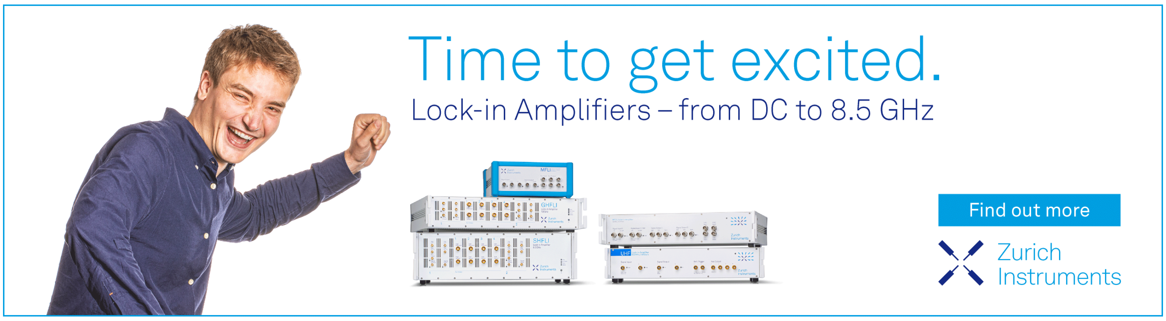
RESEARCH ARTICLE | DECEMBER 16 2016

Resolution of max $-t$-norm fuzzy linear system of equations in $BL-$algebras

K. Peeva; G. Zaharieva; Z. Zahariev

AIP Conference Proceedings 1789, 060005 (2016)
https://doi.org/10.1063/1.4968497



Resolution of max $-t$-norm Fuzzy Linear System of Equations in $BL-$algebras

K. Peeva[1,a)], G. Zaharieva[1,b)] and Z. Zahariev[1,c)]

[1]*Faculty of Appl. Mathematics, Technical University of Sofia, Sofia 1000, P.O.Box 384*

[a)]kgp@tu-sofia.bg
[b)]gpantaleeva@gmail.com
[c)]zlatko@tu-sofia

Abstract. We propose method and algorithm for inverse problem resolution of fuzzy linear system of equations in BL-algebras when the composition is max $-t-$norm.

INTRODUCTION

The notion of max-min and min-max fuzzy relational equation (FRE) was first proposed and investigated by Sanchez [21], [22] in 1972-1976 and then extended by Pedrycz [16] and Miyakoshi and Shimbo [15].

The structure of the complete solution set of sup-T equations was first characterized by Sanchez [23] and generalized to sup-T equations by Di Nola et al. [4], [5]. It now becomes well known that the complete solution set of a consistent finite system of sup-T equations can be determined by a maximum solution and a finite number of minimal solutions. The consistency of a system of sup-T equations can be easily verified by checking the potential maximum solution.

The first step for the resolution of a FRE is to establish the existence of the solution [22]. If the FRE is solvable, the solution set contains a maximum solution and several minimum solutions [11]. There exist several approaches for inverse problem resolution in the literature. De Baets [1] provided an analytical method. Peeva [19] proposed a universal algorithm which improves the algebraic method [17]. An excellent formulation of the inverse problem in fuzzy relational calculus and a review of all the literature can be found in [1], [6], [11], [12], [20], [18].

While much of the research on FREs has been done specifically for the maxmin case, there is a growing interest for more general research on sup-t-norm FREs, see [12], [2], [25], [24].

We present here inverse problem resolution in fuzzy relational calculus, when the composition is a max $-t-$norm.

In Section "BASIC NOTIONS" we introduce $t-$norms, $BL-$algebras and fuzzy matrices. Section "DIRECT AND INVERSE PROBLEMS" presents direct and inverse problem resolution for fuzzy relational calculus in $BL-$algebras. Section "INVERSE PROBLEM RESOLUTION" describes method and algorithm for inverse problem resolution for fuzzy linear system of equations, when the composition is a max $-t-$norm. Concluding section proposes ideas for next development.

Terminology for algebra, orders and lattices is given according to [8], [14], for fuzzy sets, fuzzy relations and t-norms – according to [1], [2], [6], [11], [18], for computational complexity and algorithms is as in [7].

BASIC NOTIONS

Partial order relation on a partially ordered set (poset) P is denoted by the symbol $\leq$. By a *greatest element* of a poset P we mean an element $b \in P$ such that $x \leq b$ for all $x \in P$. The *least element* of P is defined dually.

The three well known couples of $t-$norms and $s-$norms are given in Table 1.

