Two-stage Algorithm for solving Arbitrary Trapezoidal Fully Fuzzy Sylvester Matrix Equations

Ahmed Abdel Aziz Elsayed [1,*], Bassem Saassouh [2], Nazihah Ahmad [1], Ghassan Malkawi [3]

1 School of Quantitative Sciences, Universiti Utara Malaysia, Malaysia; nazihah@uum.edu.my

2 Department of Mathematics, Abu Dhabi Polytechnic college, United Arab Emirates; bassem.saassouh@adpoly.ac.ae

3 Department of Mechanical Engineering, Higher Colleges of Technology, United Arab Emirates; gmalkawi@hct.ac.ae

* Correspondence: ahmed_ahmed_abdel@ahsgs.uum.edu.my, ahmed.elsayed@aths.ac.ae; https://orcid.org/0000-0002-9255-200X

Abstract: Many authors proposed analytical methods for solving fully fuzzy Sylvester matrix equation (FFSME) based on Vec-operator and Kronecker product. However, these methods are restricted to nonnegative fuzzy numbers and cannot be extended to FFSME with near-zero fuzzy numbers. The main intention of this paper is to develop a new numerical method for solving FFSME with near-zero trapezoidal fuzzy numbers that provides a wider scope of trapezoidal fully fuzzy Sylvester matrix equation (TrFFSME) in scientific applications. This numerical method can solve the trapezoidal fully fuzzy Sylvester matrix equation with arbitrary coefficients and find all possible finite arbitrary solutions for the system. In order to obtain all possible fuzzy solutions, the TrFFSME is transferred to a system of non-linear equations based on newly developed arithmetic fuzzy multiplication between trapezoidal fuzzy numbers. The fuzzy solutions to the TrFFSME are obtained by developing a new two-stage algorithm. To illustrate the proposed method numerical example is solved.

Keywords Fully fuzzy Sylvester matrix equations; Fuzzy matrix equation; Numerical fuzzy solution; Trapezoidal fuzzy multiplication.

1. Introduction

Sylvester matrix equation plays a vital role in many fields due to its ability to solve real problems such as in control systems [1], medical imaging data acquisition, model reduction [2] and stochastic control, image processing and filtering [3]. Considering any uncertainty problems such as conflicting requirements during the system process, instability of environmental conditions, the distraction of any elements and noise, the classical matrix equation is sometimes ill-equipped, the crisp numbers need to be replaced by fuzzy numbers. When all the Sylvester matrix equation parameters are in the fuzzy form, it is called a fully fuzzy Sylvester matrix equation.

Definition 1. The matrix equation that can be written as

$$\tilde{A}\tilde{X} + \tilde{X}\tilde{B} = \tilde{C} \tag{1}$$

where, $\tilde{A} = (\tilde{a}_{ij})_{n\times n}$, $\tilde{B} = (\tilde{b}_{ij})_{m\times m}$, $\tilde{C} = \tilde{c}_{ij})_{n\times m}$ and $\tilde{X} = (\tilde{x}_{ij})_{n\times m}$ is called fully fuzzy Sylvester matrix equation (FFSME).

The FFSME in Eq. (1) can be written as follows,

$$\sum_{\substack{i,j=1 \\ k,l=1,\dots,4}}^{n} a_{ij}^{(k)} x_{ij}^{(l)} + \sum_{\substack{i,j=1 \\ k,l=1,\dots,4}}^{m} x_{ij}^{(l)} b_{ij}^{(k)} = c_{ij}^{(l)} \tag{2}$$

Triangular fully fuzzy Sylvester matrix equation (TFFSME) has been studied analytically by Shang, Guo and Bao [4], where the TFFSME is converted into a system of crisp linear equations by applying Dubois and Prade's arithmetic operator for multiplication [5]. However, the method was restricted only for positive fuzzy numbers and required a long multiplication process and consequently long computational timing. Malkawi, Ahmad, and Ibrahim [6] proposed a new associated linear system method for solving TFFSME, which is considered an extension of the method applied for solving fully fuzzy linear systems previously demonstrated in [7]. Indeed, this method required shorter computational timing than Shang's method; however, it is also restricted to positive TFFSME. In addition, both methods are limited to non-singular TFFSME. To overcome the shortcoming in these methods, Daud, Ahmad and Malkawi [8] obtain a positive solution for singular TFFSME by applying an associated linear matrix system approach where the solution is obtained by using the method of pseudoinverse. Recently, authors in [11] considered the solution of TrFFSME by transforming the TrFFSME to a system of crisp linear equations where the positive and negative fuzzy solutions are obtained by applying Vec-operator and Kronecker product method.

TFFSME with arbitrary coefficients has been studied by Daud et al., [9–12] using fuzzy Vec-operator and Kronecker products. However, these methods need further modifications as the Vec-operator and Kronecker product method is not applicable for arbitrary fuzzy systems with near-zero fuzzy numbers. It is worth mentioning that the properties of crisp numbers multiplication cannot be applied to fuzzy number multiplication, especially for near-zero fuzzy numbers. Therefore, the Vec-operator and Kronecker product approach is not applicable for arbitrary fuzzy systems with near-zero fuzzy numbers, which is proved in the following examples.

Example 1. Consider the following FFSME $\tilde{A}\tilde{X} + \tilde{X}\tilde{D} = \tilde{E}$
where
$$\tilde{A} = \begin{pmatrix} (1,2,3,4) & (1,2,5,7) \\ (1,2,3,5) & (4,5,6,7) \end{pmatrix}, \tilde{D} = \begin{pmatrix} (3,5,7,9) & (2,4,6,7) \\ (1,2,3,4) & (3,5,6,7) \end{pmatrix},$$
$$\tilde{X} = \begin{pmatrix} (1,4,5,8) & (1,3,5,7) \\ (1,2,4,5) & (4,5,7,8) \end{pmatrix} \text{ and } \tilde{E} = \begin{pmatrix} (6,38,85,167) & (10,47,110,189) \\ (12,38,88,152) & (31,64,123,182) \end{pmatrix}.$$
By applying fuzzy arithmetic operation on the given FFSME, the left-hand side is
$$\tilde{A}\tilde{X} + \tilde{X}\tilde{D} = \begin{pmatrix} (6,38,85,167) & (10,47,110,189) \\ (12,38,88,152) & (31,64,123,182) \end{pmatrix} \text{ which is equal to the right-hand side } (\tilde{E}).$$

Applying the Vec-operator and Kronecker product on the given FFSME gives,
$$\tilde{A}\tilde{X} + \tilde{X}\tilde{D} = \tilde{E} \Longrightarrow \left[(I \otimes \tilde{A}) + (\tilde{D}^T \otimes I) \right] Vec(\tilde{X}) = Vec(\tilde{E}).$$
$$\begin{pmatrix} (4,7,10,13) & (1,2,5,7) & (1,2,3,4) & (0,0,0,0) \\ (1,2,3,5) & (7,10,13,16) & (0,0,0,0) & (1,2,3,4) \\ (2,4,6,7) & (0,0,0,0) & (4,7,9,11) & (1,2,5,7) \\ (0,0,0,0) & (2,4,6,7) & (1,2,3,5) & (7,10,12,14) \end{pmatrix} \begin{pmatrix} (1,4,5,8) \\ (1,2,4,5) \\ (1,3,5,7) \\ (4,5,7,8) \end{pmatrix}$$
$$= \begin{pmatrix} (6,38,85,167) \\ (12,38,88,152) \\ (10,47,110,189) \\ (31,64,123,182) \end{pmatrix}.$$
The left-hand side of the given FFSME is,

$$[(I \otimes \tilde{A}) + (\tilde{D}^T \otimes I)]Vec(\tilde{X}) = \begin{pmatrix} (6, 38, 85, 167) \\ (12, 38, 88, 152) \\ (10, 47, 110, 189) \\ (31, 64, 123, 182) \end{pmatrix},$$

which is equal to the right-hand side $Vec(\tilde{E})$.

The following Example 2. shows that the Vec-operator and Kronecker product method cannot be applied to fuzzy systems with near-zero fuzzy numbers.

Example 2. Consider the following FFSME $\tilde{A}\tilde{X} + \tilde{X}\tilde{D} = \tilde{E}$, where

$$\tilde{A} = \begin{pmatrix} (-7, -4, 3, 4) & (-6, -3, 1, 4) \\ (-5, -4, -3, 1) & (-4, -2, 4, 7) \end{pmatrix},$$

$$\tilde{D} = \begin{pmatrix} (-3, -2, 3, 4) & (-3, -2, 3, 4) \\ (-7, -4, -3, 5) & (-2, -1, 4, 5) \end{pmatrix},$$

$$\tilde{X} = \begin{pmatrix} (-2, -1, 1, 3) & (-5, -2, 1, 4) \\ (-5, -4, 1, 5) & (-4, -1, 4, 7) \end{pmatrix} \text{ and}$$

$$\tilde{E} = \begin{pmatrix} (-88, -15, 27, 91) & (-104, -29, 19, 95) \\ (-119, -48, 24, 100) & (-88, -28, 48, 129) \end{pmatrix}.$$

In the given FFSME, the left-hand side is,

$$\tilde{A}\tilde{X} + \tilde{X}\tilde{B} = \begin{pmatrix} (-88, -15, 27, 91) & (-104, -29, 19, 95) \\ (-119, -48, 24, 100) & (-88, -28, 48, 129) \end{pmatrix}, \text{ which is equal to}$$

the right-hand side $(\tilde{E})$.

