exist, are called *universal bounds* of $P$ and are denoted by 0 and 1, respectively.

We work in the *fuzzy algebra* $\mathbb{I} = \langle [0,1], \vee, \wedge, 0, 1 \rangle$, where $[0,1]$ is the real closed unit interval, with operations $\vee$, $\wedge$, respectively defined by $a \vee b = max\{a,b\}$, $a \wedge b = min\{a,b\}$.

In $\mathbb{I}$ the operation $\alpha$ (cf. [13]) is defined as follows:

$$a \alpha b = \begin{cases} 1, & if \; a \le b \\ b, & if \; a > b \end{cases}.$$

A matrix $A = (a_{ij})_{m \times n}$, with $a_{ij} \in [0,1]$ for each $i, j$, $1 \le i \le m$, $1 \le j \le n$, is called *membership matrix* [9]. We write "matrix" instead of "membership matrix".

**Definition 1** Let the matrices $A = (a_{ij})_{m \times p}$ and $B = (b_{ij})_{p \times n}$ be given.

i. The matrix $C_{m \times n} = (c_{ij}) = A \bullet B$ is called *max-min product* of $A$ and $B$ if

$$c_{ij} = \max_{k=1}^{p}(\min(a_{ik}, b_{kj})), \text{ when } 1 \le i \le m, 1 \le j \le n.$$

ii. The matrix $C_{m \times n} = (c_{ij}) = A \alpha B$ is called $\alpha$-*product* of $A$ and $B$ if

$$c_{ij} = \min_{k=1}^{p}(a_{ik} \alpha b_{kj}), \text{ when } 1 \le i \le m, 1 \le j \le n.$$

**Theorem 1** [13] Let $A = (a_{ij})_{m \times p}$ and $C = (c_{ij})_{m \times n}$ be given matrices and $\mathbb{B}_{\bullet}$ be the set of all matrices such that $A \bullet B = C$. Then:

i. $\mathbb{B}_{\bullet} \ne \varnothing$ if $A^t \alpha C \in \mathbb{B}_{\bullet}$.

ii. If $\mathbb{B}_{\bullet} \ne \varnothing$ then $A^t \alpha C$ is the greatest element in $\mathbb{B}_{\bullet}$.

978-1-4244-1739-1/08/$25.00 © 2008 IEEE

### III. Solving Fuzzy Linear Systems

#### A. Solution Set

We first give the solution set of fuzzy linear systems (FLS) of equations (FLSE) $A \bullet X = B$, or fuzzy linear systems of inequalities (FLSI) $A \bullet X \geq B$, that provides solving optimization problem; $A$, $X$, $B$ are described in the Introduction and the *max-min* composition is written as $\bullet$.

For $X = (x_j)_{n\times 1}$ and $Y = (y_j)_{n\times 1}$ the inequality $X \leq Y$ means $x_j \leq y_j$ for each $j$, $1 \leq j \leq n$.

**Definition 2** Let the system (2) or (3) be given.

i. $X^0 = (x_j^0)_{n\times 1}$ with $x_j^0 \in [0,1]$, when $1 \leq j \leq n$, is called a *solution* of (2) (of (3), resp.) if $A \bullet X^0 = B$ ( $A \bullet X^0 \geq B$, resp.) holds.

ii. The set of all solutions $\mathbb{X}^0$ is called *complete solution set.*

iii. A solution $X^0_{\text{low}} \in \mathbb{X}^0$ is called *minimal solution* if for any $X^0 \in \mathbb{X}^0$ the relation $X^0 \leq X^0_{\text{low}}$ implies $X^0 = X^0_{\text{low}}$. Dually, a solution $X^0_{\text{u}} \in \mathbb{X}^0$ is called *maximal solution* if for any $X^0 \in \mathbb{X}^0$ the relation $X^0_{\text{u}} \leq X^0$ implies $X^0 = X^0_{\text{u}}$. When the maximal solution is unique, it is called *greatest* (or *maximum*) solution.

iv. If $\mathbb{X}^0 \neq \varnothing$ then the system is called *consistent*, otherwise it is called *inconsistent*.

