# Max-product Fuzzy Linear Systems - Application to Linear Optimization

K. Peeva, Zl. Zahariev
Technical University of Sofia, Bulgaria
e-mail: kgp@tu-sofia.bg,

**Abstract**

An optimization problem is studied with linear objective function subject to a system of fuzzy linear equations using max −product composition. Since the solution set of the system is non-convex, conventional linear programming methods cannot be applied. We apply methods for solving fuzzy linear system of equations when the composition is max-product, as proposed in [23]. These methods provide algorithm for computing the maximum solution and the set of all minimal solutions, when the system is consistent. After computing all extremal solutions of the system, we propose a method to solve the linear optimization problem.

**Keywords:** linear optimization; max −product fuzzy linear equations; inverse problem;

## 1 Introduction

We are interested in the optimization problem with a linear objective function subject to a system of fuzzy linear system of equations with max −product composition. The results for solving this optimization problem are provided by the inverse problem resolution for fuzzy linear systems as presented in [23]. Solving fuzzy linear system of equations with max −product composition is subject of great scientific interest. The main results are published in [4], [6], [21], [22]. They demonstrate long and difficult period of investigations for discovering analytical methods and procedures to determine complete solution set, as well as to develop software for computing the maximum and all minimal solutions [21], [22].

The first and most essential are Sanchez results [24] for the greatest solution of fuzzy relational equations with $\max - \min$ and $\min - \max$ composition. Sanchez gives formulas that permit to determine the potential greatest solution in any of these cases, often used as solvability criteria. After Sanchez results for the greatest solution, attention was paid on the minimal solutions [2], [10], [17]–[22]. Universal algorithm and software for solving $\max - \min$ and $\min - \max$ fuzzy relational equations is proposed in [20], [21], [22].

For fuzzy linear system of equations with max− product composition [5], [6], the results concern finding greatest solution [3] and minimal solutions, estimating time complexity of the problem, applications in optimization problems [9], [12], [13], [14].

Implementing methods, procedures and software for inverse problem resolution of fuzzy relational equations with max −product composition as presented in [23]

This paper deals with optimization problem - minimize the linear objective function

$$Z = \sum_{i=1}^{n} c_i x_i \tag{1}$$

subject to constrains - a fuzzy linear system of equations with max − product composition

$$A \odot X = B, \; 0 \le x_i \le 1, \; 1 \le i \le n \tag{2}$$

where $A$ stands for the matrix of coefficients, $X = (x_i)_{n\times 1}$ stands for the matrix of unknowns, $B$ is the right-hand side of the system, the max − product composition is written as $\odot$ and $c = (c_1, ..., c_m)$ is the weight (cost) vector. The aim is to minimize $Z$ subject to constrains (2).

When the solution set of (2) is not empty, it is completely determined by the unique maximum solution and a finite number of minimal solutions. Since the solution set can be non-convex, traditional linear programming methods cannot be applied to this problem. In this paper we apply the algorithm and software from [23] for computing the maximum solution and all minimal solutions of the consistent system $A \odot X = B$ and solve the linear optimization problem.

The paper is organized as follows. In Section 2 we introduce basic notions. In Section 3 we give the main results for solving fuzzy linear system of equations - determination of the greatest solution and all minimal solutions. Section 4 presents the effect of the weight vector and shows that the linear optimization problem (1) with constraints (2) can be devided into two parts: one with non-negative weight coefficients and the other with negative weight coefficients. The algorithm for solving the linear optimization problem (1) with constraints (2) is presented in Section 5, where we also propose software description and some comments on experimental results.

Terminology for computational complexity and algorithms is as in [1], [7], for fuzzy sets and fuzzy relations is according to [4], [6], [11], [22], [24], for lattices - as in [8], for algebra - as in [15].

# 2 Basic Notions

Partial order relation on a partially ordered set (poset) $P$ is denoted by the symbol $\le$. By a *greatest element* of a poset $P$ 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.

Set $\mathbb{I}_\odot = \langle [\,0,\,1\,], \vee, \wedge, 0, 1, \odot \rangle$, where $[\,0,\,1\,]$ is the real unit interval, $\odot$ is the usual product between real numbers and $\vee$, $\wedge$ are respectively defined by

$$a \vee b = max\{a, b\}, \;\; a \wedge b = min\{a, b\}.$$

Then $\mathbb{I}_\odot$ is a complete lattice with universal bounds 0 and 1; it is residuated with respect to $\odot$, being the residuum given by:

$$a \diamond b = \begin{cases} 1, & \text{if } a \le b \\ \frac{b}{a}, & \text{if } a > b \end{cases} .$$

The algebraic structure $\mathbb{I}_{\odot} = \langle [\,0,\, 1\,],\, \vee,\, \wedge,\, 0,\, 1,\, \odot \rangle$ is called *fuzzy algebra.*

We denote by $F(X)$ the fuzzy sets over the crisp set $X$. A *fuzzy relation* $R \in F(X \times Y)$ is defined as a fuzzy subset of the Cartesian product $X \times Y$,

$$R = \{\, (\,(x, y),\;\; \mu_R(x, y))\;\; |\, (x, y) \in X \times Y,\;\; \mu_R :\; X \times Y \;\to\; [\,0,\, 1\,]\,\}\,.$$

The *inverse* (or *transpose*) $R^{-1} = R^t \in F(Y \times X)$ of $R \in F(X \times Y)$ is defined as

$$R^{-1}(y, x) = R(x, y) \quad \text{for all pairs} \quad (y, x) \in Y \times X.$$

A matrix $A = (a_{ij})_{m\times n}$, with $a_{ij} \in [\,0,\, 1\,]$ for each $i,\, j, 1 \leq i \leq m,\, 1 \leq j \leq n$, is called a *membership matrix* [11].

In what follows we write 'matrix' instead of 'membership matrix'.

We consider operations with matrices on the fuzzy algebra $\mathbb{I}_{\odot}$.

Any fuzzy relation $R \in F(X \times Y)$ is representable by a matrix [6], written for convenience with the same letter $R = (r_{ij})$, where $r_{ij} = \mu_R(x_i, y_j)$ for any $(x_i, y_j) \in X \times Y$.

We stipulate to use the matrix $R = (r_{ij})$ for the fuzzy relation $R \in F(X\times Y)$.