Applying the Vec-operator and Kronecker product on the given FFSME gives,

$$\begin{pmatrix} (-10, -6, 6, 8) & (-6, -3, 1, 4) & (-7, -4, -3, 5) & (0, 0, 0, 0) \\ (-5, -4, -3, -1) & (-7, -4, 7, 11) & (0, 0, 0, 0) & (-7, -4, -3, 5) \\ (-3, -2, 3, 4) & (0, 0, 0, 0) & (-9, -5, 7, 9) & (-6, -3, 1, 4) \\ (0, 0, 0, 0) & (-3, -2, 3, 4) & (-5, -4, -3, 1) & (-6, -3, 8, 12) \end{pmatrix} \begin{pmatrix} (-2, -1, 1, 3) \\ (-5, -4, 1, 5) \\ (-5, -2, 1, 4) \\ (-4, -1, 4, 7) \end{pmatrix}$$

$$= \begin{pmatrix} (-88, -14, 26, 89) \\ (-119, -48, 24, 100) \\ (-96, -29, 17, 85) \\ (-88, -28, 48, 129) \end{pmatrix}.$$

The left-hand side of the given FFSME is,

$$[(I \otimes \tilde{A}) + (\tilde{D}^T \otimes I)]Vec(\tilde{X}) = \begin{pmatrix} (-88, -14, 26, 89) \\ (-119, -48, 24, 100) \\ (-96, -29, 17, 85) \\ (-88, -28, 48, 129) \end{pmatrix} \neq \begin{pmatrix} (-88, -15, 27, 91) \\ (-119, -48, 24, 100) \\ (-104, -29, 19, 95) \\ (-88, -28, 48, 129) \end{pmatrix}.$$

Example 1. and Example 2. show that the Vec-operator and Kronecker product method has two main disadvantages:

I)   It cannot be applied to fuzzy systems with near-zero fuzzy numbers.

II)   It can be applied only to fuzzy systems with positive or negative fuzzy numbers; however, the Vec-operator and Kronecker product method for $m \times n$ positive or negative fuzzy system required getting the inverse of $mn \times mn$ matrices, which is not possible for large systems.

A study was conducted by [13] on the TFFSME in the form $\tilde{A}\tilde{X} - \tilde{X}\tilde{B} = \tilde{C}$, which used the $\alpha - cuts$ expansion approach in the parameters. This method has an advantage because it provides maximal and minimal symmetric solutions of the TFFSME. However, the method required long fuzzy operations in obtaining the solution. Similarly, authors in [15] propose an algorithm for obtaining the positive solution of TFFSME with arbitrary coefficients. However, the method was restricted only to positive fuzzy solutions.

Most of the analytical methods proposed for solving TFFSME and TrFFSME in the literature are based on Dubois and Prade's arithmetic operator for multiplication which is restricted only for positive fuzzy numbers with very small fuzziness [16]. Therefore, most analytical methods are limited to

positive coefficients and positive fuzzy solutions only. In addition, many researchers have applied Kaufmann and Gupta's arithmetic multiplication operator for solving TFFSME with arbitrary coefficients; however, their methods are limited to positive fuzzy solutions only. Furthermore, most of the existing studies restricted the size of the system due to the nature of analytical methods used and arithmetic operations applied, where many methods are limited only to small-sized systems. Therefore, to deal with this shortcoming, this paper presents a new numerical method for solving TrFFSME with arbitrary TrFNs where the TrFFSME is transferred to a non-linear system based on new arithmetic fuzzy multiplication for TrFNs. With the assumption that the exact solution is not given and there is no initial value is known, the solution to the non-linear system can be obtained by a new two-stage algorithm where the first stage algorithm reduces the search area for the fuzzy solution and the second stage algorithm find it.

The proposed method is applicable for solving large size TrFFSME. In addition, it can also be applied to TFFSME and fully fuzzy matrix equation (FFME) with both TFNs and TrFNs. This paper is organized as follows: Section 2 introduces preliminary arithmetic operations of intervals and $\alpha - cut$ intervals. In section 3, new arithmetic operations for TrFNs are developed. In section 4, a proposed numerical method for solving TrFFSME is applied to a 2×2 TrFFSME along with a presentation of its algorithm. In section 5, a numerical example is presented to illustrate the proposed method. Section 6 is dedicated to the conclusion.

2. Preliminaries

This section introduces basic arithmetic operations of fuzzy numbers [17].

Definition 2. Interval arithmetic operations.

If $A = [a_1, a_2]$, $B = [b_1, b_2]$, then $\forall\, a_1, a_2, b_1, b_2 \in R$, we have,

(I)   Addition

$$A + B = [a_1 + b_1, a_2 + b_2]. \tag{3}$$

(II)   Subtraction

$$A - B = [a_1 - b_2, a_2 - b_1].$$

(III)   Multiplication

Case I) If A and B are arbitrary real numbers then:

$$A \cdot B = [min(a_1 b_1, a_1 b_2, a_2 b_1, a_2 b_2), max(a_1 b_1, a_1 b_2, a_2 b_1, a_2 b_2)] \tag{4}$$

Case II) If $A > 0$ and $B > 0$ then:

$$A \cdot B = [a_1 b_1, a_2 b_2] \tag{5}$$

Case III) If $A < 0$ and $B < 0$ then:

$$A \cdot B = [a_2 b_2, a_1 b_1] \tag{6}$$

Case IV) If $A > 0$ and $B < 0$ then:

$$A \cdot B = [a_2 b_1, a_1 b_2] \tag{7}$$

Case V) If $A < 0$ and $B > 0$ then:

$$A \cdot B = [a_1 b_2, a_2 b_1] \tag{8}$$

(IV)   Division

$A/B = [a_1, a_2]/[b_1, b_2]$
$= [min(a_1/b_1, a_1/b_2, a_2/b_1, a_2/b_2), max(a_1/b_1, a_1/b_2, a_2/b_1, a_2/b_2)]$
where $b_1, b_2 \neq 0$.

(V)   Inverse interval

$A^{-1} = [a_1, a_2]^{-1} = [min(1/a_1, 1/a_2), max(1/a_1, 1/a_2)]$, where $a_1, a_2 \neq 0$.

(VI)   Equality: Two intervals $A = [a_1, a_2]$ and $B = [b_1, b_2]$ are equal, if and only if

$$a_1 = b_1, \ a_2 = b_2. \tag{9}$$

(VII)   Scalar multiplication: Let $\lambda \in R$ then,

$$\lambda A = \lambda[a_1, a_2] = [min\,(\lambda a_1, \lambda a_2), max\,(\lambda a_1, \lambda a_2].$$

Definition 3. Operations of $\alpha - cut$ interval.

We referred to the $\alpha - cut$ interval of fuzzy numbers $A = [a_1, a_2]$ and $B = [b_1, b_2]$, as crisp set
$A_\alpha = [a_1^\alpha, a_2^\alpha]$, $B_\alpha = [b_1^\alpha, b_2^\alpha]$ respectively, $\forall \alpha \in [0,1]$, $a_1, a_2, b_1, b_2, a_1^\alpha, a_2^\alpha, b_1^\alpha, b_2^\alpha \in R$. So, A_α, B_α are crisp intervals. As a result, the operations of interval reviewed in Definition 2 can be applied to the $\alpha - cut$ interval A_α and B_α. Operations between A_α and B_α can be represented as follow:

(I)   Addition

$$A_\alpha + B_\alpha = [a_1^\alpha, a_2^\alpha] + [b_1^\alpha, b_2^\alpha] = [a_1^\alpha + b_1^\alpha, a_2^\alpha + b_2^\alpha]. \qquad (10)$$

(II)   Subtraction

$$A_\alpha - B_\alpha = [a_1^\alpha, a_2^\alpha] - [b_1^\alpha, b_2^\alpha] = [a_1^\alpha - b_2^\alpha, a_2^\alpha - b_1^\alpha].$$

(III)   Multiplication

$$A_\alpha \cdot B_\alpha = [a_1^\alpha, a_2^\alpha] \cdot [b_1^\alpha, b_2^\alpha].$$
$$= [min\,(a_1^\alpha b_1^\alpha, a_1^\alpha b_2^\alpha, a_2^\alpha b_1^\alpha, a_2^\alpha b_2^\alpha), max(a_1^\alpha b_1^\alpha, a_1^\alpha b_2^\alpha, a_2^\alpha b_1^\alpha, a_2^\alpha b_2^\alpha)] \qquad (11)$$

(IV)   Equality

Two intervals $A_\alpha = [a_1^\alpha, a_2^\alpha]$, and $B_\alpha = [b_1^\alpha, b_2^\alpha]$ are equal, if and only if $a_1^\alpha = b_1^\alpha$ and $a_2^\alpha = b_2^\alpha$.

The following are basic definitions and results related to TrFNs [5,18].

Definition 4. Let X be a universal set. Then, we define the fuzzy subset $\tilde{A}$ of X by its membership function $\mu_{\tilde{A}} : X \to [0,1]$ which assigns to each element $x \in X$ a real number $\mu_{\tilde{A}}(x)$ in the interval $[0,1]$, where the function value of $\mu_{\tilde{A}}(x)$ represents the grade of membership of x in $\tilde{A}$. A fuzzy set $\tilde{A}$ is written as $\tilde{A} = \{(x, \mu_{\tilde{A}}(x)), x \in X, \mu_{\tilde{A}}(x) \in [0,1]\}$.

Definition 5. A fuzzy set $\tilde{A}$, defined on the universal set of real number R; is said to be a fuzzy number if its membership function has the following characteristics:

1. $\tilde{A}$ is convex, i.e.,

$$\mu_{\tilde{A}}(\lambda x_1 + (1-\lambda)x_2 \geq min\big(\mu_{\tilde{A}}(x), \mu_{\tilde{A}}(x)\big) \,\forall\, x_1, x_2 \in R\ , \forall \lambda \in [0,1].$$

2. $\tilde{A}$ is normal, i.e.,

$\exists\, x_0 \in R$ such that $\mu_{\tilde{A}}(x_0) = 1$.

3. $\mu_{\tilde{A}}$ is piecewise continuous.

Definition 6. A fuzzy number $\tilde{A} = (a_1, a_2, a_3, a_4)$ is a TrFN if its membership function is:

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & x < a_1 \\ \dfrac{x - a_1}{a_2 - a_1} & a_1 \leq x \leq a_2 \\ 1 & a_2 \leq x \leq a_3 \\ \dfrac{a_4 - x}{a_4 - a_3} & a_3 \leq x \leq a_4 \\ 0 & x > a_4 \end{cases}$$

The following Figure 1 Represents the TrFN in the form (a_1, a_2, a_3, a_4).
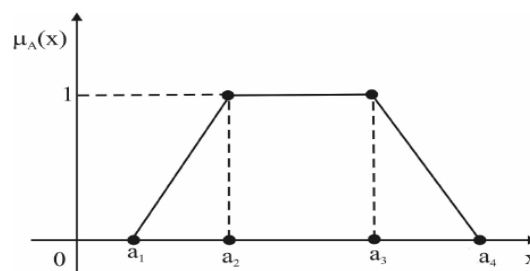
Figure 1. Representation of TrFN (a_1, a_2, a_3, a_4).