We consider inhomogeneous systems ( $B = 0$ makes the problem uninteresting).

Any consistent system has unique maximum solution $X_{\text{gr}}$:

i. $X_{\text{gr}} = A^t \alpha B$ for (2), see Theorem 1;

ii. $X_{\text{gr}} = (1,...,1)^t$ for (3).

The solution set of (2) or (3) is determined by all minimal solutions and the maximum one.

The conventional investigations [6], [8], [15] for optimization of the objective function start with finding feasible domain and one arbitrary solution of the system and proceed with the solution set during the search for minimal solutions (because up to now there do not exist methods and software for straight computing of extremal solutions). Since the solution set of the system has some specific properties, an arbitrary solution may proceed to one or more minimal solutions and this implies vague search at random. We propose a more clever choice of the objects over which the search is performed: straightforward search among the extremal solutions of the system. Thus the search space of the objective function is pruned.

In order to make the exposition clear, we give here details for solving fuzzy linear systems.

**Theorem 2** [13] The system $A \bullet X = B$ is consistent if and only if $\hat{X} = (\hat{x}_1, \hat{x}_2, ..., \hat{x}_n)^t = A^t \alpha B$ is its solution.

**Theorem 3** The system $A \bullet X \geq B$ is consistent if and only if $\hat{X} = (\hat{x}_1, \hat{x}_2, ..., \hat{x}_n)^t = (1,...,1)^t$ is its solution.

If the system is consistent, according to Theorems 2, 3, it has unique maximum solution $\hat{X} = X_{\text{gr}}$. The complete solution set is described by the set of all maximal interval solutions. They are determined by all minimal solutions and the maximum one, see [12]. Since there exists analytical expression for the maximum solution, attention in references is paid on computing minimal solutions.

We describe steps for simplifying systems in order to find the minimal solutions easier.

#### B. Associated Matrix

For the system (2) ((3), respectively) any coefficient $a_{ij} \geq b_i$ provides a way to satisfy the $i$-th equation (inequality, respectively) with $\hat{x}_j$ determined by Theorem 2 (Theorem 3, respectively). This leads to the idea to distinguish coefficients that contribute for solving the system from these that do not contribute.

We assign to $A \bullet X = B$ ( $A \bullet X \geq B$, respectively) a matrix $A^* = (a^*_{ij})$ with elements $a^*_{ij}$ determined according (4a) (or (4b), respectively):

$$a^*_{ij} = \begin{cases} b_i, \text{ if } a_{ij} \wedge \hat{x}_j = b_i, \\ 0, \quad \text{otherwise} \end{cases} \tag{4a}$$

$$a^*_{ij} = \begin{cases} 1, & \text{if } a_{ij} > b_i \\ b_i, & \text{if } a_{ij} = b_i \\ 0, & \text{otherwise} \end{cases}. \tag{4b}$$

The matrix $A^*$ with elements $a^*_{ij}$, determined by (4a) or (4b), is called *associated matrix* of the system (2) ((3), respectively).

#### C. IND vector

The vector $IND = IND_{m\times 1}$ is used to indicate consistency of the system. The components of $IND$ are computed from $A^*$. Let $IND(i)$ equals the number of elements $a^*_{ij} \neq 0$ in the $i$-th row of $A^*$.

**Lemma 1** Let the system (2) ((3), respectively) be given.

i If $IND(i) = 0$ for at least one $i = 1,...,m$ then the system is inconsistent.

ii If $IND(i) \neq 0$ the system may be consistent and the number of its potential minimal solutions does not exceed

$$PN = \prod_{i=1}^{m} IND(i). \tag{5}$$

#### D. Solving Fuzzy Linear Systems

We propose a unified and exact method and algorithm for solving inhomogeneous FLS, resulting in:

i. A necessary and sufficient condition (Corollary 5) for consistency of the FLS.
ii. Analytical expressions for the minimal solutions.
iii. Algorithm for solving FLS.