**Definition 2** Let the matrices $A = (a_{ij})_{m\times p}$ and $B = (b_{ij})_{p\times n}$ be given.

i) The matrix $C = (c_{ij})_{m\times n} = A \odot B$ is called $\odot-$ *product* of $A$ and $B$ if $c_{ij} = \max\limits_{k=1}^{p}(a_{ik}.b_{kj})$, when $1 \leq i \leq m,\, 1 \leq j \leq n$.

ii) The matrix $C = (c_{ij})_{m\times n} = A \diamond B$ is called $\diamond$*-product* of $A$ and $B$ if $c_{ij} = \min\limits_{k=1}^{p}(a_{ik} \diamond b_{kj})$, when $1 \leq i \leq m,\, 1 \leq j \leq n$.

Definition 2 permits to manipulate with the matrix products instead of with compositions of fuzzy relations.

**Theorem 2** [5] Let $A = (a_{ij})_{m\times p}$ and $C = (c_{ij})_{m\times n}$ be given matrices and let $\mathbb{B}_{\odot}$ be the set of all matrices such that $A \odot B = C$. Then:

i) $\mathbb{B}_{\odot} \neq \emptyset$ iff $A^t \diamond C \in \mathbb{B}_{\odot}$.

ii) If $\mathbb{B}_{\odot} \neq \emptyset$ then $A^t \diamond C$ is the greatest element in $\mathbb{B}_{\odot}$.

# 3 $\odot-$Fuzzy Linear Systems of Equations

We study fuzzy linear systems of equations with $\odot-$composition ($\odot-$FLSE):

$$\left|\begin{array}{ccccc} (a_{11}.\, x_1) & \vee \cdots \vee & (a_{1\,n}.\, x_n) & = & b_1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ (a_{m\,1}.\, x_1) & \vee \cdots \vee & (a_{m\,n}.\, x_n) & = & b_m \end{array}\right. , \tag{3}$$

written in the following equivalent matrix form

$$A \odot X = B,$$

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. For each $i, 1 \leq i \leq m$ and for each $j, 1 \leq j \leq n$, we have $a_{ij}$, $b_i$, $x_j \in [\,0,\,1\,]$ and the max −product composition is written as $\odot$.

For $X = (x_j)_{n\times1}$ and $Y = (y_j)_{n\times1}$ the inequality $X \leq Y$ means $x_j \leq y_j$ for each $j$, $1 \leq j \leq n$.

Let us first define solutions for $A \odot X = B$ and give a classification of the $\odot-$FLSE according to the number of its solutions.

**Definition 3** Let the system $A \odot X = B$ in $n$ unknowns be given.

i) $X^0 = (x^0_j)_{n\times1}$ with $x^0_j \in [0, 1]$, when $1 \leq j \leq n$, is called a *(point) solution* of the system $A \odot X = B$ if $A \odot X^0 = B$ holds.

ii) The set of all point solutions $\mathbb{X}^0$ of $A \odot X = B$ is called *complete solution set.*

iii) If $\mathbb{X}^0 \neq \emptyset$ then $A \odot X = B$ is called *consistent*, otherwise $A \odot X = B$ is called *inconsistent.*

In the next exposition we omit the word "point" in "point solution".

**Definition 4** Let the system $A \odot X = B$ in $n$ unknowns be given.

i) A solution $X^0_{\mathrm{low}} \in \mathbb{X}^0$ is called a *lower (minimal) solution* of $A \odot X = B$ if for any $X^0 \in \mathbb{X}^0$ the relation $X^0 \leq X^0_{\mathrm{low}}$ implies $X^0 = X^0_{\mathrm{low}}$, where $\leq$ denotes the partial order, induced in $\mathbb{X}^0$ by the order of [0, 1]. Dually, a solution $X^0_{\mathrm{u}} \in \mathbb{X}^0$ is called an *upper (maximal) solution* of $A \odot X = B$ if for any $X^0 \in \mathbb{X}$ the relation $X^0_{\mathrm{u}} \leq X^0$ implies $X^0 = X^0_{\mathrm{u}}$. When the upper solution is unique, it is called *greatest (or maximum) solution.*

ii) The $n-$tuple $(X_1, \ldots, X_n)$ with $X_j \subseteq [0,\,1]$ for each $j$, $1 \leq j \leq n$, is called an *interval solution* of the system $A \odot X = B$ if any $X^0 = (x^0_j)_{n \times 1}$ with $x^0_j \in X_j$ for each $j$, $1 \leq i \leq n$, implies $X^0 = (x^0_j)_{n \times 1} \in \mathbb{X}^0$.

iii) Any interval solution of $A \odot X = B$ whose components (interval bounds) are determined by a lower solution from the left and by the maximum solution from the right, is called *maximal interval solution* of $A \odot X = B$.

We consider inhomogeneous systems with $b_i \neq 0$ for each $i = 1, ..., m$.

If $A \odot X = B$ is consistent, according to Theorem 2, it has unique maximum solution $X_{\mathrm{gr}} = A^t \diamond B$. 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 [23]. Since there exists analytical expression for the maximum solution, attention in references is paid on computing minimal solutions.

## 3.1 Preliminary simplifications

We propose the first steps for simplifying $\odot-$FLSE so that the complete solution set can be easily found and the size of the instant can be reduced.

### 3.1.1 Step 1. Associated matrix

For the system (3) any coefficient $a_{ij} \geq b_i$ provides a way to satisfy the $i-$th equation with $a_{ij} \,.\, x_j = b_i$, when $x_j = \frac{b_i}{a_{ij}}$. This leads to the idea to distinguish coefficients that contribute for solving the system from these that do not contribute, see (4).

We assign to $A \odot X = B$ a symbolic matrix $A^* = (a^*_{ij})$ with elements $a^*_{ij}$ determined according to the next expression:

$$a^*_{ij} = \begin{cases} S, & \text{if } a_{ij} < b_i \\ E, & \text{if } a_{ij} = b_i \\ G & \text{if } a_{ij} > b_i \end{cases} . \tag{4}$$

The matrix $A^*$ with elements $a^*_{ij}$, determined by (4), is called *associated matrix* of the system (3). Its elements depend both on $A$ and on $B$.

The time complexity function for obtaining $A^*$ is O(mn).