Definition 7. The sign of the TrFN $\tilde{A} = (a_1, a_2, a_3, a_4)$. can be classified as:

- $\tilde{A}$ is positive (negative) iff $a_1 \geq 0, (a_4 \leq 0)$.
- $\tilde{A}$ is zero iff $(a_1, a_2, a_3 \ and \ a_4 = 0)$.
- $\tilde{A}$ is near zero iff $a_1 \leq 0 \leq a_4$.

Definition 8. Operation of TrFNs.

The arithmetic operations of TrFNs is presented as follows, let

$\tilde{A} = (a_1, a_2, a_3, a_4)$,

$\tilde{B} = (b_1, b_2, b_3, b_4)$ be two TrFNs then:

1.   Subtraction

$\tilde{A} - \tilde{B} = (a_1, a_2, a_3, a_4) - (b_1, b_2, b_3, b_4) = (a_1 - b_4, a_2 - b_3, a_3 - b_2, a_4 - b_1)$.

2.   Addition

$\tilde{A} + \tilde{B} = (a_1, a_2, a_3, a_4) + (b_1, b_2, b_3, b_4) = (a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4)$.

3.   Symmetric image

$$-\tilde{A} = (-a_4, -a_3, -a_2, -a_1),$$

4.   Scalar multiplication: Let $\lambda \in \mathbb{R}$ then,

$$\lambda \otimes (a_1, a_2, a_3, a_4) = \begin{cases} (\lambda a_1, \lambda a_2, \lambda a_3, \lambda a_4) & \lambda \geq 0 \\ (\lambda a_4, \lambda a_3, \lambda a_2, \lambda a_1) & \lambda < 0 \end{cases}$$

5.   Equality: The fuzzy numbers $\tilde{A} = (a_1, a_2, a_3, a_4)$ and

$\tilde{B} = (b_1, b_2, b_3, b_4)$ are equal iff

$$a_1 = b_1, \ a_2 = b_2, a_3 = b_3 \ and \ a_4 = b_4 \tag{12}$$

3. Trapezoidal fuzzy numbers multiplication

In this section, we develop new arithmetic multiplication operations between TrFNs. In the following proposition, we first find $\alpha - cut$ intervals for TrFNs.

Proposition 1. *An $\alpha - cut$ interval for $\tilde{A} = (a_1, a_2, a_3, a_4)$ can be written as:*

$$\tilde{A}_\alpha = [a_1^\alpha, a_2^\alpha] = [(a_2 - a_1)\alpha + a_1, -(a_4 - a_3)\alpha + a_4], \ \forall \alpha \in [0,1]. \tag{13}$$

Proof.

By the definition of membership function for TrFN $\tilde{A}$ Definition 6 and if we let,

$$\frac{a_1^\alpha - a_1}{a_2 - a_1} = \alpha \ and \ \frac{a_4 - a_2^\alpha}{a_4 - a_3} = \alpha$$

Solving for a_1^α and a_2^α using cross multiplication property of equality, we get:

$$a_1^\alpha = (a_2 - a_1)\alpha + a_1$$
$$a_2^\alpha = -(a_4 - a_3)\alpha + a_4$$

Thus,

$$\tilde{A}_\alpha = [a_1^\alpha, a_2^\alpha] = [(a_2 - a_1)\alpha + a_1, -(a_4 - a_3)\alpha + a_4], \forall \alpha \in [0,1].$$

$\square$

The following propositions discuss new arithmetic multiplication operations between TrFNs, namely Ahmd arithmetic multiplication operators.

Proposition 2. If $\tilde{A} = (a_1, a_2, a_3, a_4)$, $\tilde{B} = (b_1, b_2, b_3, b_4)$ be two arbitrary TrFNs respectively, then:

$$\tilde{A}\tilde{B} = (a, h, m, d). \tag{14}$$

where

$$a = min(a_1 b_1, a_1 b_4, a_4 b_1, a_4 b_4),$$
$$h = min(a_2 b_2, a_2 b_3, a_3 b_2, a_3 b_3),$$
$$m = max(a_2 b_2, a_2 b_3, a_3 b_2, a_3 b_3),$$
$$d = max(a_1 b_1, a_1 b_4, a_4 b_1, a_4 b_4).$$

Proof. Based on Proposition 1, the $\alpha - cut$ intervals for $\tilde{A} = (a_1, a_2, a_3, a_4)$ and $\tilde{B} = (b_1, b_2, b_3, b_4)$ are

$$\tilde{A}_\alpha = [a_1^\alpha, a_2^\alpha] = [(a_2 - a_1)\alpha + a_1, -(a_4 - a_3)\alpha + a_4],$$
$$\tilde{B}_\alpha = [b_1^\alpha, b_2^\alpha] = [(b_2 - b_1)\alpha + b_1, -(b_4 - b_3)\alpha + b_4],$$

$\forall \alpha \in [0,1]$ respectively.

By applying the multiplication operations of $\alpha - $ cut interval in Definition 3 in Eq. (11) on $\tilde{A}_\alpha$ and $\tilde{B}_\alpha$ we get:

$$\tilde{A}_\alpha \times \tilde{B}_\alpha = [a_1^\alpha, a_2^\alpha] \times [b_1^\alpha, b_2^\alpha]$$
$$=$$
$$[min\,(a_1^\alpha b_1^\alpha, a_1^\alpha b_2^\alpha, a_2^\alpha b_1^\alpha, a_2^\alpha b_2^\alpha), max(a_1^\alpha b_1^\alpha, a_1^\alpha b_2^\alpha, a_2^\alpha b_1^\alpha, a_2^\alpha b_2^\alpha)]$$
$$= [e_1, e_2].$$

where

$e_1 = min\big(((a_2 - a_1)\alpha + a_1) \cdot ((b_2 - b_1)\alpha + b_1), ((a_2 - a_1)\alpha + a_1) \cdot (-(b_4 - b_3)\alpha + b_4), (-(a_4 - a_3)\alpha + a_4) \cdot ((b_2 - b_1)\alpha + b_1), (-(a_4 - a_3)\alpha + a_4) \cdot (-(b_4 - b_3)\alpha + b_4)\big),$

$e_2 = max\big(((a_2 - a_1)\alpha + a_1) \cdot ((b_2 - b_1)\alpha + b_1), ((a_2 - a_1)\alpha + a_1) \cdot (-(b_4 - b_3)\alpha + b_4), (-(a_4 - a_3)\alpha + a_4) \cdot ((b_2 - b_1)\alpha + b_1), (-(a_4 - a_3)\alpha + a_4) \cdot (-(b_4 - b_3)\alpha + b_4)\big),$

$a_1^\alpha = (a_2 - a_1)\alpha + a_1,$
$a_2^\alpha = -(a_4 - a_3)\alpha + a_4,$
$b_1^\alpha = (b_2 - b_1)\alpha + b_1,$
$b_2^\alpha = -(b_4 - b_3)\alpha + b_4.$

Since the product of two TrFNs is TrFN, the left and right endpoints of the TrFN $\tilde{A}\tilde{B}$ can be found if we let $\alpha = 0$. Thus, at $\alpha = 0$ the following is obtained,

$$A_0 \times B_0 = [e_1, e_2] = [min(a_1 b_1, a_1 b_4, a_4 b_1, a_4 b_4), max(a_1 b_1, a_1 b_4, a_4 b_1, a_4 b_4)].$$

The following Figure 2 represents the product $A_0 \times B_0$ at $\alpha = 0$.
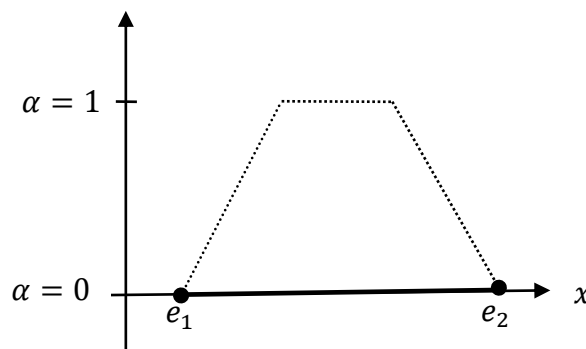
Figure 2. The product $A_0 \times B_0$ at $\alpha = 0$.

While the mean points of the TrFN $\tilde{A}\tilde{B}$ can be found if we let $\alpha = 1$. Thus, at $\alpha = 1$ the following is obtained,

$$A_1 \times B_1 = [e_1, e_2] = [min(a_2 b_2, a_2 b_3, a_3 b_2, a_3 b_3), max(a_2 b_2, a_2 b_3, a_3 b_2, a_3 b_3)].$$

The following Figure 3 represents the product $A_1 \times B_1$ at $\alpha = 1$.
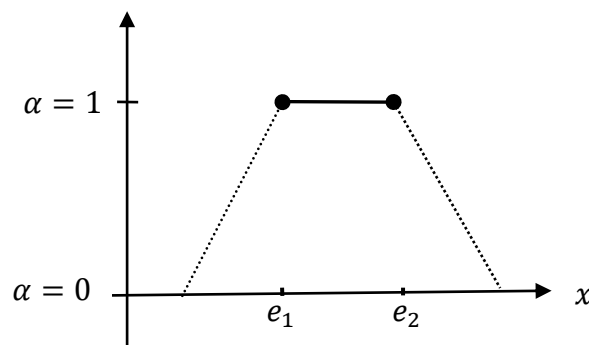
Figure 3. The product $A_1 \times B_1$ at $\alpha = 0$.