Let the following stipulations be satisfied for FLS:

1. The FLS has coefficient matrix $A = (a_{ij})_{m\times n}$, matrix of unknowns $X = (x_j)_{n\times 1}$, and right-hand side $B = (b_i)_{m\times 1}$.
2. The associated matrix $A^*$ for the FLS is obtained.
3. Any coefficient $a^*_{ij} = 0$ is called S-type coefficient, any $a^*_{ij} = b_i$ is called E-type coefficient and any $a^*_{ij} = 1$ is called G-type coefficient.
4. For each $j$, $j = 1,...,n$, $A^*(j) = (a^*_{ij})_{m\times 1}$ denotes the $j$-th column of $A^*$ and $a^*_{ij}$ denotes the $i$-th element ($1 \le i \le m$) in $A^*(j)$.

*E. Main Theoretical Results*

We give the main theoretical results that provide the software for solving FLS.

**Theorem 4** [11] Let the system $A \bullet X = B$ be given.

i. If $A^*(j)$ contains G-type coefficient $a^*_{kj} = 1$ and $k$ ($1 \le k \le m$) is the greatest number of the row with $a^*_{kj} = 1$ in $A^*(j)$, then
   a. For each $i$, $1 \le i \le m$, $a^*_{ij}$ in $A^*(j)$ and $x_j \in [0, b_k]$ imply $a^*_{ij} \wedge x_j \le b_i$.
   b. $a^*_{ij}$ in $A^*(j)$ and $x_j = b_k$ imply $a^*_{ij} \wedge x_j = b_i$:
      * $\forall i$, $1 \le i \le k$ with $a^*_{ij} \ge b_i = b_k$;
      * $\forall i$, $k < i \le m$ with $a^*_{ij} = b_i$.
ii. If $A^*(j)$ does not contain any G-type coefficient, but it contains E-type coefficient $a^*_{rj} = b_r$ and $r$ ($1 \le r \le m$) is the smallest number of the row with $a^*_{rj} = b_r$ in $A^*(j)$, then
   a. $\forall i$, $1 \le i \le m$, $a^*_{ij}$ in $A^*(j)$ and $x_j \in [0,1]$, imply $a^*_{ij} \wedge x_j \le b_i$;
   b. $\forall i$, $r \le i \le m$ with $a^*_{rj} = b_r$ and $x_j \in [b_r, 1]$, imply $a^*_{ij} \wedge x_j = b_i$.
iii. If $A^*(j)$ contains neither G-type nor E-type coefficient then $\forall i$, $1 \le i \le m$, $a^*_{ij}$ in $A^*(j)$ and $x_j \in [0,1]$ imply $a^*_{ij} \wedge x_j < b_i$.

**Theorem 5** [12] Let the system $A \bullet X \ge B$ be given.

i. If $A^*(j)$ contains G-type coefficient $a^*_{kj} = 1$ and $k$ ($1 \le k \le m$) is the least number of the row with $a^*_{kj} = 1$ in $A^*(j)$, then $\forall i$, $k \le i \le m$, $a^*_{ij}$ in $A^*(j)$ and $x_j \in [0, b_k]$ imply $a^*_{ij} \wedge x_j \ge b_i$.
ii. If $A^*(j)$ does not contain any G-type coefficient, but it contains E-type coefficient $a^*_{rj} = b_r$ and $r$ ($1 \le r \le m$) is the smallest number of the row with $a^*_{rj} = b_r$ in $A^*(j)$, then
   a. For each $i$, $1 \le i \le m$, $a^*_{ij}$ in $A^*(j)$ and $x_j \in [0,1]$, imply $a^*_{ij} \wedge x_j \le b_i$;
   b. For each $i$, $r \le i \le m$ with $a^*_{rj} = b_r$ and $x_j \in [b_r, 1]$, imply $a^*_{ij} \wedge x_j = b_i$.
iii. If $A^*(j)$ contains neither G-type nor E-type coefficient then for each $i, 1 \le i \le m$, $a^*_{ij}$ in $A^*(j)$ and $x_j \in [0,1]$ imply $a^*_{ij} \wedge x_j < b_i$.