**Interpretation of $A^*$**

- Any $a^*_{ij} = S$ in $A^*$ corresponds to $a_{ij} < b_i$ in the $i-$th equation of (3). But $a_{ij} < b_i$ means $a_{ij} \,.\, x_j < b_i$ for each $x_j \in [0,\, 1]$. Hence each $a^*_{ij} = S$ in the $i-$th row of $A^*$ indicates, that the coefficient $a_{ij}$ do not contribute to satisfy $i-$th equation of (3).
- Any $a^*_{ij} \neq S$ in (4) corresponds to $a_{ij} \geq b_i \neq 0$ in the $i-$th equation of (3) that determines a way to satisfy this equation by $x_j = \frac{b_i}{a_{ij}}$. In this case $a_{ij}.x_j = a_{ij}.\frac{b_i}{a_{ij}} = b_i$.

Hence, associated matrix $A^*$ provides first simplification. Rather than work with the system $A \odot X = B$, we use $A^*$, whose elements capture all the properties of the equations. This reduces the size of the instant and makes easier to solve the original system.

### 3.1.2 Step 2. IND vector

We introduce a vector $IND = IND_{m \times 1}$ to establish consistency of the system. We describe how the components of $IND$ depend on $A^*$. Let we denote by $|G_i|$ the number of elements $a^*_{ij} = G$ and by $|E_i|$ the number of elements $a^*_{ij} = E$ in the $i-$th row of $A^*$, $j = 1, ..., n$. Then

$$IND(i) = |G_i| + |E_i| \tag{5}$$

equals the number of elements $a^*_{ij} \neq S$ in the $i-$th row of $A^*$. It means that:

i) If $a^*_{ij} = S$ for each $j = 1, ..., n$ then $IND(i) = 0$. In this case the $i-$th equation can not be satisfied and the system is inconsistent.

ii) If $a^*_{ij} \neq S$ for some $j = 1, ..., n$ then $IND(i) = |G_i| + |E_i| \neq 0$. In this case the $i-$th equation can be satisfied by $|G_i| + |E_i|$ different paths. If $IND(i) \neq 0$ for each $i = 1, ..., m$ then the system can be either consistent or inconsistent.

**Lemma 1** Let the system $A \odot X = B$ be given. Then we have:

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

ii) If the system is consistent then the number of its potential minimal solutions does not exceed
$$PN = \prod_{i=1}^{m} IND(i). \tag{6}$$
Here $IND(i)$ is computed according to (5).

#### 3.1.3 Step 3. Rearrangement of the equations

Two systems are called *equivalent* [15] if any solution of the first one is a solution of the second one and vice versa. Any interchange of equations in the system $A \odot X = B$ results an equivalent system.

A system $A \odot X = B$, in which the equations are rearranged in such a way that the components of the index vector $IND$ are ranked non-decreasingly, i.e.

$$IND(1) \leq IND(2) \leq \ldots \leq IND(m),$$

is said to be in a *normal form*.

# 4 Solving $\odot-$Fuzzy Linear Systems

In this section we follow [23], where a unified and exact method and algorithm for solving inhomogeneous $\odot-$FLSE of the form $A \odot X = B$ is proposed.

Let the following stipulations be satisfied for inhomogeneous $A \odot X = B$:

1. The system $A \odot X = B$ has coefficient matrix $A = (a_{ij})_{m\times n}$, matrix of unknowns $X = (x_j)_{n\times 1}$, and right-hand side $B = (b_i)_{m\times 1}$ with $b_i \neq 0$ for each $i = 1, ..., m$. Hence it has $n$ unknowns and $m$ equations.

2. The associated matrix $A^*$ for the system $A \odot X = B$ is obtained.

3. Any coefficient $a^*_{ij} = S$ is called $S-$type coefficient, any $a^*_{ij} = E$ is called $E-$type coefficient and any $a^*_{ij} = G$ is called $G-$type coefficient.

4. For each $j$, $j = 1, \cdots, n$, $A^*(j) = (a^*_{ij})_{m\times 1}$ denotes the $j-$th column of $A^*$ and $a^*_{ij}$ denotes the $i-$th element $(1 \leq i \leq m)$ in $A^*(j)$.

**Theorem 3** [23] Let the system $A \odot X = B$ be given.

i) If $A^*(j)$ contains $G-$type coefficient(s) $a^*_{ij} = G$ and
$$\hat{x}_j = \min_{i=1}^{m} \{\frac{b_i}{a_{ij}}\}, \text{ when } a_{ij} > b_i,$$
then $x_j = \hat{x}_j$ implies in (3):

- $a_{ij} \,.\, x_j = b_i$ for each $i$, $1 \leq i \leq m$ when $\frac{b_i}{a_{ij}} = \hat{x}_j$,
- $a_{ij} \,.\, x_j < b_i$ for each $i$, $1 \leq i \leq m$ with $\frac{b_i}{a_{ij}} \neq \hat{x}_j$.

ii) If $A^*(j)$ does not contain any $G-$type coefficient, but it contains $E-$type coefficient(s) $a^*_{kj} = E$, then $\hat{x}_j = 1$ and $x_j = \hat{x}_j = 1$ implies:

- $a_{ij} \,.\, x_j = b_i$ for each $i$, $1 \leq i \leq m$ with $a^*_{ij} = E$,
- $a_{ij} \,.\, x_j < b_i$ for each $i$, $1 \leq i \leq m$ with $a^*_{ij} = S$.

iii) If $A^*(j)$ contains neither $G-$ nor $E-$type coefficient then $\hat{x}_j = 1$ and $x_j = \hat{x}_j = 1$ implies $a_{ij} \,.\, x_j < b_i$ for each $i$, $1 \leq i \leq m$ ($a^*_{ij} = S$ in $A^*(j)$).

The proof follows from the definition of the associated matrix, its relationship with the system (3) and expression (4).

**Corollary 1** [23] For any consistent system $A \odot X = B$,

$$X_{\text{gr}} = A^t \diamond B = \hat{X} = (\hat{x}_j)_{n\times 1}$$

where $\hat{x}_j$, $1 \leq j \leq n$, are computed according to Theorem 3.

**Corollary 2** [23] If $a^*_{ij} = S$ for each $i = 1, ..., m$, then $\check{x}_j = 0$ in any minimal solution $\check{X} = (\check{x}_j)_{n\times 1}$ of the consistent system $A \odot X = B$.