$BL-$algebra [9] is the algebraic structure:

Applications of Mathematics in Engineering and Economics (AMEE'16)
AIP Conf. Proc. 1789, 060005-1–060005-10; doi: 10.1063/1.4968497
Published by AIP Publishing. 978-0-7354-1453-2/$30.00

TABLE 1. $t-$norms and $s-$norms

$t-$norm	name	expression	$s-$norm	name	expression
t_3	minimum, Gödel $t-$norm	$t_3(x,y) = \min\{x,\ y\}$	s_3	maximum, Gödel $t-$conorm	$s_3(x,y) = \max\{x,\ y\}$
t_2	Algebraic product	$t_2(x,y) = xy$	s_2	Probabilistic sum	$s_2(x,y) = x+y-xy$
t_1	Łukasiewicz $t-$norm	$t_1(x,y) = \max\{x+y-1,\ 0\}$	s_1	Bounded sum	$s_1(x,y) = \min\{x+y,\ 1\}$

$$BL = \langle L, \vee, \wedge, *, \rightarrow, 0, 1\rangle,$$

where $\vee, \wedge, *, \rightarrow$ are binary operations, 0, 1 are constants and:

i) $L = \langle L, \vee, \wedge, 0, 1\rangle$ is a lattice with universal bounds 0 and 1;
ii) $L = \langle L, *, 1\rangle$ is a commutative semigroup;
iii) $*$ and $\rightarrow$ establish an adjoint couple:

$$z \le (x \rightarrow y) \Leftrightarrow x * z \le y, \forall x, y, z \in L.$$

iv) for all $x, y \in L$

$$x * (x \rightarrow y) = x \wedge y \quad \text{and} \quad (x \rightarrow y) \vee (y \rightarrow x) = 1.$$

We suppose in next exposition that $L = [0,1]$ and $x,\ y \in\ [0,\ 1]$.
The following algebras are examples for $BL-$algebras.

1. Gödel algebra

$$BL_G = \langle [0,\ 1], \vee, \wedge, \rightarrow_G, 0, 1\rangle,$$

where operations are:
Maximum or s_3-norm:

$$\max\{x,\ y\} = x \vee y. \tag{1}$$

Minimum or t_3-norm:

$$\min\{x,\ y\} = x \wedge y. \tag{2}$$

The residuum $\rightarrow_G$ is

$$x \rightarrow_G y = \begin{cases} 1 \text{ if } & x \le y \\ y \text{ if } & x > y \end{cases}. \tag{3}$$

A supplementary operation ε is useful

$$x\, \varepsilon\, y = \begin{cases} y, & \text{if } x < y \\ 0, & \text{if } x \ge y \end{cases}. \tag{4}$$

2. Product (Goguen) algebra

$$BL_P = \langle [0,\ 1], \vee, \wedge, \circ, \rightarrow_P, 0, 1 \rangle,$$

where max and min are as (1) and (2), respectively, $\circ$ is the conventional real number multiplication (the t_2-norm, i. e, $t_2(x, y) = xy$) and the residuum $\rightarrow_P$ is

$$x \rightarrow_P y = \begin{cases} 1 \text{ if } & x \leq y \\ \frac{y}{x} \text{ if } & x > y \end{cases}. \tag{5}$$

Here the supplementary useful operation γ is:

$$x\gamma y = \begin{cases} 0 \text{ if } & x \geq y \\ \frac{y-x}{1-x} \text{ if } & x < y \end{cases}. \tag{6}$$

3. Łukasiewicz algebra

$$BL_L = \langle [0,\ 1], \vee, \wedge, \otimes, \rightarrow_L, 0, 1 \rangle,$$

where max and min are as (1) and (2), respectively, and

$$x \otimes y = 0 \vee (x + y - 1) \equiv t_1(x, y).$$

The residuum $\rightarrow_L$ is

$$x \rightarrow_L y = 1 \wedge (1 - x + y). \tag{7}$$

A supplementary operation δ is useful

$$x\delta y = 0 \vee (y - x). \tag{8}$$

Let $E \neq \emptyset$ be a crisp set and $A \subseteq E$. A ***fuzzy set*** $\hat{A}$ on E is described as

$$\hat{A} = \{\langle x,\ \mu_A(x) \rangle \mid x \in E\},$$

where for each $x \in E$, $\mu_A : E \rightarrow [0,\ 1]$ defines the degree of membership of the element $x \in E$ to $\hat{A}$.

A ***fuzzy relation*** (FR) between two nonempty crisp sets X and Y is a fuzzy set on $X \times Y$, written $R \in F(X \times Y)$. $X \times Y$ is called ***support*** of R.