**Corollary 1**

i. For any consistent system $A \bullet X = B$, $X_{\text{gr}} = A^t \alpha B = \hat{X} = (\hat{x}_j)_{n\times 1}$ and $\hat{x}_j$, $1 \le j \le n$, are computed according to Theorem 4.
ii. For any consistent system $A \bullet X \ge B$, $X_{\text{gr}} = (1,...,1)^t = \hat{X} = (\hat{x}_j)_{n\times 1}$ and $\hat{x}_j$, $1 \le j \le n$, are computed according to Theorem 5.

**Corollary 2** If $a^*_{ij} = 0$ for each $i = 1,...,m$, then $\breve{x}_j = 0$ in any minimal solution $\breve{X} = (\breve{x}_j)_{n\times 1}$ of the consistent FLS.

**Corollary 3** If $\breve{X} = (\breve{x}_j)_{n\times 1}$ is a minimal solution of the consistent FLS, then for each $j = 1,...,n$ either $\breve{x}_j = 0$ or $\breve{x}_j = \hat{x}_j$.

**Corollary 4** For $A \bullet X = B$ and $A \bullet X \ge B$:

i. It is solvable in polynomial time whether the system is consistent or not.
ii. If the system is consistent, the maximum solution, the minimal solutions and the maximal interval solutions are computable.
iii. For inconsistent system we can determine the equations (inequalities, respectively) that can not be satisfied by $X_{\text{gr}}$.

*F. Selected elements*

Theorem 4, Theorem 5 and their Corollaries prove that all coefficients with $a_{ij} \wedge \hat{x}_j < b_i$ do not contribute for solving the FLS. We propose a selection of all coefficients that contribute to solve the system. All other coefficients are non-essential for solvability procedure and we drop them.

**Definition 4** Let the FLS with associated matrix $A^*$ be given. All non-zero elements in $A^*$ are called *selected.*

From Theorem 4 and Theorem 5 we obtain

**Corollary 5** Let the FLE be given.

i. It is consistent if and only if for each $i$ ,$1 \le i \le m$ , there exists at least one selected coefficient $a^*_{ij}$, otherwise it is inconsistent.

ii. If the system is consistent then

$$X_{\mathrm{gr}} = A^t \alpha B, \tag{6a}$$

$$X_{\mathrm{gr}} = (1,...,1)^t \tag{6b}$$

is its unique maximal (i.e. greatest, or maximum) solution for (2) and (3) respectively.

iii. The time complexity function for establishing the consistency of the system and for computing $X_{\mathrm{gr}}$ is $O(mn)$.

*G. Finding minimal solutions*

Nonzero elements in associated matrix indicate non-redundant elements for solving FLS. First we remove all zero rows (redundant equations) and all zero columns (non-essential coefficients) from $A^*$.

We expand the possible irredundant paths (called coverings in [12], i.e. different ways to satisfy simultaneously equations of the system) using the matrix $A^*$ and the algebraic properties of the logical sums, see [11], [12] for details.

For any consistent FLSE the minimal solutions are computable and the set of all its minimal solutions is finite.

## IV. Algorithm For Finding Optimal Solution

We first decompose the linear objective function $Z$ in two functions $Z'$ and $Z''$ by separating the nonnegative and negative coefficients (as it is proposed in [8] for instance). Using the extremal solutions for constraint and $Z'$, $Z''$, we solve the optimization problem.