By combining the endpoints and mean points of $\tilde{A}\tilde{B}$ using the definition of TrFNs in Definition 6, the product $\tilde{A}\tilde{B}$ is

$$\tilde{A}\tilde{B} = (a, h, m, d).$$

where,

$$a = min(a_1 b_1, a_1 b_4, a_4 b_1, a_4 b_4),$$
$$h = min(a_2 b_2, a_2 b_3, a_3 b_2, a_3 b_3),$$
$$m = max(a_2 b_2, a_2 b_3, a_3 b_2, a_3 b_3)$$
$$d = max(a_1 b_1, a_1 b_4, a_4 b_1, a_4 b_4).$$

□

The following Figure 4 represents the product $\tilde{A}\tilde{B}$
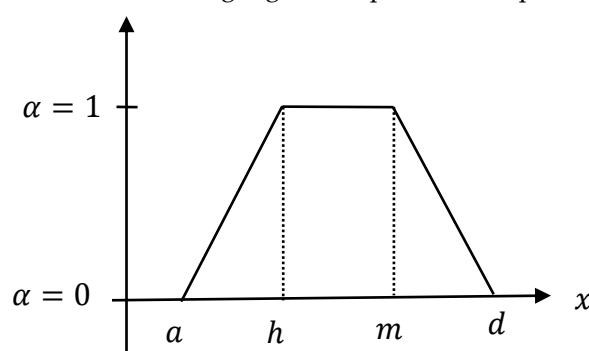
Figure 4. The product $\tilde{A}\tilde{B}$

Definition 9.

If $\tilde{A}$ and $\tilde{B}$ are two arbitrary TrFNs respectively, then the multiplication $\tilde{A}\tilde{B} = (a, h, m, d)$ is called Ahmd Arithmetic Multiplication Operator (AMO) for Arbitrary TrFNs

The implementation of AMO is illustrated in the following example.

Example 3. Let $\tilde{A} = (-4, -2, 1, 3)$ and $\tilde{B} = (-5, 2, 4, 7)$ be two arbitrary TrFNs respectively, then

$$a = min(-4 \times -5, -4 \times 7, 3 \times -5, 3 \times 7) = -28$$
$$h = min(-2 \times 2, -2 \times 4, 1 \times 2, 1 \times 4) = -8$$
$$m = max(-2 \times 2, -2 \times 4, 1 \times 2, 1 \times 4) = 4$$
$$d = max(-4 \times -5, -4 \times 7, 3 \times -5, 3 \times 7) = 21$$

Thus, $\tilde{A}\tilde{B} = (-28, -8, 4, 21)$.

Corollary 1. Positive TrFNs arithmetic multiplication operation.

If $\tilde{A} = (a_1, a_2, a_3, a_4)$, $\tilde{B} = (b_1, b_2, b_3, b_4)$ be two positive TrFNs then:

$$\tilde{A}\tilde{B} = (a_1 b_1, a_2 b_2, a_3 b_3, a_4 b_4).$$

Proof.

From Proposition 2 and by Eq. (14), we have:

$$\tilde{A}\tilde{B} = (a, h, m, d).$$

And using Definition 2 by Eq. (5) $\tilde{A}\tilde{B}$ can be reduced as follows:

$$a = min(a_1 b_1, a_1 b_4, a_4 b_1, a_4 b_4) = a_1 b_1,$$
$$h = min(a_2 b_2, a_2 b_3, a_3 b_2, a_3 b_3) = a_2 b_2,$$
$$m = max(a_2 b_2, a_2 b_3, a_3 b_2, a_3 b_3) = a_3 b_3$$
$$d = max(a_1 b_1, a_1 b_4, a_4 b_1, a_4 b_4) = a_4 b_4.$$

Thus,

$$\tilde{A}\tilde{B} = (a_1 b_1, a_2 b_2, a_3 b_3, a_4 b_4).$$ □

Corollary 2. Negative TrFNs arithmetic multiplication operation.

If $\tilde{A} = (a_1, a_2, a_3, a_4)$, $\tilde{B} = (b_1, b_2, b_3, b_4)$ be two negative TrFNs then:

$$\tilde{A}\tilde{B} = (a_4 b_4, a_3 b_3, a_2 b_2, a_1 b_1).$$

Proof.

From Proposition 2 and by Eq. (14), we have:

$$\tilde{A}\tilde{B} = (a, h, m, d).$$

And using Definition 2 by Eq. (6) $\tilde{A}\tilde{B}$ can be reduced as follows:

$$a = min(a_1b_1, a_1b_4, a_4b_1, a_4b_4) = a_4b_4,$$
$$h = min(a_2b_2, a_2b_3, a_3b_2, a_3b_3) = a_3b_3,$$
$$m = max(a_2b_2, a_2b_3, a_3b_2, a_3b_3) = a_2b_2$$
$$d = max(a_1b_1, a_1b_4, a_4b_1, a_4b_4) = a_1b_1.$$

Thus,

$$\tilde{A}\tilde{B} = (a_1b_1, a_2b_2, a_3b_3, a_4b_4).$$

□

Corollary 3. Positive and negative TrFNs arithmetic multiplication operation.

If $\tilde{A} = (a_1, a_2, a_3, a_4) > 0$, $\tilde{B} = (b_1, b_2, b_3, b_4) < 0$ *be two TrFNs then:*

$$\tilde{A}\tilde{B} = (a_4b_1, a_3b_2, a_2b_3, a_1b_4).$$

Proof.

From Proposition 2 and by Eq. (14), we have:

$$\tilde{A}\tilde{B} = (a, h, m, d).$$

And using Definition 2 by Eq. (7) $\tilde{A}\tilde{B}$ can be reduced as follows:

$$a = min(a_1b_1, a_1b_4, a_4b_1, a_4b_4) = a_4b_1,$$
$$h = min(a_2b_2, a_2b_3, a_3b_2, a_3b_3) = a_3b_2,$$
$$m = max(a_2b_2, a_2b_3, a_3b_2, a_3b_3) = a_2b_3$$
$$d = max(a_1b_1, a_1b_4, a_4b_1, a_4b_4) = a_1b_4.$$

Thus,

$$\tilde{A}\tilde{B} = (a_4b_1, a_3b_2, a_2b_3, a_1b_4).$$

□

Corollary 4. Negative and Positive TrFNs arithmetic multiplication operation.

If $\tilde{A} = (a_1, a_2, a_3, a_4) < 0$, $\tilde{B} = (b_1, b_2, b_3, b_4) > 0$ *be two TrFNs then:*

$$\tilde{A}\tilde{B} = (a_1b_4, a_2b_3, a_3b_2, a_4b_1).$$

Proof.

From Proposition 2 and by Eq. (14), we have:

$$\tilde{A}\tilde{B} = (a, h, m, d).$$

And using Definition 2 by Eq. (8) $\tilde{A}\tilde{B}$ can be reduced as follows:

$$a = min(a_1b_1, a_1b_4, a_4b_1, a_4b_4) = a_1b_4,$$
$$h = min(a_2b_2, a_2b_3, a_3b_2, a_3b_3) = a_2b_3,$$
$$m = max(a_2b_2, a_2b_3, a_3b_2, a_3b_3) = a_3b_2$$
$$d = ma\,x(a_1b_1, a_1b_4, a_4b_1, a_4b_4) = a_4b_1.$$

Thus,

$$\tilde{A}\tilde{B} = (a_1b_4, a_2b_3, a_3b_2, a_4b_1).$$

□

The following section proposes a new method for solving arbitrary TrFFSME based on the arithmetic multiplication operation proposed in Proposition 2.

4. Proposed method

This section proposes a new method for solving arbitrary TrFFSME based on the arithmetic multiplication operation proposed in Proposition 2. In the following theorem, the FFSME Eq. (1) extended into systems matrix equations.

Theorem 1

If $\tilde{A}, \tilde{B}$ and $\tilde{X}$ are arbitrary TrFNs, then the FFSME Eq. (1) is equivalent to:

$$\begin{cases} \sum_{i,j=1}^{n} min\left(a_{ij}^{(1)}x_{ij}^{(1)}, a_{ij}^{(1)}x_{ij}^{(4)}, a_{ij}^{(4)}x_{ij}^{(1)}, a_{ij}^{(4)}x_{ij}^{(4)}\right) + \sum_{i,j=1}^{m} min\left(x_{ij}^{(1)}b_{ij}^{(1)}, x_{ij}^{(1)}b_{ij}^{(4)}, x_{ij}^{(4)}b_{ij}^{(1)}, x_{ij}^{(4)}b_{ij}^{(4)}\right) = c_{ij}^{(1)} \\ \sum_{i,j=1}^{n} min\left(a_{ij}^{(2)}x_{ij}^{(2)}, a_{ij}^{(2)}x_{ij}^{(3)}, a_{ij}^{(3)}x_{ij}^{(2)}, a_{ij}^{(3)}x_{ij}^{(3)}\right) + \sum_{i,j=1}^{m} min\left(x_{ij}^{(2)}b_{ij}^{(2)}, x_{ij}^{(2)}b_{ij}^{(3)}, x_{ij}^{(3)}b_{ij}^{(2)}, x_{ij}^{(3)}b_{ij}^{(3)}\right) = c_{ij}^{(2)} \\ \sum_{i,j=1}^{n} max\left(a_{ij}^{(2)}x_{ij}^{(2)}, a_{ij}^{(2)}x_{ij}^{(3)}, a_{ij}^{(3)}x_{ij}^{(2)}, a_{ij}^{(3)}x_{ij}^{(3)}\right) + \sum_{i,j=1}^{m} max\left(x_{ij}^{(2)}b_{ij}^{(2)}, x_{ij}^{(2)}b_{ij}^{(3)}, x_{ij}^{(3)}b_{ij}^{(2)}, x_{ij}^{(3)}b_{ij}^{(3)}\right) = c_{ij}^{(3)} \\ \sum_{i,j=1}^{n} max\left(a_{ij}^{(1)}x_{ij}^{(1)}, a_{ij}^{(1)}x_{ij}^{(4)}, a_{ij}^{(4)}x_{ij}^{(1)}, a_{ij}^{(4)}x_{ij}^{(4)}\right) + \sum_{i,j=1}^{m} max\left(x_{ij}^{(1)}b_{ij}^{(1)}, x_{ij}^{(1)}b_{ij}^{(4)}, x_{ij}^{(4)}b_{ij}^{(1)}, x_{ij}^{(4)}b_{ij}^{(4)}\right) = c_{ij}^{(4)} \end{cases} \quad (15)$$

Proof. Assuming

$\tilde{A} = (\tilde{a}_{ij})_{n \times n} = \left(a_{ij}^{(1)}, a_{ij}^{(2)}, a_{ij}^{(3)}, a_{ij}^{(4)}\right)$, $\tilde{B} = (\tilde{b}_{ij})_{m \times m} = (\tilde{b}_{ij}) = \left(b_{ij}^{(1)}, b_{ij}^{(2)}, b_{ij}^{(3)}, b_{ij}^{(4)}\right)$,
$\tilde{X} = (\tilde{x}_{ij})_{n \times m} = \left(x_{ij}^{(1)}, x_{ij}^{(2)}, x_{ij}^{(3)}, x_{ij}^{(4)}\right)$ and $\tilde{C} = (\tilde{c}_{ij})_{n \times m} = \left(c_{ij}^{(1)}, c_{ij}^{(2)}, c_{ij}^{(3)}, c_{ij}^{(4)}\right)$ arbitrary TrFNs.