Any FR $R \in F(X \times Y)$ is given as follows:

$$R = \{\langle (x,\ y),\ \mu_R(x,\ y) \rangle \mid (x,\ y) \in X \times Y,\ \mu_R :\ X \times Y \rightarrow [0,\ 1]\},$$

for each $(x, y) \in X \times Y$.

The matrix $A = (\mu^A_{ij})_{m\times n}$ with $\mu^A_{ij} \in [0,\ 1]$ is called a ***fuzzy matrix*** (FM) of type $m \times n$.

When the FR is over finite support, it is representable by FM, written for convenience with the same letter. For instance, if the FR $R \in F(X \times Y)$ is over finite support, its representative matrix is stipulated to be the matrix $R = (\mu^R_{x_i y_j})_{m\times n}$ such that

$$\mu^R_{x_i y_j} = \mu_R(x_i, y_j).$$

According to this stipulation we describe FR with its corresponding fuzzy matrix. Composition of FRs is presented by operations with fuzzy matrices.

DIRECT AND INVERSE PROBLEMS

Two finite FMs $A = (\mu_{ij}^{A})_{m\times p}$ and $B = (\mu_{ij}^{B})_{p\times n}$ are called ***conformable*** in this order, if the number of columns in A is equal to the number of rows in B.

Definition 1. Let $A = (\mu_{ij}^{A})_{m\times p}$ and $B = (\mu_{ij}^{B})_{p\times n}$ be finite conformable FMs. The matrix $C = (\mu_{ij}^{C})_{m\times n}$, written $C = A *_{BL} B$ in $BL-$algebra is called $\max -t$ ***product*** or $s_3 - t$ product of A and B, if for each $i,\ j, 1 \le i \le m,\ \ 1 \le j \le n$ it holds:

$$\mu_{ij}^{C} = \max{}_{k=1}^{p} \left(t_r(\mu_{ik}^{A},\ \mu_{kj}^{B}) \right), \tag{9}$$

where max is the maximum norm and t_r is a minimum norm respectively; for $r = 1,\ 2,\ 3$ see Table 1.

Definition 2. Let $A = (\mu_{ij}^{A})_{m\times p}$ and $B = (\mu_{ij}^{B})_{p\times n}$ be finite conformable FMs. The matrix $C = (\mu_{ij}^{C})_{m\times n}$, writen $C = A \rightarrow_{BL} B$ in $BL-$algebra is called:

1. $t_3 - \rightarrow_r$ *product* of A and B in $BL-$algebra, if for each $i,\ j, 1 \le i \le m,\ \ 1 \le j \le n$ it holds:

$$\mu_{ij}^{C} = t_3{}_{k=1}^{p} \left((\mu_{ik}^{A} \rightarrow_r\ \mu_{kj}^{B}) \right), \tag{10}$$

   where t_3 is the minimum norm and $\rightarrow_r$ is a residuum:
   $\rightarrow_G$ for $r = 1$ (see (3));
   $\rightarrow_P$ for $r = 2$ (see (5));
   $\rightarrow_L$ for $r = 3$ (see (7)).
2. $s_3 - \overset{r}{\rightarrow}_{BL}$ *product* of A and B in $BL-$algebra, if for each $i,\ j, 1 \le i \le m,\ \ 1 \le j \le n$ it holds:

$$\mu_{ij}^{C} = s_3{}_{k=1}^{p} \left((\mu_{ik}^{A} \overset{r}{\rightarrow}_{BL}\ \mu_{kj}^{B}) \right), \tag{11}$$

   where s_3 is the maximum norm and $\overset{r}{\rightarrow}_{BL}$ is the operation (8), (6) or (4), respectively;
   $\overset{r}{\rightarrow}_{BL} = \delta$ for $r = 1$, see (8);
   $\overset{r}{\rightarrow}_{BL} = \gamma$ for $r = 2$, see (6);
   $\overset{r}{\rightarrow}_{BL} = \varepsilon$ for $r = 3$, see (4).

Definition 3. When the finite conformable FMs $A = (\mu_{ij}^{A})_{m\times p}$ and $B = (\mu_{ij}^{B})_{p\times n}$ are given, computing the product $C = A *_{BL} B$ or $C = A \rightarrow_{BL} B$ is called ***direct problem resolution*** for $*_{BL}$ or for $\rightarrow_{BL}$ composition of the matrices A and B.

Direct problem is solvable in polynomial time. Software for direct problem resolution is given in [27]. It provides the operations in Section "BASIC NOTIONS" and Section "DIRECT AND INVERSE PROBLEMS" in MATLAB environment. The user may compute various compositions: $s_3 - t$, $t_3 - s$, s_3-operation, s_3-implication, etc.