The linear objective function (1) determines a cost vector $Z = (c_1, c_2, ..., c_n)$. We decompose $Z$ into two vectors with suitable components $Z' = (c'_1, c'_2, ..., c'_n)$ and $Z'' = (c''_1, c''_2, ..., c''_n)$,

where

$$c'_j = \begin{cases} c_j, & if \quad c_j \ge 0 \\ 0, & if \quad c_j < 0 \end{cases} \quad (7), \quad c''_j = \begin{cases} 0, & if \quad c_j \ge 0 \\ c_j, & if \quad c_j < 0 \end{cases} \quad (8).$$

Hence the objective value is $Z = Z' + Z''$ and cost vector components are $c_j = c'_j + c''_j$, for each $j = 1,...,n$.

The components of $Z'$ are non-negative, the components of $Z''$ are non-positive.

In this section we present algorithm that covers several optimization problems:

1. Minimize the linear objective function (1), subject to constraint (2) or (3).
2. Maximize the linear objective function (1), subject to constraint (2) or (3).

*H. Minimize the linear objective function, subject to constraint $A \bullet X = B$ or $A \bullet X \ge B$*

The original problem: to minimize (1) subject of constraint (2) or (3) (with max-min composition) splits into two problems, namely to minimize both

$$Z' = \sum_{j=1}^{n} c'_j x_j \tag{9}$$

and

$$Z'' = \sum_{j=1}^{n} c''_j x_j \tag{10}$$

with constraint (2) or (3), i.e. for the problem (1) $Z$ takes its minimum when both $Z'$ and $Z''$ take minimum.

Since the components $c'_j, 1 \le j \le n$, in $Z'$ are nonnegative, $Z'$ takes its minimum among the minimal solutions of FLS. Hence for the problem (9) the optimal solution $\breve{X} = (\breve{x}_1, ..., \breve{x}_n)$ is among the minimal solutions of the FLS.

Since the components $c''_j, 1 \le j \le n$, in $Z''$ are no positive, $Z''$ takes its minimum for the greatest solution of FLS. Hence for the problem (10) the optimal solution is $\hat{X} = (\hat{x}_1, ..., \hat{x}_n) = X_{gr}$.

The optimal solution of the problem (1) with FLS constraint is $X^* = (x^*_1, ..., x^*_n)$, where

$$x^*_i = \begin{cases} \hat{x}_i, & if \quad c_i < 0 \\ \breve{x}_i, & if \quad c_i \ge 0 \end{cases} \tag{11}$$

and the optimal value is

$$Z^* = \sum_{j=1}^{n} c_j x^*_j = \sum_{j=1}^{n} c''_j \hat{x}_j + \sum_{j=1}^{n} c'_j \breve{x}_j. \tag{12}$$

*I. Maximize the linear objective function, subject to constraint $A \bullet X = B$ or $A \bullet X \ge B$*

If the aim is to maximize the linear objective function (1), we again split it, but now for $Z''$ the optimal solution is among the minimal solutions of the system (2) or (3), for $Z'$ the optimal solution is $X_{gr}$. In this case the optimal solution of the problem is $X^* = (x^*_1, ..., x^*_n)$, where

$$x^*_j = \begin{cases} \breve{x}_j, & if \quad c_j < 0 \\ \hat{x}_j, & if \quad c_j \ge 0 \end{cases}. \tag{13}$$

and the optimal value is

$$Z^* = \sum_{j=1}^{n} c_j x_j^* = \sum_{j=1}^{n} c_j' \hat{x}_j + \sum_{j=1}^{n} c_j'' \check{x}_j. \qquad (14)$$

*J. Algorithm for finding optimal solutions*

We propose algorithm for finding optimal solutions if the linear objective function is (1) and the constraint is (2) or (3).

In any of these cases the optimal value is

$$Z^* = \sum_{j=1}^{n} c_j x_j^*. \qquad (15)$$

***Algorithm for finding optimal solutions***

STEP 1. Enter data for the matrices $A_{m\times n}$, $B_{m\times 1}$ and the cost vector $C_{1\times n}$.

STEP 2. Establish consistency of the FLS. If the FLS is inconsistent go to step STEP 8.