**Corollary 3** [23] If $\check{X} = (\check{x}_j)_{n\times 1}$ is a minimal solution of the consistent system $A \odot X = B$, then for each $j = 1, ..., n$ either $\check{x}_j = 0$ or $\check{x}_j = \hat{x}_j$ .

## 4.1 Selected elements

Theorem 3 and its Corollaries 2, 3 prove that all $S-$ type coefficients do not contribute for solving the system and there may exist redundant coefficients of type $G$ and $E$ in the system. We propose a selection of all coefficients that contribute to solve the system. All other coefficients are called non-essential for solvability procedure and we drop them.

**Definition 4** Let the system $A \odot X = B$ with associated matrix $A^*$ be given.

i) If $A^*(j) = (a^*_{ij})_{m\times 1}$ contains $G-$type coefficient $a^*_{kj} = G$, such that

$$\frac{b_k}{a_{kj}} = \min_{i=1,}^{m}\{\frac{b_i}{a_{ij}}\} \text{ when } a_{ij} > b_i,$$

then each $G-$type coefficient $a^*_{ij}$ in $A^*(j)$ with $\frac{b_i}{a_{ij}} = \frac{b_k}{a_{kj}}$ is called *selected*.

ii) If $A^*(j) = (a^*_{ij})_{m\times 1}$ does not contain $G-$type coefficient, but it contains $E-$type coefficient(s), then all $E-$type coefficients in $A^*(j)$, namely $a^*_{ij} = E$ when $1 \leq i \leq m$, are called *selected*.

iii) If $A^*(j)$ does not contain neither $G-$, nor $E-$type coefficient, then there does not exist selected coefficient in $A^*(j)$.

From Theorem 3 we obtain

**Corollary 4** [23] Let the system $A \odot X = B$ be given.

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

ii) If the system is consistent then

$$X_{\mathrm{gr}} = A^t \diamond B \tag{7}$$

is its unique maximal (i.e. greatest, or maximum) solution.

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

## 4.2 Help matrix and dominance matrix

Now we propose the next simplification steps.

### 4.2.1 Step 4. Help matrix

We introduce a *help matrix* $H = (h_{ij})_{m\times n}$ with elements

$$h_{ij} = \begin{cases} 1 & \text{if } a^*_{ij} \text{ is selected} \\ 0 & \text{otherwise} \end{cases} . \tag{8}$$

We upgrade the components of the vector $IND = IND_{m\times 1}$ to establish the consistency of the system and to diminish the potential number $PN$ (see (6)) of minimal solutions. Now the $i-$th component $IND(i)$ of $IND$ equals the number of selected coefficients in the $i-$th equation of the system, i.e.

$$IND(i) = \sum_{j=1}^{n} h_{ij}. \tag{9}$$

If there are no selected coefficients in the $i-$th equation, then $IND(i) = 0$ and the system is inconsistent, see Corollary 4 i).

Obviously, now the potential number $PN1$ of minimal solutions will be diminished in comparison with $PN$, i.e.

$$PN1 = \prod_{i=1}^{m} IND(i) \leq PN. \tag{10}$$

### 4.2.2 Step 5. Dominance matrix

In order to determine the minimal solutions of a $\odot-$FLSE, a suitable dominance relation for the rows of the help matrix $H$ is introduced.

**Definition 5** [23] Let $h_l = (h_{lj})$ and $h_k = (h_{kj})$ be the $l-$th and the $k-$th rows, respectively, in the help matrix $H$. If for each $j$, $1 \leq j \leq n$, $h_{lj} \leq h_{kj}$, then

- $h_l$ is said to be a *dominant row* to $h_k$ in $H$.
- $h_k$ is *redundant row* with respect to $h_l$ for solving the system (3).

If $h_k$ is redundant row with respect to $h_l$ for solving (3) it means that:

- $k-$th equation is automatically satisfied whenever $l-$th equation is satisfied.
- It is meaningless to investigate the $k-$th equation, because it will not lead to smaller solution than the $l-$th equation.
- When we eliminate $k-$th equation from next consideration we cut redundant branches from the search (they not lead to minimal solutions), making a more clever choice of the objects over which the search is performed.

Using Definition 5, we introduce a *dominance matrix* $D = (d_{ij})$ obtained from $H$ as described below. If the row $h_l$ dominates the row $h_k$ in $H$, then in $D$:

We preserve all elements of the row $h_l$ , i.e. $d_{ij} = h_{lj}$ for $j = 1, ..., n$. This preserves non-redundant (or essential for solution procedure) equation.

We replace all elements of the row $h_k$ by 0, i.e. $d_{kj} = 0$ for $j = 1, ..., n$. This eliminates redundant equations and also removes redundant branches of the search.

We again upgrade the components of the vector $IND$,

$$IND(i) = \sum_{j=1}^{n} d_{ij}.$$

Now the $i-$th component $IND(i)$ equals the number of non-redundant selected coefficients in the $i-$th equation of the system.

Next, the potential number $PN2$ of minimal solutions will be diminished in comparison with $PN1$ and $PN$, i.e.

$$PN2 = \prod_{i=1, IND(i) \neq 0}^{m} IND(i) \leq PN1 \leq PN. \tag{11}$$

## 4.3 Finding minimal solutions

From dominance matrix $D = (d_{ij})$ we go to the next simplification. We form a matrix $M = (m^*_{ij})$ indicating non-redundant elements for solving (3). First we remove all zero rows (redundant equations) and all zero columns (non-essential coefficients) from $D$. From the rest, we obtain:

$$m^*_{ij} = \begin{cases} \frac{b_i}{a_{ij}} & \text{if } h^*_{ij} = 1 \\ 0 & \text{if } h^*_{ij} = 0 \end{cases} \tag{12}$$

In what follows we work with the matrix $M$.

### 4.3.1 Algebraic properties

We expand the possible irredundant paths, i.e. different ways to satisfy simultaneously equations of the system using the matrix $M$ and the algebraic properties of the logical sums, as described below.

If the element $m^*_{ij} \neq 0$, we symbolize this with $\left\langle \frac{m^*_{ij}}{j} \right\rangle$. In this case $a_{ij} \,.\, m^*_{ij} = b_i$ and hence $\hat{x}_j = m^*_{ij}$ gives a lower bound to fulfill the $i-$th equation of the system; $\check{x}_j = \hat{x}_j = m^*_{ij}$ is the minimum value for the $j-$th component.