Let A and B be conformable FMs.

i) The equation

$$A *_{BL} B = C, \tag{12}$$

where one of the matrices on the left side is unknown and the other two matrices are given, is called $*_{BL}$ ***fuzzy matrix equation***.

ii) The equation

$$A \rightarrow_{BL} B = C, \tag{13}$$

where one of the matrices on the left side is unknown and the other two matrices are given, is called $\rightarrow_{BL}$ ***fuzzy matrix equation***.

In (12) and (13) $A = (a_{ij})_{m\times p}$ stands for the FM of coefficients, $B = (b_{ij})_{p\times n}$ – for the FM of unknowns, $C = (c_{ij})_{m\times n}$ is the right-hand side of the equation, $a_{ij}, b_{ij}, c_{ij} \in [\,0,\ 1\,]$ for each $i = 1, ..., m$ and each $j = 1, ..., n$.

Solving (12) or (13) for the unknown matrix is called ***inverse problem resolution for fuzzy matrix equation in*** $BL-$***algebra***. In this paper we present inverse problem resolution for (12).

For $X = (\langle \mu_{ij}(x) \rangle)_{p\times n}$ and $Y = (\langle \mu_{ij}(y) \rangle)_{p\times n}$ the inequality

$$X \le Y$$

means $\mu_{ij}(x) \le \mu_{ij}(y)$ for each $i = 1, ..., p,\ j = 1, ..., n$.

Definition 4. For the FRE $A *_{BL} B = C$:

i) The matrix $X^0_{p\times m}$ with $x^0_{ij} \in [0,1]$, when $1 \le i \le p$, $1 \le j \le n$, is called a ***solution*** of $A *_{BL} B = C$ if $A *_{BL} X^0 = C$ holds.

ii) The set of all solutions of (12) is called ***complete solution set*** and it is denoted by $\mathbb{X}$. If $\mathbb{X} \neq \emptyset$ then (12) is called ***consistent***, otherwise it is called ***inconsistent***.

iii) A solution $\check{X} \in \mathbb{X}$ is called a ***lower*** or ***minimal solution*** of (12) if for any $X \in \mathbb{X}$ the relation $X \le \check{X}$ implies $X = \check{X}$, where $\le$ denotes the partial order, induced in $\mathbb{X}$ by the order of [0, 1]. Dually, a solution $\hat{X} \in \mathbb{X}$ is called an ***upper*** or ***maximal*** solution of (12) if for any $X \in \mathbb{X}$ the relation $\hat{X} \le X$ implies $X = \hat{X}$. When the upper solution is unique, it is called the ***greatest*** or ***maximum solution***.

We present finding the greatest solution of fuzzy relational equation (12), we also give a criterion for its consistency.

Theorem 1. Let A and C be finite FMs, and let $\boldsymbol{B}$ be the set of all matrices B, such that $A *_{BL} B = C$. Then:

i) $\boldsymbol{B} \neq \emptyset \quad \Leftrightarrow \quad A^t \to_{BL} C \in \boldsymbol{B}$;

ii) If the equation (12) is solvable for B then $A^t \to_{BL} C$ is its greatest solution.

iii) There exists polynomial time algorithm for computing $A^t \to_{BL} C$.

Here A^t denotes the transpose of A.

Corollary 1. The following statements are valid for the equation $A *_{BL} B = C$:

i) The FRE (12) is solvable iff $C = A *_{BL} (A^t \to_{BL} C)$ holds;

ii) There exists polynomial time algorithm for establishing solvability of the FRE (12) and for computing its greatest solution $A^t \to_{BL} C$.

In what follows we denote by $\hat{B}_{BL} = A^t \to_{BL} C$ the greatest solution of FRE (12).

INVERSE PROBLEM RESOLUTION

Let's focus on solving *fuzzy linear systems of equations* presented in matrix form:

$$A *_{BL} X = B \tag{14}$$

where $A = (a_{ij})_{m\times n}$ is the matrix of coefficients, $B = (b_i)_{m\times 1}$ holds for the right-hand side vector, $X = (x_j)_{1\times n}$ is the vector of unknowns and $*_{BL}$ is according to (9).

