RESEARCH ARTICLE | OCTOBER 30 2008

Solving Max-min Relational Equations. Software and Applications

Z. Zahariev

AIP Conference Proceedings 1067, 516–523 (2008)
https://doi.org/10.1063/1.3030826

Solving Max-min Relational Equations. Software and Applications

Z. Zahariev

Faculty of Applied Mathematics and Informatics, Technical University of Sofia
1000 Sofia, Bulgaria

Abstract. Method, algorithm and software for solving max-min relational equations are presented. They are based on an existing algorithm, but with some significant improvements, developed by the author. Applications with software implementation are also provided.

Keywords: Fuzzy equations, max-min, algorithm, software, applications.
PACS: 01.30.Cc, 01.50.hv, 02.10.-v, 07.05.Mh

INTRODUCTION

This paper is focused on solving fuzzy max-min linear systems of equations (FLSE) with some applications. More specific, here is presented a new software, intended to deal with this problems. This software is based on algorithm presented in [5], but introduces a lot of improvements, which highly rise its productivity. Since there is also available software, based strictly on the original algorithm [6], in the article is provided comparison between this two different programs.

The studied applications are actually problems which are solved by use at first the base algorithm and software, and then using presented here modifications and new software.

PRELIMINARIES

In this section all needed preliminaries will be defined. The main problem, studied in this paper, will be promoted here, as well. All notions and symbols will be used according to definitions given in this section.

Basic Definitions

Definition 1 A matrix $A = (a_{ij})_{m\times n}$, where $a_{ij} \in [0,1]$ for each $i = \overline{1,m}$, $j = \overline{1,n}$, is called *membership* or *fuzzy matrix*. Fuzzy matrix with $n = 1$ is called *fuzzy vector*.

In the further statement 'matrix' is used instead of 'membership matrix' or 'fuzzy matrix', 'vector' is used instead of 'fuzzy vector'.

Definition 2 Matrices $A = (a_{ij})_{m\times p}$ and $B = (b_{jk})_{p\times n}$, where p is the number of the columns in the matrix A and the number of the rows in the matrix B, are called *conformable*.

Definition 3 Inequality between vectors $X = (x_j)_{n\times 1}$ and $Y = (y_j)_{n\times 1}$, marked as $X \le Y$ or $X \ge Y$ is defined as follows:

$$X \le Y \text{ if } x_j \le y_j \text{ for each } j = \overline{1,n}. \tag{1}$$

or

$$X \ge Y \text{ if } x_j \ge y_j \text{ for each } j = \overline{1,n}. \tag{2}$$

Operations

Definition 4 Operation $\vee$ is defined as:

$$a \vee b = \max(a,b). \tag{3}$$

Definition 5 Operation $\wedge$ is defined as:

$$a \wedge b = min(a,b). \tag{4}$$

Definition 6 Operation α is defined as:

$$a\alpha b = \begin{cases} 1, \text{ if } a \le b \\ b, \text{ if} a > b \end{cases} \tag{5}$$

Products

Definition 7 $Max - min$ product, marked as $\bullet$ $(1 \le i \le m,\ 1 \le j \le n)$:

$$A \bullet B = X = (x_{ij})_{m\times n} \quad \text{where} \quad x_{ij} = \overset{n}{\underset{k=1}{\vee}} (a_{ik} \wedge b_{kj}) \tag{6}$$

Definition 8 $Min - \alpha$ product, marked as α $(1 \le i \le m,\ 1 \le j \le n)$:

$$A\alpha B = X = (x_{ij})_{m\times n} \quad \text{where} \quad x_{ij} = \overset{n}{\underset{k=1}{\wedge}} (a_{ik}\, \alpha\, b_{kj}) \tag{7}$$

Max-min Fuzzy Linear System of Equations

Max-min FLSE is a system of the following form:

$$\left| \begin{array}{l} (a_{11} \wedge x_1) \vee (a_{12} \wedge x_2) \vee \ldots \vee (a_{1m} \wedge x_m) = b_1 \\ (a_{21} \wedge x_1) \vee (a_{22} \wedge x_2) \vee \ldots \vee (a_{2m} \wedge x_m) = b_2 \\ \ldots \\ (a_{n1} \wedge x_1) \vee (a_{n2} \wedge x_2) \vee \ldots \vee (a_{nm} \wedge x_m) = b_n \end{array} \right. \tag{8}$$

where $a_{ij}, b_i, x_j \in [0,1]$, $i = \overline{1,n}$, $j = \overline{1,m}$, here x_i marks the unknown values in the system.