For each $i$, $1 \leq i \leq m$, the elements $m^*_{ij} \neq 0$ in $M$ mark the potential lower bounds of different ways, to satisfy the $i$-th equation of the system, written $M_i$ and symbolized by the sign $\sum$:

$$M_i = \sum_{1 \leq j \leq n} \left\langle \frac{m^*_{ij}}{j} \right\rangle. \tag{13}$$

We have to consider equations simultaneously, i.e., to compute the *concatenation* $W$ of all ways, symbolized by the sign $\prod$:

$$W = \prod_{1 \leq i \leq m} \left( \sum_{1 \leq j \leq n} \left\langle \frac{m^*_{ij}}{j} \right\rangle \right). \tag{14}$$

In order to compute complete solution set, it is important to determine different ways to satisfy simultaneously equations of the system. To achieve this aim we list the properties of concatenation (14).

Concatenation is *distributive* with respect to addition, i.e.

$$\left\langle \frac{m^*_{i_1 j_1}}{j_1} \right\rangle \left( \left\langle \frac{m^*_{i_2 j_2}}{j_2} \right\rangle + \left\langle \frac{m^*_{i_2 j_3}}{j_3} \right\rangle \right) =$$

$$= \left\langle \frac{m^*_{i_1 j_1}}{j_1} \right\rangle \left\langle \frac{m^*_{i_2 j_2}}{j_2} \right\rangle + \left\langle \frac{m^*_{i_1 j_1}}{j_1} \right\rangle \left\langle \frac{m^*_{i_2 j_3}}{j_3} \right\rangle. \tag{15}$$

This analytical expression demonstrates simultaneous satisfaction of both equations $(i_1, i_2)$ by selected elements in two different ways – the first way, that corresponds to the first summand, is by the selected elements $m^*_{i_1 j_1}$ and $m^*_{i_2 j_2}$ in rows $i_1, i_2$ and columns $j_1, j_2$, respectively; the second way corresponds to the second summand and it is formed by the selected elements $m^*_{i_1 j_1}, m^*_{i_2 j_3}$.

Concatenation is *commutative*:

$$\left\langle \frac{m^*_{i_1 j_1}}{j_1} \right\rangle \left\langle \frac{m^*_{i_2 j_2}}{j_2} \right\rangle = \left\langle \frac{m^*_{i_2 j_2}}{j_2} \right\rangle \left\langle \frac{m^*_{i_1 j_1}}{j_1} \right\rangle. \tag{16}$$

This provides the validity of Step 3 – rearrangement of equations in the $\odot-$FLSE.

The next property is called *absorption for multiplication*:

$$\left\langle \frac{m^*_{i_1 j_1}}{j_1} \right\rangle \left\langle \frac{m^*_{i_2 j_1}}{j_1} \right\rangle = \left\langle \frac{m^*_{i_1 j_1}}{j_1} \right\rangle \tag{17}$$

Expression (17) gives the lower solution for simultaneous satisfying of two different equations $i_1$ and $i_2$, when selected coefficients belong to the same column $j_1$. Hence, expanding along the non-zero elements in the the $i-$th row, we automatically satisfy all equations in the system, having the same $m^*_{ij}$. It is clear that this property reduces the number of the ways that have to be investigated.

We apply (15), (16), (17) to expand the parentheses in (14). We obtain the set of ways, from which we extract the minimal solutions:

$$W = \sum_{(j_1,\cdots,j_m)} \left\langle \frac{m^*_{i_1j_1}}{j_1} \right\rangle \left\langle \frac{m^*_{i_2j_2}}{j_2} \right\rangle \cdots \left\langle \frac{m^*_{i_mj_m}}{j_m} \right\rangle. \tag{18}$$

We simplify (18) according to the next described *absorption for addition* (missing $\left\langle \frac{m^*_{ij}}{j} \right\rangle$ are supposed to be $\left\langle \frac{0}{j} \right\rangle$):

$$\left\langle \frac{m^*_{i_1j_1}}{j_1} \right\rangle \cdots \left\langle \frac{m^*_{i_mj_m}}{j_m} \right\rangle + \left\langle \frac{m^*_{s_1j_1}}{j_1} \right\rangle \cdots \left\langle \frac{m^*_{s_mj_m}}{j_m} \right\rangle =$$

$$= \begin{cases} \left\langle \frac{m^*_{i_1j_1}}{j_1} \right\rangle \cdots \left\langle \frac{m^*_{i_mj_m}}{j_m} \right\rangle, & \text{if} \quad m^*_{i_tj_t} \le m^*_{s_tj_t} \quad \text{for} \quad t = 1, \cdots, m \\ \text{unchanged, otherwise} \end{cases} . \tag{19}$$

From two compatible point solutions with respect to the relation $\le$, expression (19) selects the smaller, because complete solution set $\mathbb{X}^0$ is a poset [6].

Property (19) provides reduction of the number of terms in (18) that we investigate to obtain lower solutions. In particular,

$$\left\langle \frac{m^*_{i_1j_1}}{j_1} \right\rangle \cdot \left\langle \frac{m^*_{i_2j_2}}{j_2} \right\rangle + \left\langle \frac{m^*_{i_1j_1}}{j_1} \right\rangle \cdot \left\langle \frac{m^*_{i_2j_2}}{j_2} \right\rangle = \left\langle \frac{m^*_{i_1j_1}}{j_1} \right\rangle \cdot \left\langle \frac{m^*_{i_2j_2}}{j_2} \right\rangle. \tag{20}$$

A property called *combined absorption* follows from (17), (19) and (20):

$$\left\langle \frac{m^*_{i_1j_1}}{j_1} \right\rangle \left[ \left\langle \frac{m^*_{i_2j_1}}{j_1} \right\rangle + \left\langle \frac{m^*_{i_2j_2}}{j_2} \right\rangle \right] =$$

$$= \left\langle \frac{m^*_{i_1j_1}}{j_1} \right\rangle \left\langle \frac{m^*_{i_2j_2}}{j_1} \right\rangle + \left\langle \frac{m^*_{i_1j_1}}{j_1} \right\rangle \left\langle \frac{m^*_{i_2j_2}}{j_2} \right\rangle = \left\langle \frac{m^*_{i_1j_1}}{j_1} \right\rangle . \tag{21}$$