The system can be fully written as:

$$\left| \begin{array}{l} t_r(a_{11}, x_1) \vee t_r(a_{12}, x_2) \vee \ldots \vee t_r(a_{1n}, x_n) = b_1 \\ t_r(a_{21}, x_1) \vee t_r(a_{22}, x_2) \vee \ldots \vee t_r(a_{2n}, x_n) = b_2 \\ \ldots \\ t_r(a_{m1}, x_1) \vee t_r(a_{m2}, x_2) \vee \ldots \vee t_r(a_{mn}, x_n) = b_m \end{array} \right. \tag{15}$$

where $a_{ij}, b_i \in [0,1]$ are given, $x_j \in [0,1]$ marks the unknowns in the system and t_r is according to Table 1. In this paper for the indices we suppose $i = 1, ..., m,\ j = 1, ..., n$

Greatest solution

Any solvable max $-t$-norm fuzzy linear system of equations has unique greatest solution. In order to find all solutions of the solvable system, it is necessary to find its greatest solution and all of its minimal solutions. Finding the greatest solution is relatively simple task which can be used as a criteria for establishing solvability of the system. Finding all minimal solutions is reasonable only when the greatest solution exists.

Classical approach

The traditional approach to solve (15) is based on Theorem 1 and Corollary 1.

If the system (15) is solvable, its greatest solution is $\widehat{X} = (\widehat{x_j}) = A^t \to_{BL} B$.

Using this fact, an appropriate algorithm for checking consistency of the system and for finding its greatest solution can be obtained. Its computational complexity is $O(m.n^2)$. Nevertheless that it is simple, it is too hard for such a task.

More efficient approach

Here we propose a simpler way to answer both questions, simultaneously computing the greatest solution and establishing consistency of (15). Following the approach, proposed by Z. Zahariev [28] for the *max − min* case, instead of using Theorem 1, we work with four types of coefficients (S, E, G and H) and a boolean vector (*IND*).

In the system (15):

- If $r = 3$ i.e. operation is minimum ($t_3(x, y) = \min\{x, y\}$):
  - a_{ij} is called *S-type coefficient* if $a_{ij} < b_i$.
  - a_{ij} is called *E-type coefficient* if $a_{ij} = b_i$.
  - a_{ij} is called *G-type coefficient* if $a_{ij} > b_i$.
  - a_{ij} is called *H-type coefficient* if $a_{ij} \geq b_i$.
- If $r = 2$ i.e. operation is the algebraic product ($t_2(x, y) = xy$):
  - a_{ij} is called *S-type coefficient* if $a_{ij} < b_i$.
  - a_{ij} is called *E-type coefficient* if $a_{ij} = b_i = 0$.
  - a_{ij} is called *G-type coefficient* if $a_{ij} \geq b_i > 0$.
  - a_{ij} is called *H-type coefficient* if $a_{ij} \geq b_i$.
- If $r = 1$ i.e. operation is the Łukasiewicz t−norm ($t_1(x, y) = \max\{x + y - 1, 0\}$):
  - a_{ij} is called *S-type coefficient* if $a_{ij} - 1 > b_i$.
  - a_{ij} is called *E-type coefficient* if $a_{ij} = b_i = 0$.
  - a_{ij} is called *G-type coefficient* if $a_{ij} \geq b_i > 0$.
  - a_{ij} is called *H-type coefficient* if $a_{ij} - 1 \leq b_i$.

The algorithm uses the fact that it is possible to find the value of the unknown $\widehat{x_j}$ only by the j^{th} column of the matrix A. For every $i = 1, ..., m$ and depending on the operation, we consider the following cases:

- When the operation is t_3
  - If a_{ij} is E-type coefficient then the i-th equation can be satisfied by $a_{ij} \wedge x_j$ when $x_j \geq b_i$ because $a_{ij} \wedge x_j = b_i \wedge x_j = b_i$.
  - If a_{ij} is G-type coefficient then the i-th equation can be satisfied by $a_{ij} \wedge x_j$ only when $x_j = b_i$ because $a_{ij} \wedge x_j = a_{ij} \wedge b_i = b_i$.
  - If a_{ij} is S-type coefficient then the i-th equation cannot be satisfied by $a_{ij} \wedge x_j$ for any $x_j \in [0, 1]$.
- When the operation is t_2
  - If a_{ij} is E-type coefficient then the i-th equation can be satisfied by $a_{ij}x_j$ when $x_j \in [0, 1]$ because $a_{ij}x_j = 0x_j = b_i = 0$.
  - If a_{ij} is G-type coefficient then the i-th equation can be satisfied by $a_{ij}x_j$ only when $x_j = b_i/a_{ij}$ because $a_{ij}x_j = a_{ij}(b_i/a_{ij}) = b_i$.
  - If a_{ij} is S-type coefficient then the i-th equation cannot be satisfied by $a_{ij}x_j$ for any $x_j \in [0, 1]$.
- When the operation is t_1
  - If a_{ij} is E-type coefficient then the i-th equation can be satisfied by $a_{ij} \otimes x_j$ when $x_j \in [0, 1]$ because $a_{ij} \otimes x_j = \max\{0 + x_j - 1,\ 0\} = \max\{x_j - 1,\ 0\} = b_i = 0$.
  - If a_{ij} is H-type coefficient then the i-th equation can be satisfied by $a_{ij} \otimes x_j$ only when $x_j = 1 - a_{ij} + b_i$ because $a_{ij} \otimes x_j = a_{ij} + 1 - a_{ij} + b_i - 1 = b_i$.
  - If a_{ij} is S-type coefficient then the i-th equation cannot be satisfied by $a_{ij} \otimes x_j$ for any $x_j \in [0, 1]$.