STEP 3. Compute $X_{\text{gr}}$ and all minimal solutions of FLS.

STEP 4. If finding $Z_{\max}$ go to Step STEP 6.

STEP 5. For finding $Z_{\min}$ compute $x_j^*$, $j = 1,...,n$ according to (8). Go to Step STEP 7.

STEP 6. For finding $Z_{\max}$ compute $x_j^*$, $j = 1,...,n$ according to (10).

STEP 7. Compute the optimal value according to (15).

STEP 8. End.

## V. SOFTWARE

Software is developed, based on the described algorithm. It can be obtained free by contact with any of the authors. This software is divided into two independent programs, one written in Java and one written in MATLAB. Both have their advantages. Java program has its own GIU, which make it stand alone application for optimization. In addition it is faster than the MATLAB program. It is open source and can be extended or just used in other Java applications. MATLAB program on the other hand does not have GIU, but it can be run easily from the command line interpreter (CLI) of MATLAB. In this way the program can be used separately as well as in combination with any of the MATLAB packages for rapid producing of feature rich programs.

In the next section some examples are given. All of them are tested in both Java and MATLAB applications. The results are given like a snapshot of a MATLAB session but, of course, they are identical in the Java program.

In the described programs we first solve the FLS of the constraints. Then we implement the algorithm for minimizing (maximizing, respectively) the objective function $Z$, as explained in Section IV.

## VI. EXPERIMENTAL RESULTS

Since the solution set of (2) or (3) can be non-convex, traditional linear programming methods cannot be applied. In order to find the optimal solution for (1), the problem is converted into a $0-1$ integer programming problem. Fang and Li in [4] solved this $0-1$ integer programming problem by branch and bound method with jump-tracking technique. Improvements of this method are proposed in [16] by providing fixed initial upper bound for the branch and bound part (by Wu et al.) and with update of this initial upper bound in [15]. Conventional investigations [4], [6], [8], [14], [15], [16] begin with various attempts to find some feasible solutions, then to find an arbitrary solution of (2) or (3) and proceed with it in the solution set for search of minimal solutions. The procedures are clumsy; the main obstacle is how to solve (2) or (3).

It is well-known [7], [12] that the solution set of any consistent system (2) or (3) is completely determined by the unique maximum solution and a finite number of minimal solutions. One possible way to find optimal solution is to compute all extremal solutions and then to find the optimal solution by suitable decomposition of (1), as we propose here.

**Example 1:** Minimize

$Z = 4x_1 - x_2 + 6x_3 + 4x_4 - 3x_5$, subject to

$A \bullet X = B$, $0 \le x_j \le 1, 1 \le i, j \le 5$, if

$$A = \begin{pmatrix} 0.40 & 0.50 & 0.45 & 0.50 & 0.50 \\ 0.70 & 0.60 & 0.70 & 0.70 & 0.20 \\ 0.60 & 0.30 & 0.80 & 0.80 & 0.80 \\ 0.90 & 0.95 & 0.60 & 0.80 & 0.80 \\ 1.00 & 0.70 & 1.00 & 1.00 & 1.00 \end{pmatrix},$$

$$B' = \begin{pmatrix} 0.50 & 0.70 & 0.80 & 0.90 & 1.00 \end{pmatrix}.$$

**In MATLAB:**

```
>> sol = fuzzy_minimize(A,B,Z,'max-min')
 sol = rows: 5
 cols: 5
 gr: [1 0.9000 1 1 1]
 gr_solved: [5x1 double]
 selected: [1x5 struct]
 low: [5x15 double]
 final_x: [5x2 double]
 final_z: -1.1000
 z: [1x15 double]
```