We have from Proposition 2 by Eq. (14),

$$\tilde{A}\tilde{X} = (M, N, P, Q)$$

where,

$$M = min\left(a_{ij}^{(1)}x_{ij}^{(1)}, a_{ij}^{(1)}x_{ij}^{(4)}, a_{ij}^{(4)}x_{ij}^{(1)}, a_{ij}^{(4)}x_{ij}^{(4)}\right),$$
$$N = min\left(a_{ij}^{(2)}x_{ij}^{(2)}, a_{ij}^{(2)}x_{ij}^{(3)}, a_{ij}^{(3)}x_{ij}^{(2)}, a_{ij}^{(3)}x_{ij}^{(3)}\right),$$
$$P = max\left(a_{ij}^{(2)}x_{ij}^{(2)}, a_{ij}^{(2)}x_{ij}^{(3)}, a_{ij}^{(3)}x_{ij}^{(2)}, a_{ij}^{(3)}x_{ij}^{(3)}\right),$$
$$Q = max\left(a_{ij}^{(1)}x_{ij}^{(1)}, a_{ij}^{(1)}x_{ij}^{(4)}, a_{ij}^{(4)}x_{ij}^{(1)}, a_{ij}^{(4)}x_{ij}^{(4)}\right).$$

And,

$$\tilde{X}\tilde{B} = (K, L, H, R)$$

where,

$$K = min\left(x_{ij}^{(1)}b_{ij}^{(1)}, x_{ij}^{(1)}b_{ij}^{(4)}, x_{ij}^{(4)}b_{ij}^{(1)}, x_{ij}^{(4)}b_{ij}^{(4)}\right),$$
$$L = min\left(x_{ij}^{(2)}b_{ij}^{(2)}, x_{ij}^{(2)}b_{ij}^{(3)}, x_{ij}^{(3)}b_{ij}^{(2)}, x_{ij}^{(3)}b_{ij}^{(3)}\right),$$
$$H = max\left(x_{ij}^{(2)}b_{ij}^{(2)}, x_{ij}^{(2)}b_{ij}^{(3)}, x_{ij}^{(3)}b_{ij}^{(2)}, x_{ij}^{(3)}b_{ij}^{(3)}\right),$$
$$R = max\left(x_{ij}^{(1)}b_{ij}^{(1)}, x_{ij}^{(1)}b_{ij}^{(4)}, x_{ij}^{(4)}b_{ij}^{(1)}, x_{ij}^{(4)}b_{ij}^{(4)}\right).$$

Using Definition 2 by Eq. (2), the arbitrary TrFFSME $\tilde{A}\tilde{X} + \tilde{B}\tilde{X} = \tilde{C}$ can be written as:

$$\begin{cases} \sum_{i,j=1}^{n} min\left(a_{ij}^{(1)}x_{ij}^{(1)}, a_{ij}^{(1)}x_{ij}^{(4)}, a_{ij}^{(4)}x_{ij}^{(1)}, a_{ij}^{(4)}x_{ij}^{(4)}\right) + \sum_{i,j=1}^{m} min\left(x_{ij}^{(1)}b_{ij}^{(1)}, x_{ij}^{(1)}b_{ij}^{(4)}, x_{ij}^{(4)}b_{ij}^{(1)}, x_{ij}^{(4)}b_{ij}^{(4)}\right) = c_{ij}^{(1)} \\ \sum_{i,j=1}^{n} min\left(a_{ij}^{(2)}x_{ij}^{(2)}, a_{ij}^{(2)}x_{ij}^{(3)}, a_{ij}^{(3)}x_{ij}^{(2)}, a_{ij}^{(3)}x_{ij}^{(3)}\right) + \sum_{i,j=1}^{m} min\left(x_{ij}^{(2)}b_{ij}^{(2)}, x_{ij}^{(2)}b_{ij}^{(3)}, x_{ij}^{(3)}b_{ij}^{(2)}, x_{ij}^{(3)}b_{ij}^{(3)}\right) = c_{ij}^{(2)} \\ \sum_{i,j=1}^{n} max\left(a_{ij}^{(2)}x_{ij}^{(2)}, a_{ij}^{(2)}x_{ij}^{(3)}, a_{ij}^{(3)}x_{ij}^{(2)}, a_{ij}^{(3)}x_{ij}^{(3)}\right) + \sum_{i,j=1}^{m} max\left(x_{ij}^{(2)}b_{ij}^{(2)}, x_{ij}^{(2)}b_{ij}^{(3)}, x_{ij}^{(3)}b_{ij}^{(2)}, x_{ij}^{(3)}b_{ij}^{(3)}\right) = c_{ij}^{(3)} \\ \sum_{i,j=1}^{n} max\left(a_{ij}^{(1)}x_{ij}^{(1)}, a_{ij}^{(1)}x_{ij}^{(4)}, a_{ij}^{(4)}x_{ij}^{(1)}, a_{ij}^{(4)}x_{ij}^{(4)}\right) + \sum_{i,j=1}^{m} max\left(x_{ij}^{(1)}b_{ij}^{(1)}, x_{ij}^{(1)}b_{ij}^{(4)}, x_{ij}^{(4)}b_{ij}^{(1)}, x_{ij}^{(4)}b_{ij}^{(4)}\right) = c_{ij}^{(4)} \end{cases}$$

In the following subsection, the proposed method is applied to 2×2 arbitrary TrFFSME in Eq. (1) and followed by developing a numerical algorithm to obtain the fuzzy solution(s).

4.1 Proposed method for 2×2 arbitrary TrFFSME

In this section, we apply the proposed method for solving 2×2 arbitrary TrFFSME: Since, $\tilde{A} = (\tilde{a}_{ij})_{n \times n}$ where, $(\tilde{a}_{ij}) = (a_{ij}^{(1)}, a_{ij}^{(2)}, a_{ij}^{(3)}, a_{ij}^{(4)})$, $\forall\ 1 \leq ij \leq 2$. Thus, without loss of generality, we will assume $\tilde{A}$ to be 2×2. Then $\tilde{A}$ can be written as:

$$A = \begin{pmatrix} (a_{11}^{(1)}, a_{11}^{(2)}, a_{11}^{(3)}, a_{11}^{(4)}) & (a_{12}^{(1)}, a_{12}^{(2)}, a_{12}^{(3)}, a_{12}^{(4)}) \\ (a_{21}^{(1)}, a_{21}^{(2)}, a_{21}^{(3)}, a_{21}^{(4)}) & (a_{22}^{(1)}, a_{22}^{(2)}, a_{22}^{(3)}, a_{22}^{(4)}) \end{pmatrix}$$

Similarly,

$$\tilde{B} = \begin{pmatrix} (b_{11}^{(1)}, b_{11}^{(2)}, b_{11}^{(3)}, b_{11}^{(4)}) & (b_{12}^{(1)}, b_{12}^{(2)}, b_{12}^{(3)}, b_{12}^{(4)}) \\ (b_{21}^{(1)}, b_{21}^{(2)}, b_{21}^{(3)}, b_{21}^{(4)}) & (b_{22}^{(1)}, b_{22}^{(2)}, b_{22}^{(3)}, b_{22}^{(4)}) \end{pmatrix},$$

$$\tilde{X} = \begin{pmatrix} \tilde{x}_{11} & \tilde{x}_{12} \\ \tilde{x}_{21} & \tilde{x}_{22} \end{pmatrix} = \begin{pmatrix} (x_{11}^{(1)}, x_{11}^{(2)}, x_{11}^{(3)}, x_{11}^{(4)}) & (x_{12}^{(1)}, x_{12}^{(2)}, x_{12}^{(3)}, x_{12}^{(4)}) \\ (x_{21}^{(1)}, x_{21}^{(2)}, x_{21}^{(3)}, x_{21}^{(4)}) & (x_{22}^{(1)}, x_{22}^{(2)}, x_{22}^{(3)}, x_{22}^{(4)}) \end{pmatrix}$$

and,

$$\tilde{C} = \begin{pmatrix} (c_{11}^{(1)}, c_{11}^{(2)}, c_{11}^{(3)}, c_{11}^{(4)}) & (c_{12}^{(1)}, c_{12}^{(2)}, c_{12}^{(3)}, c_{12}^{(4)}) \\ (c_{21}^{(1)}, c_{21}^{(2)}, c_{21}^{(3)}, c_{21}^{(4)}) & (c_{22}^{(1)}, c_{22}^{(2)}, c_{22}^{(3)}, c_{22}^{(4)}) \end{pmatrix}.$$

The steps of the proposed method are as follows:

Step 1. Multiplying $\tilde{A}\tilde{X}$ using the proposed multiplication operator Proposition 2, Eq. (14):

$$\tilde{A}\tilde{X} = \begin{pmatrix} (a_{11}^{(1)}, a_{11}^{(2)}, a_{11}^{(3)}, a_{11}^{(4)}) & (a_{12}^{(1)}, a_{12}^{(2)}, a_{12}^{(3)}, a_{12}^{(4)}) \\ (a_{21}^{(1)}, a_{21}^{(2)}, a_{21}^{(3)}, a_{21}^{(4)}) & (a_{22}^{(1)}, a_{22}^{(2)}, a_{22}^{(3)}, a_{22}^{(4)}) \end{pmatrix} \begin{pmatrix} (x_{11}^{(1)}, x_{11}^{(2)}, x_{11}^{(3)}, x_{11}^{(4)}) & (x_{12}^{(1)}, x_{12}^{(2)}, x_{12}^{(3)}, x_{12}^{(4)}) \\ (x_{21}^{(1)}, x_{21}^{(2)}, x_{21}^{(3)}, x_{21}^{(4)}) & (x_{22}^{(1)}, x_{22}^{(2)}, x_{22}^{(3)}, x_{22}^{(4)}) \end{pmatrix}.$$