Hence, S-type coefficients are not interesting because they do not lead to solution.

For the purposes of the next theorem, $\widehat{b_j}$ is introduced as follows:

- If the operation is t_3:

$$\widehat{b_j} = \begin{cases} \min_{i=1}^{m} \{b_i\}, \text{ for all } i \text{ such that } a_{ij} > b_i \\ 1 \text{ otherwise} \end{cases} \quad (16)$$

- If the operation is t_2:

$$\widehat{b_j} = \begin{cases} \min_{i=1}^{m} \{b_i/a_{ij}\}, \text{ for all } i \text{ such that } a_{ij} > b_i \\ 1 \text{ otherwise} \end{cases} \quad (17)$$

- If the operation is t_1:

$$\widehat{b_j} = \begin{cases} \min_{i=1}^{m} \{1 - a_{ij} + b_i\}, \text{for all } i \text{ such that } a_{ij} - 1 < b_i \\ 1 \text{ otherwise} \end{cases} \quad (18)$$

Theorem 2. [6] The system $A *_{BL} X = B$ is solvable iff $\widehat{X} = (\widehat{b_j})$ is its solution. □

Corollary 1. In a solvable system (15), choosing $x_j > \widehat{b_j}$ for at least one $j = 1, ..., n$ makes the system inconsistent.

Proof. Suppose $\widehat{b_j} = b_k \neq 1$. Let we choose $x_j > b_k$. This means that the left-hand side of the k^{th} equation is greater than b_k and this proves the theorem. □

Corollary 2. In a solvable system (15), for every $j = 1, ..., n$, the greatest admissible value for x_j is $\widehat{b_j}$. □

Corollary 3. If the system (15) is solvable, its greatest solution is $\widehat{X} = (\widehat{x_j}) = (\widehat{b_j})$, $j = 1, ..., n$. □

In general, Theorem 2 and its corollaries show that instead of calculating $\widehat{X} = A^t \to_{BL} B$ we can use faster algorithm to obtain $\widehat{X} = (\widehat{x_j}) = (\widehat{b_j})$ (presented further in the paper).

$\widehat{X}$ is only the eventual greatest solution of the system (15), because it can be obtained for any system (15), even if the system is unsolvable, so the eventual solution should be checked in order to confirm that it is solution of (15). Explicit checking for the eventual solution will increase the computational complexity of the algorithm. To avoid this in the next presented algorithm this is done by the extraction of the coefficients of the potential greatest solution. For every $(\widehat{x_j}) \in \widehat{X}$ we check, which equations of (15) are satisfied (hold in the boolean vector IND). If in the end of the algorithm all the equations of (15) are satisfied (i.e. all the coefficients in IND are set to $TRUE$) this means that the system is consistent and the computed solution is its greatest solution, otherwise the system (15) is unsolvable.

Algorithm 1 Greatest solution of (15).

1. Initialize the vector $\widehat{X} = (\widehat{x_j})$ with $\widehat{x_j} = 1$ for $j = 1, ..., n$.
2. Initialize a boolean vector IND with $IND_i = FALSE$ for $i = 1, ..., m$. This vector is used to mark equations that are satisfied by the eventual greatest solution.
3. For each column $j = 1, ..., n$ in A: walk successively through all coefficients $a_{ij}, i = 1, ..., m$ seeking the smallest G-type coefficient.
   (a) If a_{ij} is E-type coefficient, according to (16), (17) or (18) respectively (depending on the operation), it means that the i^{th} equation in the system can be solved through this coefficient, but $\widehat{b_j}$ still should be found. Correct IND_i to $TRUE$.
   (b) For the smallest H-type (which is not E-type) coefficient correct IND_i to $TRUE$. All other H-type (which are not E-type) coefficients are now insignificant, as they are not leading to a solution. In $\widehat{X}$ correct the value for $\widehat{x_j} = \widehat{b_j}$.
   Go to the next j.
4. Check if all components of IND are set to $TRUE$.
   (a) If $IND_i = FALSE$ for some i the system $A *_{BL} X = B$ is inconsistent.
   (b) If $IND_i = TRUE$ for all $i = 1, ..., m$ the system $A *_{BL} X = B$ is consistent and its greatest solution is stored in $\widehat{X}$.
5. Exit.