The system of constraints has one greatest and 15 minimal solutions. They are listed below:

```
>> sol.gr
ans =
 1.0000    0.9000    1.0000    1.0000    1.0000

>> sol.low
ans =
 Columns 1 through 8
1.0000 1.0000 1.0000 1.0000 1.0000 0.9000 0.9000 0.9000
0.5000      0      0      0      0 0.5000      0      0
0.8000 0.8000 0.8000      0      0 1.0000 1.0000 1.0000
     0 0.5000      0 0.8000      0      0 0.5000      0
     0      0 0.5000      0 0.8000      0      0 0.5000

 Columns 9 through 15
     0 0.9000      0 0.9000 0.7000      0      0
0.9000      0 0.9000      0 0.9000 0.9000 0.9000
1.0000      0      0      0      0 0.7000      0
     0 1.0000 1.0000      0      0      0 0.7000
     0      0      0 1.0000 1.0000 1.0000 1.0000
```

Since the second and the fifth coefficients of the objective function $Z$ are negative we have:

$Z' = -x_2 - 3x_5$, $Z'' = 4x_1 + 6x_3 + 4x_4$.

We take the second and the fifth components in the optimal solution from the greatest solution. Hence $Z' = -0.9 - 3 = -3.9$.

The other components must be chosen from one of the minimal solutions. For instance, $Z''(1) = 4.1 + 6.0,8 + 0 = 8.8$ and in this manner $Z(1) = Z' + Z''(1) = 4.9$

The algorithm computes value of objective function for all minimal solutions:

*>> sol.z*
*ans =*
*Columns 1 through 8*
*4.9000 6.9000 4.9000 3.3000 0.1000 5.7000 7.7000 5.7000*

*Columns 9 through 15*
*2.1000 3.7000 0.1000 -0.3000 -1.1000 0.3000 -1.1000*

We have to take the minimal solution for which the objective function gains its minimum.

Minimum of the objective function is gained for 13th and 15th minimal solution, hence

*>> sol.final_z*
*ans =*
*-1.1000*

*>> sol.final_x*
*ans =*
*0.7000 0*
*0.9000 0.9000*
*0 0*
*0 0.7000*
*1.0000 1.0000*

The execution time for this example tested on computer with Sempron 2800+ processor on 1.61GHz, with 512MB RAM and MATLAB R2006b is 1.11 seconds.

**Example 2**. Minimize

$$Z = 4x_1 - x_2 + 6x_3 + 4x_4 - 3x_5 + 8x_7 - 7x_8 - 3x_9$$

subject to $A \bullet X \geq B, 0 \leq x_j \leq 1, 1 \leq i, j \leq 9$, if

$$A = \begin{pmatrix} 0.25 & 0.4 & 0.4 & 0.6 & 0.1 & 0.8 & 0.8 & 0.9 & 1.0 \\ 0.12 & 0.1 & 0.5 & 0.5 & 0.6 & 0.8 & 0.95 & 0.8 & 0.8 \\ 0.2 & 0.15 & 0.2 & 0.6 & 0.7 & 0.5 & 0.1 & 0.4 & 0.4 \\ 0.15 & 0.2 & 0.5 & 0.6 & 0.5 & 0.8 & 0.3 & 0.9 & 1.0 \\ 0.1 & 0.14 & 0.5 & 0.6 & 0.6 & 0.2 & 0.9 & 0.4 & 1.0 \\ 0.2 & 0.4 & 0.4 & 0.2 & 0.7 & 0.8 & 0.9 & 0.98 & 0.5 \\ 0.15 & 0.25 & 0.5 & 0.6 & 0.2 & 0.8 & 0.8 & 0.75 & 1.0 \\ 0.2 & 0.4 & 0.45 & 0.2 & 0.7 & 0.85 & 0.5 & 0.9 & 0.6 \\ 0.15 & 0.35 & 0.5 & 0.6 & 0.65 & 0.7 & 0.9 & 0.95 & 0.9 \end{pmatrix}$$

$$B^t = \begin{pmatrix} 0.2 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 & 0.8 & 0.95 \end{pmatrix}$$

When we minimize the objective function we have to minimize both $Z' = -x_2 - 3x_5 - 7x_8 - 3x_9$ and $Z'' = 4x_1 + 6x_3 + 4x_4 + 8x_7$. Here we have three minimal solutions for the FLSI constraints.