Same system can be represented by its matrix form, written as follows:

$$A \bullet X = B \tag{9}$$

where $A = (a_{ij})_{m\times n}$ is a matrix with the coefficients of the system (8), $B = (b_i)_{m\times 1}$ is the right hand side of (8) and $X = (x_j)_{1\times n}$ is a matrix with all unknown values of (8)

Types of Solutions for the a FLSE

Definition 9 $X^0 = (x_j^0)_{n\times 1}$ with $x_j^0 \in [0,1]$ and $j = \overline{1,n}$ is called point solution of the system (9) if $A \bullet X^0 = B$ holds.

Definition 10 The set of all point solutions of (9) is called complete solution set and is denoted by $\mathbb{X}^0$.

Definition 11 A solution $X_{low}^0 \in \mathbb{X}^0$ is called a lower solution of (9) if there is not such a $X^0 \in \mathbb{X}^0$ for which $X^0 \leq X_{low}^0$.

Definition 12 A solution $X_u^0 \in \mathbb{X}^0$ is called an upper solution of (9) if there is not such a $X^0 \in \mathbb{X}^0$ for which $X_u^0 \leq X^0$.

Definition 13 When there is unique upper solution, it is called greatest (or maximum) solution.

It is well known fact that every $max - min$ FLSE has unique upper solution (i.e. greatest solution) and many lower solutions [5].

Definition 14 $(X_1,...,X_n)$ with $X_j \subset [0,1]$ for each j, $1 \leq j \leq n$, is called an interval solution of the system (9) if any $X^0 = (x_j^0)_{n\times 1}$ with $x_j^0 \in X_j$ for each j, $1 \leq j \leq n$, implies $X^0 = (x_j^0)_{n\times 1} \in \mathbb{X}^0$.

Definition 15 Any interval solution of (9) whose components (interval bounds) are determined by a lower solution from the left and by the greatest solution from the right, is called maximal interval solution of (9).

Definition 16 If for a given system (9), $\mathbb{X}^0 \neq \emptyset$ then the system is called consistent, otherwise it is called inconsistent.

SOLVING FLSE

In spite of that in the last years the interest in solving FLSE increases rapidly and many results were published in this area, there are still only few relatively simple and strain forward algorithms and less than few implemented programs [5], [6], [2], [1], [4], [7]. Software implementation of such algorithms can be found in [6]. Most of the algorithms are still too complicated and sometimes confusing, thus it is hard to write software based on them.

In this work the algorithm presented in [5] is used, because it is strain forward, and a prove for this is the available software implementation of it in [6]. However by this moment a lot of improvements has been done to the original algorithm in order to decrease its complexity and increase its simplicity. Although the original software was used at the beginning of the work discussed here, a complete new program is used for it at this moment.

In this section just brief description of the original algorithm, is given. Many of the notions will not be introduced as well as most of the details. The point of the section is just to present the algorithm and its steps for producing the complete solution set of a FLSE and to show where the improvements are made. More descriptive information of the algorithm can be found in [5], [6], [7]. The original as well as the new software can be obtained for free and are distributed over GPL license [13].

Original Algorithm

Step 1 Rearrangement of the equations

Step 2 Obtain associated system

Step 3 Obtain *IND* vector and the greatest solution of the system. If they do not exist then the system is incompatible - Go to End. If they exist the system is compatible - Go to Step 4.

Step 4 Obtain help matrix

Step 5 Obtain dominance matrix form the help matrix.

Step 6 Use the algebraic-logical method described in [5], [6], [7] to find the minimal solutions of the system from its dominance matrix.

Step 7 Obtain all maximal interval solutions from the greatest solution and from the set of all minimal solution.

Improvements of the Original Algorithm

Step 1 Instead of rearranging the whole system (matrix A with the coefficients and vector B with the right hand side of the system) it is with less computational complexity to arrange only vector B, save the indexes of arrangement in another vector and use it for any further indexing in the algorithm.

Step 2 Although it is really easier for humans to solve a FLSE using its associated system, from programmers point of view there is no difference. That is why this step is totally omitted in the new program.

Step 3 This step is kept as is.

Step 4, Step 5 and Step 6 In general these three steps are kept too, but they are performed simultaneously and in the end, the dominance matrix is produced by obtaining the help matrix and use the algebraic-logical method used in Step 6 on every row of the help matrix, row by row. In this manner the program makes less cycles with bigger complexity but in the end it is much faster, especially for bigger systems.

Step 7 This step is kept as is.