Theorem 2 and its corollaries provide that if the system is consistent, $\widehat{X}$ computed by Algorithm 1 is its greatest solution. With Algorithm 1 $\widehat{X}$ can be obtained in efficient way. In addition there is no need to substitute $\widehat{X}$ in order to establish consistency of the system.

IND vector proposed first in [17] here is used for a similar purpose. The algorithm uses this vector to check which equations are satisfied by the eventual $\widehat{X}$. At the end of the algorithm if all components in IND are $TRUE$ then $\widehat{X}$ is the greatest solution of the system, otherwise the system is inconsistent.

Lower solutions

It is important that every equation in the system (15) can be satisfied only by the terms with H-type coefficients. Also, the minimal value for every component in the solution is either the value of the corresponding $\hat{b}$ or 0. Along these lines, the hearth of the presented here algorithm is to find H-type components a_{ij} in A and to give to X_{low_j} either the value of the corresponding $\hat{b}$ when the coefficient contributes to solve the system or 0 when it doesn't.

Using this, the set of candidates for solutions can be obtained. All candidate solutions are of three different types:

- Lower solution;
- Non-lower solution;
- Not solution at all.

The aim of the algorithm is to extract all lower solutions and to skip the second and third types. In order to extract all lower solutions a new method, based on the idea of the *dominance matrix* [20] in combination with list manipulation techniques is developed here.

Domination

For the purposes of presented here algorithm, a modified version of the definition for domination is given. Original definition can be found in [20].

Definition 1. Let a_l and a_k be the l^{th} and the k^{th} equations, respectively, in (15) and $b_l \geq b_k$. Equation a_l is called dominant to a_k and equation a_k is called dominated by a_l, if for each $j = 1, ..., n$ it holds: if a_{lj} is H-type coefficient then a_{kj} is also H-type coefficient.

Extracting lower solutions

Lower solutions are extracted by removing from A the dominated rows. A new matrix is produced and marked with $\widetilde{A} = (a_{\widetilde{i}j})$ where $\widetilde{i} = 1, ..., \widetilde{m}$, $\widetilde{m} < m$ for obvious reasons. It preserves all the needed information from A to obtain the solutions.

Extraction introduced here is based on the following recursive principle. If in the j^{th} column of $\widetilde{A}$ there are one or more rows $(\widetilde{i^*})$ such that coefficients $a_{\widetilde{i^*}j}$ are H-type then x_j should be taken equal to the smallest $b_{\widetilde{i^*}}$ and all rows $\widetilde{i^*}$ should be removed from $\widetilde{A}$. The same procedure is repeated for $(j + 1)^{th}$ column of the reduced $\widetilde{A}$. "Backtracking" based algorithm using this principle is presented next:

Algorithm 2 Extract the lower solutions from $\widetilde{A}$.

1. Initialize solution vector $X_{low_0}(j) = 0,\ j = 1, ..., n$.
2. Initialize a vector $rows(\widetilde{i}),\ i = 1, ..., \widetilde{m}$ which holds all consecutive row numbers in $\widetilde{A}$. This vector is used as a stopping condition for the recursion. Initially it holds all the rows in $\widetilde{A}$. On every step some of the rows there are removed. When $rows$ is empty the algorithm exits from the current recursive branch.
3. Initialize $sols$ to be the empty set of vectors, which is supposed to be the set of all minimal solutions for current problem.
4. Check if $rows = \emptyset$. If so, add X_{low_0} to $sols$ and go to step 7.
5. Fix $\widetilde{i}$ equal to the first element in $rows$, then for every $j = 1, ..., n$ such that $a_{\widetilde{i}j}$ is H-type coefficient
   (a) Create a copy of X_{low_0} and update its j^{th} coefficient to be equal to $b_{\widetilde{i}}$. Create a copy of $rows$.
   (b) For all $\widetilde{k}$ in $rows$ if $a_{\widetilde{k}j}$ is a H-type coefficient, remove $\widetilde{k}$ from the copy of $rows$.
   (c) Go to step 5 with copied in this step $rows$ and X_{low_0}, i.e. start new recursive branch with reduced $rows$ and changed X_{low_0}.
6. Exit.

General algorithm

The next algorithm is based on the above given Algorithms 1 and 2.