Which can be written as,

$$\tilde{A}\tilde{X} = \begin{pmatrix} (m_1, n_1, \alpha_1, \beta_1) & (m_2, n_2, \alpha_2, \beta_2) \\ (m_3, n_3, \alpha_3, \beta_3) & (m_4, n_4, \alpha_4, \beta_4) \end{pmatrix} \tag{16}$$

where,

$m_1 = min(a_{11}^{(1)} x_{11}^{(1)}, a_{11}^{(1)} x_{11}^{(4)}, a_{11}^{(4)} x_{11}^{(1)}, a_{11}^{(4)} x_{11}^{(4)}) + min(a_{12}^{(1)} x_{21}^{(1)}, a_{12}^{(1)} x_{21}^{(4)}, a_{12}^{(4)} x_{21}^{(1)}, a_{12}^{(4)} x_{21}^{(4)})$,

$n_1 = min(a_{11}^{(1)} x_{12}^{(1)}, a_{11}^{(1)} x_{12}^{(4)}, a_{11}^{(4)} x_{12}^{(1)}, a_{11}^{(4)} x_{12}^{(4)}) + min(a_{12}^{(1)} x_{22}^{(1)}, a_{12}^{(1)} x_{22}^{(4)}, a_{12}^{(4)} x_{22}^{(1)}, a_{12}^{(4)} x_{22}^{(4)})$,

$\alpha_1 = min(a_{21}^{(1)} x_{11}^{(1)}, a_{21}^{(1)} x_{11}^{(4)}, a_{21}^{(4)} x_{11}^{(1)}, a_{21}^{(4)} x_{11}^{(4)}) + min(a_{22}^{(1)} x_{21}^{(1)}, a_{22}^{(1)} x_{21}^{(4)}, a_{22}^{(4)} x_{21}^{(1)}, a_{22}^{(4)} x_{21}^{(4)})$,

$\beta_1 = min(a_{21}^{(1)} x_{12}^{(1)}, a_{21}^{(1)} x_{12}^{(4)}, a_{21}^{(4)} x_{12}^{(1)}, a_{21}^{(4)} x_{12}^{(4)}) + min(a_{22}^{(1)} x_{22}^{(1)}, a_{22}^{(1)} x_{22}^{(4)}, a_{22}^{(4)} x_{22}^{(1)}, a_{22}^{(4)} x_{22}^{(4)})$,

$m_2 = min(a_{11}^{(2)} x_{11}^{(2)}, a_{11}^{(2)} x_{11}^{(3)}, a_{11}^{(3)} x_{11}^{(2)}, a_{11}^{(3)} x_{11}^{(3)}) + min(a_{12}^{(2)} x_{21}^{(2)}, a_{12}^{(2)} x_{21}^{(3)}, a_{12}^{(3)} x_{21}^{(2)}, a_{12}^{(3)} x_{21}^{(3)})$,

$n_2 = min(a_{11}^{(2)} x_{12}^{(2)}, a_{11}^{(2)} x_{12}^{(3)}, a_{11}^{(3)} x_{12}^{(2)}, a_{11}^{(3)} x_{12}^{(3)}) + min(a_{12}^{(2)} x_{22}^{(2)}, a_{12}^{(2)} x_{22}^{(3)}, a_{12}^{(3)} x_{22}^{(2)}, a_{12}^{(3)} x_{22}^{(3)})$,

$\alpha_2 = min(a_{21}^{(2)} x_{11}^{(2)}, a_{21}^{(2)} x_{11}^{(3)}, a_{21}^{(3)} x_{11}^{(1)}, a_{21}^{(3)} x_{11}^{(3)}) + min(a_{22}^{(2)} x_{21}^{(2)}, a_{22}^{(2)} x_{21}^{(3)}, a_{22}^{(3)} x_{21}^{(2)}, a_{22}^{(3)} x_{21}^{(3)})$,

$\beta_2 = min(a_{21}^{(2)} x_{12}^{(2)}, a_{21}^{(2)} x_{12}^{(3)}, a_{21}^{(3)} x_{12}^{(1)}, a_{21}^{(3)} x_{12}^{(3)}) + min(a_{22}^{(2)} x_{22}^{(2)}, a_{22}^{(2)} x_{22}^{(3)}, a_{22}^{(3)} x_{22}^{(2)}, a_{22}^{(3)} x_{22}^{(3)})$,

$m_3 = max(a_{11}^{(2)} x_{11}^{(2)}, a_{11}^{(2)} x_{11}^{(3)}, a_{11}^{(3)} x_{11}^{(2)}, a_{11}^{(3)} x_{11}^{(3)}) + max(a_{12}^{(2)} x_{21}^{(2)}, a_{12}^{(2)} x_{21}^{(3)}, a_{12}^{(3)} x_{21}^{(2)}, a_{12}^{(3)} x_{21}^{(3)})$,

$n_3 = max(a_{11}^{(2)} x_{12}^{(2)}, a_{11}^{(2)} x_{12}^{(3)}, a_{11}^{(3)} x_{12}^{(1)}, a_{11}^{(3)} x_{12}^{(3)}) + max(a_{12}^{(2)} x_{22}^{(2)}, a_{12}^{(2)} x_{22}^{(3)}, a_{12}^{(3)} x_{22}^{(2)}, a_{12}^{(3)} x_{22}^{(3)})$,

$\alpha_3 = max(a_{21}^{(2)} x_{11}^{(2)}, a_{21}^{(2)} x_{11}^{(3)}, a_{21}^{(3)} x_{11}^{(1)}, a_{21}^{(3)} x_{11}^{(3)}) + max(a_{22}^{(2)} x_{21}^{(2)}, a_{22}^{(2)} x_{21}^{(3)}, a_{22}^{(3)} x_{21}^{(2)}, a_{22}^{(3)} x_{21}^{(3)})$,

$\beta_3 = max(a_{21}^{(2)} x_{12}^{(2)}, a_{21}^{(2)} x_{12}^{(3)}, a_{21}^{(3)} x_{12}^{(1)}, a_{21}^{(3)} x_{12}^{(3)}) + max(a_{22}^{(2)} x_{22}^{(2)}, a_{22}^{(2)} x_{22}^{(3)}, a_{22}^{(3)} x_{22}^{(2)}, a_{22}^{(3)} x_{22}^{(3)})$,

$m_4 = max(a_{11}^{(1)} x_{11}^{(1)}, a_{11}^{(1)} x_{11}^{(4)}, a_{11}^{(4)} x_{11}^{(1)}, a_{11}^{(4)} x_{11}^{(4)}) + max(a_{12}^{(1)} x_{21}^{(1)}, a_{12}^{(1)} x_{21}^{(4)}, a_{12}^{(4)} x_{21}^{(1)}, a_{12}^{(4)} x_{21}^{(4)})$,

$n_4 = max(a_{11}^{(1)} x_{12}^{(1)}, a_{11}^{(1)} x_{12}^{(4)}, a_{11}^{(4)} x_{12}^{(1)}, a_{11}^{(4)} x_{12}^{(4)}) + max(a_{12}^{(1)} x_{22}^{(1)}, a_{12}^{(1)} x_{22}^{(4)}, a_{12}^{(4)} x_{22}^{(1)}, a_{12}^{(4)} x_{22}^{(4)})$,

$\alpha_4 = max(a_{21}^{(1)} x_{11}^{(1)}, a_{21}^{(1)} x_{11}^{(4)}, a_{21}^{(4)} x_{11}^{(1)}, a_{21}^{(4)} x_{11}^{(4)}) + max(a_{22}^{(1)} x_{21}^{(1)}, a_{22}^{(1)} x_{21}^{(4)}, a_{22}^{(4)} x_{21}^{(1)}, a_{22}^{(4)} x_{21}^{(1)})$

and,

$\beta_4 = max(a_{21}^{(1)} x_{12}^{(1)}, a_{21}^{(1)} x_{12}^{(4)}, a_{21}^{(4)} x_{12}^{(1)}, a_{21}^{(4)} x_{12}^{(4)}) + max(a_{22}^{(1)} x_{22}^{(1)}, a_{22}^{(1)} x_{22}^{(4)}, a_{22}^{(4)} x_{22}^{(1)}, a_{22}^{(4)} x_{22}^{(4)})$.

Step 2. Multiplying $\tilde{X}\tilde{B}$ using the proposed multiplication operator Proposition 2, Eq. (14):

$$\tilde{X}\tilde{B} = \begin{pmatrix} (x_{11}^{(1)}, x_{11}^{(2)}, x_{11}^{(3)}, x_{11}^{(4)}) & (x_{12}^{(1)}, x_{12}^{(2)}, x_{12}^{(3)}, x_{12}^{(4)}) \\ (x_{21}^{(1)}, x_{21}^{(2)}, x_{21}^{(3)}, x_{21}^{(4)}) & (x_{22}^{(1)}, x_{22}^{(2)}, x_{22}^{(3)}, x_{22}^{(4)}) \end{pmatrix} \begin{pmatrix} (b_{11}^{(1)}, b_{11}^{(2)}, b_{11}^{(3)}, b_{11}^{(4)}) & (b_{12}^{(1)}, b_{12}^{(2)}, b_{12}^{(3)}, b_{12}^{(4)}) \\ (b_{21}^{(1)}, b_{21}^{(2)}, b_{21}^{(3)}, b_{21}^{(4)}) & (b_{22}^{(1)}, b_{22}^{(2)}, b_{22}^{(3)}, b_{22}^{(4)}) \end{pmatrix}$$

Which can be written as,

$$\tilde{X}\tilde{B} = \begin{pmatrix} (\gamma_1, \delta_1, \mu_1, \sigma_1) & (\gamma_2, \delta_2, \mu_2, \sigma_2) \\ (\gamma_3, \delta_3, \mu_3, \sigma_3) & (\gamma_4, \delta_4, \mu_4, \sigma_4) \end{pmatrix} \tag{17}$$

where,

$\gamma_1 = min(x_{11}^{(1)}b_{11}^{(1)}, x_{11}^{(1)}b_{11}^{(4)}, x_{11}^{(4)}b_{11}^{(1)}, x_{11}^{(4)}b_{11}^{(4)}) + min(x_{12}^{(1)}b_{21}^{(1)}, x_{12}^{(1)}b_{21}^{(4)}, x_{12}^{(4)}b_{21}^{(1)}, x_{12}^{(4)}b_{21}^{(4)}),$