TABLE 1. Execution time comparison

Size	Base algorithm	New algorithm
5×5	0.0195s	0.0016s
9×9	15.9913s	0.0064s
18×14	4.4776s	0.0116s
32×13	1143.9373s	0.1055s

Many other "little" improvements were made to the original program on order to get it faster. Some of them are preallocating of most of the used variables, reuse of variables, etc.

In addition, the way in which the algebraic-logical method from Step 6 is used, is partially changed in the new program, which constitutes the biggest part of the improvement of the execution time of the program.

Programs Comparison

The result of all these little improvements is one great improvement of the whole program. Table 1 contains comparison results of running the two programs on same machine and same examples. This results are obtained on a computer with AMD Sempron 2800+ processor working on 1.6 GHz with 512Mb RAM, using MATLAB7.

APPLICATIONS

All described here applications have implemented programs also available under GPL license. More specific these are:

- Solving fuzzy linear system of inequalities (FLSI) of $\geq$ form.
- Check for linear dependence and linear independence in fuzzy algebra.
- Computing behavior matrix of finite fuzzy machines (FFM), and minimization of FFM.
- Fuzzy optimization.

Fuzzy Linear Systems of Inequalities (FLSI) [13]

This problem is actually easy solvable once we had technique to solve system of equations. The only difference between FLSE and FLSI is the maximum solution. While the maximum solution of FLSE is a vector which have to be computed by one or other technique, the maximum solution of FLSI is always vector from ones. Keeping this in mind, we can use vector from ones for the greatest solution and obtain all lower solutions and all maximal interval solutions of the system using described algorithm and software.

Check for Linear combination, Linear Dependence or Linear Independence over Max-min Fuzzy Algebra [7], [10]

Definition 17 An expression of the form

$$(\lambda_1 \wedge v_1) \vee (\lambda_2 \wedge v_2) \vee \ldots \vee (\lambda_n \wedge v_n) \tag{10}$$

where v_i are fuzzy vectors and $\lambda_1, ..., \lambda_n \in [0,1]$ for $i = \overline{1,n}$ is called $max - min$ linear combination of the fuzzy vectors v_i with coefficients λ_i for $i = \overline{1,n}$.

Definition 18 The fuzzy vector B is called $max - min$ linear combination of the fuzzy vectors $A = a_1, a_2, ..., a_n$ if there exist X such that $A \bullet X = B$.

With direct usage of the above definition and presented here software, it is easy to determine if some vector is a linear combination of set of vectors.

Definition 19 The set $A = a_1, ..., a_n$ of fuzzy vectors is called $max - min$ linearly dependent if one of the vectors from A can be expressed as a $max - min$ linear combination of the others.

Below is a simple algorithm for checking for linear dependence or independence, which uses described algorithm for solving FLSE:

Step 1 Enter $A = a_1, ..., a_n$.

Step 2 Solve n-times a system of the form $A \bullet X = B$. Every time omit (initialize with zeros) a vector from the matrix A and set it to the right-hand side. If each system is inconsistent, then the set $A = a_1, ..., a_n$ is max-min linearly independent, otherwise it is max-min linearly dependent.

Computing Behavior Matrix and Minimization of Finite Fuzzy Machines (FFM) [7] [11], [12], [8]

Definition 20 A finite fuzzy machine (FFM) is a quadruple $\mathscr{A} = (X, Q, Y, \mathscr{M})$, where:

- X, Q, Y are nonempty finite sets of input letters, states and output letters, respectively.
- $\mathscr{M}$ is the set of transition-output matrices of $\mathscr{A}$, that determines its stepwise behavior. Each matrix $M(x|y) = (m_{qq'}(x|y)) \in \mathscr{M}$ is a square matrix of order $|Q|$ and $x \in X$, $y \in Y$, $q, q' \in Q$, $m_{qq'}(x|y) \in [0,1]$.

Definition 21 Let $\mathscr{A} = (X, Q, Y, \mathscr{M})$ be max-min FFM.

For any $(u|v) \in (X|Y)^*$ the extended input-output behavior of $\mathscr{A}$ is determined by the square matrix $M(u|v)$ of order $|Q|$:

$$M(u\,|\,v) = \begin{cases} M(x_1\,|y_1) \bullet \ldots \bullet M(x_k\,|y_k), & \\ \qquad \text{if} \quad (u|v) = (x_1 \ldots x_k|y_1 \ldots y_k), k \geq 1 & \\ U, \quad \text{if} \quad (u|v) = (e|e) & \end{cases}, \tag{11}$$

where $U = (\delta_{ij})$ is the square matrix of order $|Q|$ with elements δ_{ij} determined as follows:

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} . \tag{12}$$

Definition 22 Complete input-output behavior matrix ($T_{\mathscr{A}}$) for a FFM ($\mathscr{A}$) is determined as follows:

$$T(u|v)_{|Q|\times 1} = \big(t_q(u|v)\big) = \begin{cases} M(u|v) * E, \text{ if } (u|v) \neq (e|e); \\ E, \text{ if } (u|v) = (e|e), \end{cases} \tag{13}$$

where E is the $|Q| \times 1$ column-matrix with all elements equal to 1.