Algorithm 3 Solving $A *_{BL} X = B$.

1. Obtain input data for the matrices A and B.
2. Obtain the greatest solution for the system and check it for consistency (Algorithm 1).
3. If the system is inconsistent go to step 6.
4. Obtain the matrix $\widetilde{A}$.
5. Obtain all minimal solutions from $\widetilde{A}$ and B (Algorithm 2).
6. Exit.

Algorithm 2 (Step 5) is the slowest part of the Algorithm 3. In general this algorithm has its best and worst cases and this is the most important improvement according to algebraic-logical approach from [20] (from the time complexity point of view). Algorithm 2 is going to have the same time complexity as the algorithm presented in [20] only in its worst case.

CONCLUSIONS

The proposed software [27] is for direct and inverse problem resolution. The next step for investigation and realization will be the inf $-s$-norm case.

REFERENCES

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

[2] **R. Belohlavek**, *Sup-t-norm and inf-residuum are one type of relational product: Unifying framework and consequences*, Fuzzy Sets and Systems 197, 45-58, 2012.

[3] **L. Chen, P. Wang**, *Fuzzy relational equations (I): the general and specialized solving algorithm*, Soft Computing 6, 428-435, 2002.

[4] **A. Di Nola, W. Pedrycz, S. Sessa**, *On solution of fuzzy relational equations and their characterization.*, BUSEFAL, 12, 60-71, 1982.

[5] **A. Di Nola, W. Pedrycz, S. Sessa, P. Z. Wang**, *Fuzzy relation equations under triangular norms: a survey and new results*, Stochastica, 8, 99-145, 1984.

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

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

[9] **P. Hájek**, *Metamathematics of Fuzzy Logic*, Kluwer Academic Publishers, Dordrecht, 1998.

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

[11] **G. Klir, U. Clair, B. Yuan**, *Fuzzy Set Theory Foundations and Applications*, Prentice Hall PRT, 1997.

[12] **P. Li, S. Fang**, *On the resolution and optimization of a system of fuzzy relational equations with sup-T composition*, Fuzzy Optim Decis Making (2008) Vol. 7 Issue 2, 169-214, 2008.

[13] **P. Li, S. Fang**, *A survey on fuzzy relational equations, part I: classification and solvability*, Fuzzy Optimization and Decision Making, 2009.

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

[15] **M. Miyakoshi, M. Shimbo**, *Solutions of composite fuzzy relational equations with triangular norms*, Fuzzy Sets and Systems, 16, 53-63, 1985.

[16] **W. Pedrycz**, *Fuzzy relational equations with triangular norms and their resolutions*, BUSEFAL, 11, 2432, 1982.

[17] **K. Peeva**, *Fuzzy linear systems*, Fuzzy Sets and Systems, Volume 49, Issue 3, 339-355, 1992.

[18] **K. Peeva, 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.

[19] **K. Peeva**, *Universal algorithm for solving fuzzy relational equations*, Italian J. Pure Appl. Math., 19, 169-188, 2006.

[20] **K. Peeva**, *Resolution of fuzzy relational equations - Method, algorithm and software with applications*, Information Sciences: an International Journal, Volume 234, 44-63, 2013

[21] **E. Sanchez**, *Equations de Relations Floves*, Thise Biologie Humaine, Marseille, France, 1972.

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

[23] **E. Sanchez**, *Solutions in composite fuzzy relation equations: application to medical diagnosis in Brouwerian logic*, M. M. Gupta, G. N. Saridis, B. R. Gaines (Eds.), Fuzzy automata and decision processes, 221-234, Amsterdam: North-Holland, 1977.

[24] **B. Shieh**, *Solutions of fuzzy relation equations based on continuous t-norms*, Inform. Sci, 177, 4208-4215, 2007.

[25] **G. Stamou, S. Tzafestas**, *Resolution of composite fuzzy relation equations based on Archimedean triangular norms*, Fuzzy Sets and Systems, 120, 395-407, 2001.

[26] **Y. Wu, S. Guu**, *Finding the complete set of minimal solutions for fuzzy relational equations with max-product composition*, International Journal of Operations Research Vol. 1, No. 1, 2936, 2004.

[27] **Z. Zahariev**, http://www.mathworks.com/matlabcentral/fileexchange/27046-fuzzy-calculus-core-fc2ore, 2010.

[28] **Z. Zahariev**, *Methods and algorithms in fuzzy relational calculus with some applications in the artificial intelligence*, Ph.D. Thesis, Dept. of Algebra and Geometry, Technical University of Sofia, Bulgaria, 2013 (in Bulgarian).