After simplifying (18) according to (19) – (21) any term

$$\left\langle \frac{m^*_{i_1j_1}}{j_1} \right\rangle \left\langle \frac{m^*_{i_2j_2}}{j_2} \right\rangle \cdots \left\langle \frac{m^*_{i_mj_m}}{j_m} \right\rangle$$

determines a minimal solution $\check{X} = (\check{x}_j)$, with components (obtained after expanding brackets in (14) by rules (15) – (22)), see also Corollary 3:

$$\check{x}_{j_t} = \begin{cases} m^*_{i_tj_t} = \hat{x}_{j_t} \text{ if } m^*_{i_tj_t} \ne 0 & \text{in} \quad (18) \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

**Corollary 3** [23] For any consistent $\odot-$FLSE the minimal solutions are computable and the set of all its minimal solutions is finite.

#### 4.3.2 Method based on expansion along the non-zero elements of $M$

The proposed formalism in 4.3.1 provides the next quite simple method based on expansion along the non-zero elements of the row in $M$.

1. Take the non-zero elements of $i-$th row (for $i = 1, ..., m$) of $M$ and form the sum $M_i$, see (13).
2. Expand $M$ : From each element $m_{ij}^* \neq 0$ in $M_i$ we form a summand, consisting of $\left\langle \frac{m_{ij}^*}{j} \right\rangle$, multiplied by a submatrix $M_{ij}$ of $M$; $M_{ij}$ is obtained as follows: we delete in $M$ the $i-$th row and the $j-$th column, see (17), as well as all rows with the same $m_{ij}^* \neq 0$ - they are automatically satisfied, see (20). From the resulting submatrix we remove redundant rows, zero rows and zero columns.
3. If $i > m$ – stop, otherwise take the next $i$.

### 4.4 The algorithm

Conventional reasoning to solve $\odot-$FLSE leads to combinatorial problem [16]. Using the theoretical background from Sections 3 and 4, we devise algorithm that computes maximal and all minimal solutions (without listing duplications of minimal solutions or non-minimal solutions) and that is smaller time consuming in comparison with the algorithms given in [9], [12], [13], [14], [16].

**Algorithm** [23] for solving $A \odot X = B$.

1. Enter the matrices $A_{m\times n}$ and $B_{m\times 1}$.
2. Compute $A^* = (a_{ij}^*)$ with $a_{ij}^*$ according to (4).
3. Compute $H$, $IND$, $X_{\text{gr}}$.
4. Transform the system in normal form.
5. If $IND(i) = 0$ for some $i = 1, \cdots, m$, then the system is inconsistent and the equation(s) with $IND(i) = 0$ can not be satisfied simultaneously with the other equation(s) (that have $IND(i) \neq 0$) in the system.

   Go to Step 10.
6. If $IND(i) = 1$ for each $i = 1, ..., m$, the system is consistent with unique: maximum solution, minimum solution (expression (22)) and maximal interval solution; $X_{\text{gr}}$ contains the maximum solution; $X_{\text{low}}$ is determined according expression (22); $X_{\max}$ is determined by $X_{\text{low}}$ on the left and by $X_{\text{gr}}$ on the right.

   Go to Step 10.
7. Compute the dominance matrix $D = (d_{ij})_{m\times n}$ as described in 4.2.2.

8. Compute the matrix $M$ with elements computed by (12). Expand $M$ along non-zero elements by rows as given in 4.3.2. Simplify $W$ according to algebraic properties in 4.3.1.

9. The system is consistent, $X_{\mathrm{gr}}$ contains the maximum solution. Determine the minimal solutions according to expressions (14) – (22). Obtain the maximal interval solutions by minimal solutions and by maximum solution.

10. End.

The algorithm for solving $\odot-$FLSE is provided by Theorem 3 and its Corollaries, algebraic-logic properties of the terms as described in Section 4.3 and expansion along $M$. Based on simplifications, help and dominance matrices, as well as the matrix $M$, the algorithm has smaller computational complexity in comparison with the algorithms proposed in [9], [12], [13], [14], [16].

**Theorem 5**

If the system is consistent the maximum solution, the minimal solutions and the maximal interval solutions are computable.

For inconsistent system we can determine the equations that can not be satisfied by $A^t \diamond B$.

By this theoretical background in MATLAB workspace we develop software for computing the complete solution set or for establishing inconsistency of the system $A \odot X = B$.

# 5 Algorithm for finding optimal solution

The aim is to solve the optimization problem - to minimize the linear objective function

$$Z = \sum_{i=1}^{m} c_i x_i \tag{23}$$

subject to constrains - a fuzzy linear system of equations with $\max -$ product composition

$$A \odot X = B,\ 0 \le x_i \le 1,\ 1 \le i \le n \tag{24}$$

where $A$ stands for the matrix of coefficients, $X = (x_i)_{n\times 1}$ stands for the matrix of unknowns, $B$ is the right-hand side of the system, the $\max -$ product composition is written as $\odot$ and $c = (c_1, ..., c_m)$ is the weight (cost) vector. The aim is to minimize $Z$ subject to constrains (2).

We apply the algorithm and software from [23] for computing the greatest solution and all minimal solutions of the consistent system $A \odot X = B$. Then we solve the linear optimization problem, first decomposing (1) into two vectors with suitable components $Z' = (c'_1, c'_2, ..., c'_n)$ and $Z'' = (c''_1, c''_2, ..., c''_n)$, such that:

$$c_i = c_i' + c_i'', \text{ for each } i = 1, ..., n, \ Z = Z' + Z'',$$

$$c_i' = \begin{cases} c_i, & \text{if } c_i \geq 0, \\ 0, & \text{if } c_i < 0 \end{cases} . \tag{25}$$

$$c_i'' = \begin{cases} 0, & \text{if } c_i \geq 0, \\ c_i, & \text{if } c_i < 0 \end{cases} . \tag{26}$$

Now the original ploblem: to minimize $Z$ subject of constrains (2), is split into two problems. To minimize both

$$Z' = \sum_{i=1}^{m} c_i' x_i \tag{27}$$

and

$$Z'' = \sum_{i=1}^{m} c_i'' x_i \tag{28}$$

with constrains (2).