Definition 23 Behavior matrix ($B_{\mathscr{A}}$) for a FFM ($\mathscr{A}$) is obtained from $T_{\mathscr{A}}$ by removing all columns in $T_{\mathscr{A}}4$ which are linear combination of the columns before them.

From Definition 23 it is visible, that all linear dependent columns of the matrix $T_{\mathscr{A}}$ have to be determined and removed. This is possible by using the algorithm (and software) for checking for linear dependence. By using this algorithm, next follows an algorithm for obtaining $B_{\mathscr{A}}$:

Step 1 Enter the set of matrices $\mathscr{M}$.
Step 2 Find $k \in \mathbb{N}$, such that $T(k) \cong T(k+1)$.
Step 3 Obtain $B(k) = B_{\mathscr{A}}$ excluding all linear combinations from $T(k)$.
Step 4 End.

Definition 24 A FFM $\mathscr{A}$ is in minimal form if there are not any rows in $B_{\mathscr{A}}$ which are linear combination of the other rows.

Step 1 Enter the set of matrices $\mathscr{M}$ and obtain $B_{\mathscr{A}}$.
Step 2 Exclude all rows which are linear combination of the other rows from $B_{\mathscr{A}}$.
Step 3 End.

Fuzzy Optimization [3] [9]

Fuzzy optimization problem is optimization problem with linear objective function:

$$Z = \sum_{j=1}^{n} c_j x_j \tag{14}$$

whit traditional addition and multiplication, where $c_j \in \mathbb{R}$, $0 \leq x_j \leq 1$, $1 \leq j \leq n$, which is subject to a FLSE ($A \bullet X = B$) of FLSI ($A \bullet X \geq B$) as constraint.

In general the algorithm for optimization needs to split the negative and non negative coefficients of the objective function into two sub functions:

$$Z' = \sum_{j=1}^{n} c_j^{+} x_j \tag{15}$$

$$Z'' = \sum_{j=1}^{n} c_j^- x_j \tag{16}$$

Then depending on the type of optimization (minimization or maximization) one of the function reach its extremum from the greatest solution of the system of constraint and the other reach its extremum among the minimal solutions. This meant that to solve this optimization problem there is a need of finding the complete solution set for the system of constraint for the optimization problem. Described in this paper algorithm can be used here as well as in the other showed examples.

REFERENCES

1. B. De Baets, "Analytical solution methods for fuzzy relational equations", in the series *Fundamentals of Fuzzy Sets, The Handbooks of Fuzzy Sets Series*, Vol. 1, edited by D. Dubois and H. Prade, Kluwer, Academic Publishers, 2000, pp. 291–340.
2. L. Cheng and P. Wang, *Soft Computing* **6**, 428–435 (2002).
3. S.-G. Fang and G. Li, *Fuzzy Sets and Systems* **103**, 107–113 (1999).
4. M. Miyakoshi and M. Shimbo, *Fuzzy Sets and Systems* **19**, 37–46 (1986).
5. K. Peeva, *Italian Journal of Pure and Applied Mathematics* **19**, 9–20 (2006).
6. 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/`
7. K. Peeva, "Fuzzy linear systems – Theory and applications in Artificial Intelligence areas", DSc Thesis, Sofia, 2002. (in Bulgarian)
8. K. Peeva and Z. Zahariev, "Computing Behavior of Finite Fuzzy Machines – Algorithm and its Application to Reduction and Minimization", *Information Sciences*, Manuscript number: INS-D-07-727R1.
9. K. Peeva, Z. Zahariev, and I. Atanasov, "Software for optimization of linear objective function with fuzzy relational constraint," in *Intern. Conf. on Intelligent Systems, September 2008, Varna, Bulgaria.* (accepted)
10. K. Peeva and Z. Zahariev, "Software for testing linear dependence in Fuzzy Algebra", in *Intern. Sci. Conf. Computer Science, September 2005, Chalkidiki, Greece*, 2005, pp. 294–299.
11. E. S. Santos, *J. Math. Anal. Appl* **24**, 246–259 (1968).
12. E. S. Santos, *J. Math. Anal. Appl* **40**, 60–78 (1972).
13. `http://www.gnu.org/copyleft/gpl.html`