**In MATLAB:**

*>> sol.gr*
*ans =*
*1 1 1 1 1 1 1 1 1*

*>> sol.low*
*ans =*
*0 0 0*
*0 0 0*
*0 0 0*
*0.5000 0 0*
*0 0.5000 0*
*0 0 0.5000*
*0 0 0*
*0.9500 0.9500 0.9500*
*0.9000 0.9000 0.9000*

The minimum is achieved for the second and third minimal solutions. Value of the objective function is:

*>> sol.final_z*
*ans = -14*

The execution time for this example tested on computer with Intel Core 2 Duo processor on 1.86GHz, with 2GB RAM and Java 6.0 is 0.016 seconds.

## REFERENCES


[1] B. De Baets, "Analytical solution methods for fuzzy relational equations" in *The Handbooks of Fuzzy Sets Series*, Fundamentals of Fuzzy Sets Vol. 1, Kluwer Academic Publishers, 2000, pp. 291-340.

[2] A. Di Nola and A. Lettieri, "Relation equations in residuated lattices" in *Rendiconti del Circolo Matematico di Palermo*, vol. II, XXXVIII, 1989, pp 246-256.

[3] A. Di Nola, W. Pedrycz, S. Sessa and E. Sanchez, "Fuzzy relation equations and their application to knowledge engineering", *Kluwer Academic Press*, Dordrecht, 1989.

[4] S.-G. Fang and G. Li, "Solving fuzzy relation equations with linear objective function", *Fuzzy Sets and Systems*, vol. 103, pp. 107-113, 1999.

[5] M. Garey and D. Johnson, "Computers and intractability a guide to the theory of NP-completeness", *Freeman*, San Francisco, CA, 1979.

[6] S. M. Guu and Y-K. Wu, "Minimizing a linear objective function with fuzzy relation equation constraints", *Fuzzy Optimization and Decision Making*, vol. 4 (1), 2002, pp. 347-360.

[7] G. Klir, U. Clair and B. Yuan, "Fuzzy set theory foundations and applications", *Prentice Hall PRT*, 1997.

[8] J. Loetamonphong and S.-C. Fang, "Optimization of fuzzy relation equations with max-product composition", *Fuzzy Sets and Systems*, vol. 118 (3), 2001, pp. 509-517.

[9] S. MacLane and G. Birkhoff, "Algebra", *Macmillan*, New York, 1979.

[10] A. Markovskii, "On the relation between equations with max-product composition and the covering problem", *Fuzzy Sets and Systems*, vol. 153, 2005, pp. 261-273.

[11] K. Peeva, "Universal algorithm for solving fuzzy relational equations", *Italian Journal of Pure and Applied Mathematics*, vol. 19, 2006, pp. 9-20.

[12] K. Peeva and Y. Kyosev, "Fuzzy relational calculus - theory, applications and software (with CD-ROM)", *Advances in Fuzzy Systems - Applications and Theory*, Vol. 22, World Scientific Publishing Company, 2004, Software downloadable from http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=6214

[13] E. Sanchez, "Resolution of composite fuzzy relation equations", *Information and Control*, vol. 30, 1976, pp. 38-48.

[14] P. Z. Wang, D. Z. Zhang, E. Sanchez and E. S. Lee, "Latticized linear programming and fuzzy relation inequalities", *J. Math. Anal. Appl.*, vol. 159 (1), 1991, pp. 72-87.

[15] Y-K. Wu and S. M. Guu "Minimizing a linear function under a fuzzy max–min relational equation constraint", *Fuzzy Sets and Systems*, vol 150, 2005, pp 147-162.

[16] Y-K. Wu, S. M. Guu and Y-C. Liu, "An accelerated Approach for solving fuzzy relation equations with a linear objective function", *IEEE Transactions on Fuzzy Systems*, vol. 10(4), 2002, pp. 552-558.