$\delta_1 = min(x_{11}^{(1)}b_{12}^{(1)}, x_{11}^{(1)}b_{12}^{(4)}, x_{11}^{(4)}b_{12}^{(1)}, x_{11}^{(4)}b_{12}^{(4)}) + min(x_{12}^{(1)}b_{22}^{(1)}, x_{12}^{(1)}b_{22}^{(4)}, x_{12}^{(4)}b_{22}^{(1)}, x_{12}^{(4)}b_{22}^{(4)}),$

$\mu_1 = min(x_{21}^{(1)}b_{11}^{(1)}, x_{21}^{(1)}b_{11}^{(4)}, x_{21}^{(4)}b_{11}^{(1)}, x_{21}^{(4)}b_{11}^{(4)}) + min(x_{22}^{(1)}b_{21}^{(1)}, x_{22}^{(1)}b_{21}^{(4)}, x_{22}^{(4)}b_{21}^{(1)}, x_{22}^{(4)}b_{21}^{(4)}),$

$\sigma_1 = min(x_{21}^{(1)}b_{12}^{(1)}, x_{21}^{(1)}b_{12}^{(4)}, x_{21}^{(4)}b_{12}^{(1)}, x_{21}^{(4)}b_{12}^{(4)}) + min(x_{22}^{(1)}b_{22}^{(1)}, x_{22}^{(1)}b_{22}^{(4)}, x_{22}^{(4)}b_{22}^{(1)}, x_{22}^{(4)}b_{22}^{(4)}),$

$\gamma_2 = min(x_{11}^{(2)}b_{11}^{(2)}, x_{11}^{(2)}b_{11}^{(3)}, x_{11}^{(3)}b_{11}^{(2)}, x_{11}^{(3)}b_{11}^{(3)}) + min(x_{12}^{(2)}b_{21}^{(2)}, x_{12}^{(2)}b_{21}^{(3)}, x_{12}^{(3)}b_{21}^{(2)}, x_{12}^{(3)}b_{21}^{(3)}),$

$\delta_2 = min(x_{11}^{(2)}b_{12}^{(2)}, x_{11}^{(2)}b_{12}^{(3)}, x_{11}^{(3)}b_{12}^{(2)}, x_{11}^{(3)}b_{12}^{(3)}) + min(x_{12}^{(2)}b_{22}^{(2)}, x_{12}^{(2)}b_{22}^{(3)}, x_{12}^{(3)}b_{22}^{(2)}, x_{12}^{(3)}b_{22}^{(3)}),$

$\mu_2 = min(x_{21}^{(2)}b_{11}^{(2)}, x_{21}^{(2)}b_{11}^{(3)}, x_{21}^{(3)}b_{11}^{(2)}, x_{21}^{(3)}b_{11}^{(3)}) + min(x_{22}^{(2)}b_{21}^{(2)}, x_{22}^{(2)}b_{21}^{(3)}, x_{22}^{(3)}b_{21}^{(2)}, x_{22}^{(3)}b_{21}^{(3)}),$

$\sigma_2 = min(x_{21}^{(2)}b_{12}^{(2)}, x_{21}^{(2)}b_{12}^{(3)}, x_{21}^{(3)}b_{12}^{(2)}, x_{21}^{(3)}b_{12}^{(3)}) + min(x_{22}^{(2)}b_{22}^{(2)}, x_{22}^{(2)}b_{22}^{(3)}, x_{22}^{(3)}b_{22}^{(2)}, x_{22}^{(3)}b_{22}^{(3)}),$

$\gamma_3 = max(x_{11}^{(2)}b_{11}^{(2)}, x_{11}^{(2)}b_{11}^{(3)}, x_{11}^{(3)}b_{11}^{(2)}, x_{11}^{(3)}b_{11}^{(3)}) + max(x_{12}^{(2)}b_{21}^{(2)}, x_{12}^{(2)}b_{21}^{(3)}, x_{12}^{(3)}b_{21}^{(2)}, x_{12}^{(3)}b_{21}^{(3)}),$

$\delta_3 = max(x_{11}^{(2)}b_{12}^{(2)}, x_{11}^{(2)}b_{12}^{(3)}, x_{11}^{(3)}b_{12}^{(2)}, x_{11}^{(3)}b_{12}^{(3)}) + max(x_{12}^{(2)}b_{22}^{(2)}, x_{12}^{(2)}b_{22}^{(3)}, x_{12}^{(3)}b_{22}^{(2)}, x_{12}^{(3)}b_{22}^{(3)}),$

$\mu_3 = max(x_{21}^{(2)}b_{11}^{(2)}, x_{21}^{(2)}b_{11}^{(3)}, x_{21}^{(3)}b_{11}^{(2)}, x_{21}^{(3)}b_{11}^{(3)}) + max(x_{22}^{(2)}b_{21}^{(2)}, x_{22}^{(2)}b_{21}^{(3)}, x_{22}^{(3)}b_{21}^{(2)}, x_{22}^{(3)}b_{21}^{(3)}),$

$\sigma_3 = max(x_{21}^{(2)}b_{12}^{(2)}, x_{21}^{(2)}b_{12}^{(3)}, x_{21}^{(3)}b_{12}^{(2)}, x_{21}^{(3)}b_{12}^{(3)}) + max(x_{22}^{(2)}b_{22}^{(2)}, x_{22}^{(2)}b_{22}^{(3)}, x_{22}^{(3)}b_{22}^{(2)}, x_{22}^{(3)}b_{22}^{(3)}),$

$\gamma_4 = max(x_{11}^{(1)}b_{11}^{(1)}, x_{11}^{(1)}b_{11}^{(4)}, x_{11}^{(4)}b_{11}^{(1)}, x_{11}^{(4)}b_{11}^{(4)}) + max(x_{12}^{(1)}b_{21}^{(1)}, x_{12}^{(1)}b_{21}^{(4)}, x_{12}^{(4)}b_{21}^{(1)}, x_{12}^{(4)}b_{21}^{(4)}),$

$\delta_4 = max(x_{11}^{(1)}b_{12}^{(1)}, x_{11}^{(1)}b_{12}^{(4)}, x_{11}^{(4)}b_{12}^{(1)}, x_{11}^{(4)}b_{12}^{(4)}) + max(x_{12}^{(1)}b_{22}^{(1)}, x_{12}^{(1)}b_{22}^{(4)}, x_{12}^{(4)}b_{22}^{(1)}, x_{12}^{(4)}b_{22}^{(4)}),$

$\mu_4 = max(x_{21}^{(1)}b_{11}^{(1)}, x_{21}^{(1)}b_{11}^{(4)}, x_{21}^{(4)}b_{11}^{(1)}, x_{21}^{(4)}b_{11}^{(4)}) + max(x_{22}^{(1)}b_{21}^{(1)}, x_{22}^{(1)}b_{21}^{(4)}, x_{22}^{(4)}b_{21}^{(1)}, x_{22}^{(4)}b_{21}^{(4)})$

and,

$\sigma_4 = max(x_{21}^{(1)}b_{12}^{(1)}, x_{21}^{(1)}b_{12}^{(4)}, x_{21}^{(4)}b_{12}^{(1)}, x_{21}^{(4)}b_{12}^{(4)}) + max(x_{22}^{(1)}b_{22}^{(1)}, x_{22}^{(1)}b_{22}^{(4)}, x_{22}^{(4)}b_{22}^{(1)}, x_{22}^{(4)}b_{22}^{(4)}).$

Step 3. Adding Eq. (16) and Eq. (17), we get the following:

$$\tilde{A}\tilde{X} + \tilde{X}\tilde{B} = \begin{pmatrix} (m_1, n_1, \alpha_1, \beta_1) & (m_2, n_2, \alpha_2, \beta_2) \\ (m_3, n_3, \alpha_3, \beta_3) & (m_4, n_4, \alpha_4, \beta_4) \end{pmatrix} + \begin{pmatrix} (\gamma_1, \delta_1, \mu_1, \sigma_1) & (\gamma_2, \delta_2, \mu_2, \sigma_2) \\ (\gamma_3, \delta_3, \mu_3, \sigma_3) & (\gamma_4, \delta_4, \mu_4, \sigma_4) \end{pmatrix}.$$

Step 4: By applying Theorem 1, Eq. (15), the arbitrary TrFFSME in Eq. (1) can be written as:

$$\begin{pmatrix} (m_1, n_1, \alpha_1, \beta_1) & (m_2, n_2, \alpha_2, \beta_2) \\ (m_3, n_3, \alpha_3, \beta_3) & (m_4, n_4, \alpha_4, \beta_4) \end{pmatrix} + \begin{pmatrix} (\gamma_1, \delta_1, \mu_1, \sigma_1) & (\gamma_2, \delta_2, \mu_2, \sigma_2) \\ (\gamma_3, \delta_3, \mu_3, \sigma_3) & (\gamma_4, \delta_4, \mu_4, \sigma_4) \end{pmatrix}$$
$$= \begin{pmatrix} (c_{11}^{(1)}, c_{11}^{(2)}, c_{11}^{(3)}, c_{11}^{(4)}) & (c_{12}^{(1)}, c_{12}^{(2)}, c_{12}^{(3)}, c_{12}^{(4)}) \\ (c_{21}^{(1)}, c_{21}^{(2)}, c_{21}^{(3)}, c_{21}^{(4)}) & (c_{22}^{(1)}, c_{22}^{(2)}, c_{22}^{(3)}, c_{22}^{(4)}) \end{pmatrix}. \tag{18}$$

Step 5. The obtained equation Eq. (18) can be converted into the following system of 16 equations. It worth mentioning that the number of the equation obtained from $n \times m$ arbitrary TrFFSME is equal to $2n \times 2m$ equations. Since the proposed method is applied for a 2×2 TrFFSME we will get a system of 16 crisp equations as follows:

$$\begin{cases} m_1 + \gamma_1 = c_{11}^{(1)} \\ n_1 + \delta_1 = c_{12}^{(1)} \\ \alpha_1 + \mu_1 = c_{21}^{(1)} \\ \beta_1 + \sigma_1 = c_{22}^{(1)} \\ m_2 + \gamma_2 = c_{11}^{(2)} \\ n_2 + \delta_2 = c_{12}^{(2)} \\ \alpha_2 + \mu_2 = c_{21}^{(2)} \\ \beta_2 + \sigma_2 = c_{22}^{(2)} \\ m_3 + \gamma_3 = c_{11}^{(3)} \\ n_3 + \delta_3 = c_{12}^{(3)} \\ \alpha_3 + \mu_3 = c_{21}^{(3)} \\ \beta_3 + \sigma_3 = c_{22}^{(3)} \\ m_4 + \gamma_4 = c_{11}^{(4)} \\ n_4 + \delta_4 = c_{12}^{(4)} \\ \alpha_4 + \mu_4 = c_{21}^{(4)} \\ \beta_4 + \sigma_4 = c_{22}^{(4)} \end{cases} \tag{19}$$

This non-linear system of equations Eq. (19) can be solved using the following two-stage numerical algorithm.