$Z$ takes its minimum when $Z'$ takes its minimum and $Z''$ takes its maximum. Hence for the problem (28) the optimal solution is $\hat{X} = (\hat{x_1}, ..., \hat{x_n}) = X_{gr}$, for the problem (27) the optimal solution $\breve{X} = (\breve{x_1}, ..., \breve{x_n})$ is among the minimal solutions of the system (2). In this case the optimal solution of the problem (1) is $X^* = (x_1^*, ..., x_n^*)$, where

$$x_i^* = \begin{cases} \hat{x_i^*}, & \text{if } c_i \leq 0 \\ \breve{x_i^*}, & \text{if } c_i > 0, \end{cases} . \tag{29}$$

If the aim is to maximize the linear objective function (1), we again split it, but now for the problem (28) the optimal solution is among the minimal solutions of the system (2), for the problem (27) the optimal solution is $X_{gr}$. In tis case the optimal solution of the problem (1) is $X^* = (x_1^*, ..., x_n^*)$, where

$$x_i^* = \begin{cases} \breve{x_i^*}, & \text{if } c_i \leq 0 \\ \hat{x_i^*}, & \text{if } c_i > 0, \end{cases} . \tag{30}$$

In any of these cases the optimal value is

$$Z^* = \sum_{i=1}^{n} c_i x_i^* \tag{31}$$

The algorithm for finding optimal solutions is based on the above results.

**Algorithm** for finding optimal solutions.

1. Enter the matrices $A_{m \times n}$, $B_{m \times 1}$ and the weight vector $C_{1 \times n}$.

2. Solve the system. If the system is inconsistent go to step 8.

3. Otherwise compute $X_{\rm gr}$ and all minimal solutions according to expressions (14) – (22).
4. If finding $Z_{\max}$ go to Step 6.
5. For finding $Z_{\min}$ compute $x_i^*$, $i = 1, ..., n$ according to (29). Go to Step 7.
6. For finding $Z_{\max}$ compute $x_i^*$, $i = 1, ..., n$ according to (30).
7. Compute the optimal value according to (31).
8. End.

**Example** [13]. Minimize

$$Z = -4x_1 + 3x_2 + 2x_3 + 3x_4 + 5x_5 + 2x_6 + x_7 + 2x_8 + 5x_9 + 6x_10$$

subject to

$$A \odot X = B,\ 0 \le x_i \le 1,\ 1 \le i \le 10, \tag{32}$$

where
A = 0.6 0.5 0.1 0.1 0.3 0.8 0.4 0.6 0.2 0.1 0.2 0.6 0.9 0.6 0.8 0.4 0.5 0.3 0.5 0.3 0.5 0.9 0.4 0.2 0.8 0.1 0.4 0.4 0.7 0.6 0.3 0.5 0.7 0.5 0.8 0.1 0.8 0.3 0.4 0.6 0.7 0.8 0.5 0.4 0.8 0.2 0.4 0.1 0.9 0.6 0.5 0.9 0.7 0.1 0.5 0.8 0.7 0.2 0.9 0.4 0.2 0.3 0.4 0.7 0.5 0.8 0.3 0.5 0.7 0.4 0.8 0.8 0.7 0.5 0.8 0.3 0.4 0.7 0.2 0.8
B' = 0.48 0.56 0.72 0.56 0.64 0.72 0.42 0.64

The system is consistent with
greatest Solution - transposed
ans = 0.8000 0.8000 0.6222 0.6000 0.7000 0.5250 0.7000 0.8000 0.6000 0.8000
minimal Solutions - transposed
ans = 0.8000 0.8000 0.6222 0.6000 0 0 0.7000 0 0 0 0.8000 0.8000 0.6222 0 0 0.5250 0.7000 0 0 0 0.8000 0.8000 0.6222 0 0 0 0.7000 0 0.6000 0 0.8000 0.8000 0 0.6000 0.7000 0 0 0 0 0 0.8000 0.8000 0 0 0.7000 0.5250 0 0 0 0 0.8000 0.8000 0 0 0.7000 0 0 0 0.6000 0 0 0.8000 0.6222 0.6000 0 0 0.7000 0.8000 0 0 0 0.8000 0.6222 0 0 0.5250 0.7000 0.8000 0 0 0 0.8000 0.6222 0 0 0 0.7000 0.8000 0.6000 0 0 0.8000 0 0.6000 0.7000 0 0 0.8000 0 0 0 0.8000 0 0 0.7000 0.5250 0 0.8000 0 0 0 0.8000 0 0 0.7000 0 0 0.8000 0.6000 0

# 6 Software description and some experimental results

We develop software, based on this method and algorithm in MATLAB workspace.

The algebraic - logical approach and matrix based approach are programmed as alternative programming techniques. The algebraic-logical approach uses the MATLAB library published in [22] and free available under General Public License for construction and operation with terms. This approach has two advantages – to operate only with essential (non-zero) elements of the matrix, not wasting computational time for checking duplicated or non-minimal solutions (see absorptions in 4.3.1), so directly whole branches of redundant solutions are cut.

The matrix approach is based on the operation with and within matrices, without building new structures. Applying dominance rules before each new sub-step can speed up the calculation process, and thus the method seems to be preferable for larger systems.

Theoretically both methods are equivalent. Which one is faster depends on the properties of the instant. A comparison between computational times at this moment is not suitable, because the MATLAB Environment has a set of pre-compiled functions for matrix operations, which are very fast. In contrast, our MATLAB Library with implementation of the algebraic - logical approach is currently used as not compiled set of functions, which are working slower.

We include some prints from MATLAB session. For the above Example they confirm the same results as these in Markovskii [16]:

```
Help matrix:
    0.0000    0.5000    0.0000    0.0000    0.2500    0.0000
    0.0000    0.5000    1.0000    0.0000    0.0000    0.0000
    0.3000    0.5000    1.0000    0.5000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.5000    0.2500    0.0000
    0.0000    0.0000    1.0000    0.0000    0.2500    0.0000
    0.3000    0.5000    0.0000    0.0000    0.2500    0.0000

Greatest Solution transposed =
    0.3000    0.5000    1.0000    0.5000    0.2500    1.0000

Dominance matrix initial
    0.0000    1.0000    0.0000    0.0000    1.0000    0.0000
    0.0000    1.0000    1.0000    0.0000    0.0000    0.0000
    1.0000    1.0000    1.0000    1.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    1.0000    1.0000    0.0000
    0.0000    0.0000    1.0000    0.0000    1.0000    0.0000
    1.0000    1.0000    0.0000    0.0000    1.0000    0.0000

Significant rows from the dominance matrix
    0.0000    1.0000    0.0000    0.0000    1.0000    0.0000
    0.0000    1.0000    1.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    1.0000    1.0000    0.0000
    0.0000    0.0000    1.0000    0.0000    1.0000    0.0000
    1.0000    1.0000    1.0000    1.0000    0.0000    0.0000

Minimal Solutions - transposed
         0    0.5000    1.0000    0.5000         0         0
         0    0.5000         0         0    0.2500         0
         0         0    1.0000         0    0.2500         0
```

```
Short solution summary: s =
      exists: 1
         low: [3x6 double]
    sol_numb: 3
         Xgr: [0.3000 0.5000 1 0.5000 0.2500 1]
         Ind: [5x1 double]
         hlp: [6x6 double]
           A: [6x6 double]
           B: [6x1 double]
           d: [5x6 double]
```

The presented structure consists information about the input matrix and data from different solution steps. More detailed solution summary is also available, where also solution times for the different routines are saved.

The next example is for a consistent system with three minimal solutions, while following the procedure in [12] we should obtain 18 solutions - the procedure yield to some non-minimal solutions as well.

```
A=
    0.5600    0.6000    0.2000    0.4000    0.2000    0.7000    0.7000    0.5000
    0.4200    0.3000    0.7000    0.6000    0.1000    0.3000    0.5000    0.3000
    0.5000    0.8000    0.7000    0.4000    0.7000    0.8000    0.3000    0.8000
    0.2000    0.4000    0.5000    0.1000    0.3000    0.5000    0.8000    0.4000
    0.4200    0.2000    0.5000    0.5000    0.1000    0.4000    0.7000    0.2000
    0.7200    0.9000    0.8000    0.2000    0.8000    0.6000    0.1000    0.4000

B =
    0.5600
    0.4200
    0.6400
    0.4000
    0.4200
    0.7200

Greatest Solution transposed =
    1.0000    0.8000    0.6000    0.7000    0.9000    0.8000    0.5000    0.8000

Minimal Solutions - transposed
    1.0000    0.8000         0         0         0         0    0.5000         0
    1.0000         0         0         0         0    0.8000         0         0
    1.0000         0         0         0         0         0    0.5000    0.8000
```

# 7 Conclusions

In this paper we develop exact method and universal algorithm for solving max −product fuzzy linear systems of equations and max −product fuzzy relational equations.

Various applications of inverse problem for max −product composition in finite fuzzy machines, as inference engine, for fuzzy modeling, for some optimization problems, are possible. They will be a subject of next publications.

## References


[1] A. Aho, J. Hopcroft and J. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley Publ. Co., London, 1976.

[2] L. Cheng and P. Wang, Fuzzy Relational Equations (I): the general and specialized solving algorithms, Soft Computing 6 (2002) 428-435.

[3] M. M. Bourke and D. G. Fisher, Solution algorithms for fuzzy relational equations with max-product composition, Fuzzy Sets and Systems 94 (1998) 61-69.

[4] B. De Baets, Analytical solution methods for fuzzy relational equations, in the series: Fundamentals of Fuzzy Sets, The Handbooks of Fuzzy Sets Series, D. Dubois and H. Prade (eds.), Vol. 1, Kluwer Academic Publishers (2000) 291-340.

[5] A. Di Nola, A. Lettieri, Relation Equations in Residuated Lattices, Rendiconti del Circolo Matematico di Palermo, s. II, XXXVIII (1989), pp.246-256.

[6] A. Di Nola, W. Pedrycz, S. Sessa and E. Sanchez, Fuzzy Relation Equations and Their Application to Knowledge Engineering, Kluwer Academic Press, Dordrecht, 1989.

[7] M. R. Garey and D. S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.

[8] G. Grätzer, General Lattice Theory, Akademie-Verlag, Berlin, 1978.

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

[10] M. Higashi and G. J. Klir, Resolution of finite fuzzy relation equations, Fuzzy Sets and Systems 13 (1) (1984) 65 - 82.

[11] G. Klir, U. H. St. Clair and Y. Bo, Fuzzy Set Theory Foundations and Applications, Prentice Hall PRT, 1997.

[12] J. Loetamonphong and S.-C. Fang, An efficient solution procedure for fuzzy relational equations with max-product composition, IEEE Transactions on Fuzzy Systems 7 (4) (1999) 441–445.

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

[14] J. Loetamonphong, S.-C. Fang and R. E. Young, Multi-objective optimization problems with fuzzy relation equation consrtaints, Fuzzy Sets and Systems 127 (3) (2002) 141–164.

[15] S. MacLane, G. Birkhoff, Algebra, Macmillan, New York, 1979.

[16] A. V. Markovskii, On the relation between equations with max −product composition and the covering problem, Fuzzy Sets and Systems 153 (2005) 261-273.

[17] M. Miyakoshi and M. Shimbo, Lower solutions of systems of fuzzy equations, Fuzzy Sets and Systems 19 (1986) 37-46.

[18] C. P. Pappis and M. Sugeno, Fuzzy relational equations and the inverse problem, Fuzzy Sets and Systems 15 (1985) 79 - 90.

[19] K. Peeva, Fuzzy linear systems, Fuzzy Sets and Systems 49 (1992) 339 - 355.

[20] K. Peeva, Fuzzy linear systems – Theory and applications in Artificial Intelligence areas, DSc Thesis, Sofia (2002), pp. 239 (in Bulgarian).

[21] K. Peeva, Universal algorithm for solving fuzzy relational equations, Italian Journal of Pure and Applied Mathematics 19 (2006) 9-20.

[22] K. Peeva and Y. Kyosev, Fuzzy Relational Calculus-Theory, Applications and Software (with CD-ROM), In the series 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

[23] K. Peeva and Y. Kyosev, Algorithm for Solving Max-product Fuzzy Relational Equations, Soft Computing (in press)

[24] E. Sanchez, Resolution of composite fuzzy relation equations, Information and Control 30 (1976) 38-48.