4.2 Introduction to the two-stage numerical algorithm.

To our knowledge, the above non-linear system of equations Eq. (19) cannot be solved analytically and has to be solved numerically.

Solving this system of 16 unknowns with MATLAB or Octave using built-in functions gives results that are not most of the time is not the exact solution. In other words, in some cases, it provides a solution to the equation which is not "fuzzy". When adding the "fuzzy number constraints" ($x_{ij}^{(1)} \leq x_{ij}^{(2)} \leq x_{ij}^{(3)} \leq x_{ij}^{(4)}$) to the built-in function "FSOLVE" in MATLAB, it makes the program more unstable (i.e. gives solution incorrect and farther from the correct solution). Therefore, in the following section, a new two-stage algorithm is proposed to solve such examples while imposing the constraints listed above for the solution.

The algorithm's objective is to show that using the proposed numerical programming method, solution(s) can be found.

In this section, the algorithm presented allows finding a solution(s). The first part (first-stage algorithm) is designed to narrow the search of a solution for each vector in the fuzzy solution $\tilde{X}$ Eq. (15), while the second algorithm proceeds to search the four components of each vector within the range found by the first stage algorithm.

4.2.1 Assumptions

The presence of "min" and "max" operators within each equation makes it hard, in most cases, to think about a unique solution, especially that the domain of the variables is the whole set of real numbers.

Although the algorithm developed may support iterations over decimal numbers, the search for solutions was designed with a "unit" step that allows searching for solutions among integer numbers only to save computation time and memory available.

The application of a more robust experimental design would offer faster execution of the algorithm and, therefore, the selection of smaller steps (e.g. 0.1 or 0.01, …). This would allow finding more solutions. This aspect of searching for optimal experimental design is not investigated in the below algorithm.

4.2.2 The two-stage algorithm

As shown in Appendix *A*, the total number of combinations for any system is exponential with respect to the number of variables, which make its application for a system of 16 unknowns with a range of 100 for each variable is computationally expensive, even while taking into account all the assumptions of the previous sections.

Therefore, the two-stage algorithm that is proposed below aims first to reduce the search region for each primary variable (say range $r = 10 \, or \, 16$), which means that the fuzziness of each fuzzy number in the solution is assumed to be within 16 integers, then to solve the system gradually by considering all the combinations within this narrow range.

4.2.2.1 The First Stage Algorithm

This first stage algorithm is designed to find a narrow region of search for each of the four main variables x_{11}, x_{21}, x_{12} and x_{22} (i.e. find the range of

solutions). The algorithm below shows how to find the region of search for each of the four independent variables. This can be seen as searching for an average value $\left(x_{ij}^{Avg}\right)$ of each of the main four vectors in Eq. (15), then suggesting a range around the found value(s).

In order to do so, the first algorithm executes the following steps:

1. Only eight equations of Eq. (19) are considered. Those eight equations are listed in Appendix B. Those equations consider the boundary values in all the matrices (i.e. the superscripts (1) and (4)).

2. In addition, we assume that $x_{ij}^{(1)} = x_{ij}^{(2)} = x_{ij}^{(3)} = x_{ij}^{(4)} = x_{ij}^{Avg}$ for every i and j, this will allow reducing the number of unknowns to 4 unknowns $\left(\tilde{X}_{11}^{Avg}, \tilde{X}_{21}^{Avg}, \tilde{X}_{12}^{Avg}, \tilde{X}_{22}^{Avg}\right)$. In other words, the fuzzy unknown numbers are considered crisp while keeping the same fuzzy multiplication operations.

3. Limiting those variables to 4 independent variables, approximated average values in a range of 100 values can be searched. For example, each of the variables is considered as varying within the interval $[-49, +50]$ ($i.e\ for\ i = -49 : +50$). By allowing each variable to take 100 values, the number of combinations could be $N = 100^4 = 10^8$, which is still computationally feasible (less than 10 s).

4. Selecting the average values for each variable is conducted by substituting the set of combinations in each of the eight equations in Appendix C, "equation after the other". In other words, only the combinations satisfying the first equation are retained and substituted in the following equation, and so on.

5. In case of having more than one x_{ij}^{Avg} solution per variable, those solutions can be considered one after the other in the second stage algorithm, or the average can be considered.

4.2.2.2 The Second Stage Algorithm

The second-stage algorithm consists of finding the fuzzy variables' solutions within the specified intervals found by the first-stage algorithm. In other words, the domain of each variable $x_{ij}^{(k)}$ is defined based on the x_{ij}^{avg} found in the previous algorithm.

Assuming a range of r values for each $x_{ij}^{(k)}$, then potential $x_{ij}^{(k)}$ values would be within the set below:

$x_{ij}^{(k)} \in \left\{x_{ij}^{avg} - \frac{r}{2} + 1, \dots, x_{ij}^{avg} + \frac{r}{2}\right\}$ for every $i, j = 1, 2$

$k = 1, \dots, 4$ with the constraints $x_{ij}^{(1)} \le x_{ij}^{(2)} \le x_{ij}^{(3)} \le x_{ij}^{(4)}$

In the examples considered, the range r is taken equal to 16.

It is worth mentioning that this system can be divided into two quasi-independent[1] systems; each is composed of 8 equations.

---

[1] In case more than one solution is found for variables with superscripts (1) and (4), the retained solutions are those offering solutions for variables with superscripts (2) and (3) within their range values.

The first system contains all equations having variables with superscripts (1) and (4), and the second system includes all the equations having variables with superscripts (2) and (3). This fact allows reducing the computational time for finding solutions significantly. For example, instead of $N = 10^{12}$ iterations for 12 equations with 12 unknowns, one can get $N = 2 \times 10^6$ iterations for solving 2 systems of 6 equations and six unknowns).

Therefore, the 2nd-stage algorithm proceeds following two main steps:

1-   Find a solution for the eight variables $x_{ij}^{(1)}$ and $x_{ij}^{(4)}$, with $i, j = 1, 2$. More details about solving this system and optimizing the number of iterations are given in Appendix D.

2-   The same procedure described in Appendix D is followed to solve the second system of 8 variables $x_{ij}^{(2)}$ and $x_{ij}^{(3)}$ by considering a narrower interval range for those variables, which are $[x_{ij}^{(1)}, x_{ij}^{(4)}]$.

5. Numerical Examples

For all the examples selected, to have computationally feasible solutions, it is assumed that the solution is included in the interval of 200 integers around zero ($i.e. [-99,100]$). Moreover, the spread of the fuzzy numbers is assumed to be within 16 consecutive integers).

Below is a typical example of a system that is solved using the two suggested methods. The first one using the Octave built-in function "fsolve", and the second one using the two sub-algorithms.

Example 4. Consider the following TrFFSME and solve it by the proposed method:

$$\begin{pmatrix} (-12, 22, 35, 52) & (-30, 20, 43, 66) \\ (-25, -10, 29, 33) & (10, 44, 50, 100) \end{pmatrix} \begin{pmatrix} (x_{11}^{(1)}, x_{11}^{(2)}, x_{11}^{(3)}, x_{11}^{(4)}) & (x_{12}^{(1)}, x_{12}^{(2)}, x_{12}^{(3)}, x_{12}^{(4)}) \\ (x_{21}^{(1)}, x_{21}^{(2)}, x_{21}^{(3)}, x_{21}^{(4)}) & (x_{22}^{(1)}, x_{22}^{(2)}, x_{22}^{(3)}, x_{22}^{(4)}) \end{pmatrix}$$

$$+ \begin{pmatrix} (x_{11}^{(1)}, x_{11}^{(2)}, x_{11}^{(3)}, x_{11}^{(4)}) & (x_{12}^{(1)}, x_{12}^{(2)}, x_{12}^{(3)}, x_{12}^{(4)}) \\ (x_{21}^{(1)}, x_{21}^{(2)}, x_{21}^{(3)}, x_{21}^{(4)}) & (x_{22}^{(1)}, x_{22}^{(2)}, x_{22}^{(3)}, x_{22}^{(4)}) \end{pmatrix} \begin{pmatrix} (-25, 0, 49, 53) & (-2, 1, 39, 47) \\ (-30, 19, 32, 65) & (-40, -30, 44, 66) \end{pmatrix}$$

$$= \begin{pmatrix} (-3810, -1885, 202, 2555) & (-8446, -5259, -1429, 3886) \\ (-7815, -3597, -348, 6045) & (-16805, -9266, -554, 4016) \end{pmatrix}$$

Solution.

I)   <u>Solution using the newly developed algorithm</u>

The algorithm proposes the application of 2 sub-algorithms as explained in the previous sections.

a) The first sub-algorithm provides the region of search by finding the "seed" (x_{ij}^{avg}) for each of the fuzzy numbers. For the numerical example suggested above, the obtained seed or x_{ij}^{avg} is

$$x_{11}^{avg} = -31 ; \ x_{12}^{avg} = 15 ; \ x_{12}^{avg} = 7 ; \ x_{22}^{avg} = -100$$

This seed was obtained using a range of 210 integers around 0 with a value of $\varepsilon = 150$.

b) The second sub-algorithm searches for the four fuzzy numbers solutions for the 16 equations using all the combinations around a relatively small range r around x_{ij}^{avg}. For this example, a range $r = 22$ allows us to find solutions for the system.

Below are the four possible solutions found for this system:

$$\tilde{X}_1 = \begin{pmatrix} (-30, -25, -23, -23) & (-3, 5, 6, 7) \\ (5, 6, 12, 15) & (-100, -98, -94, -92) \end{pmatrix},$$

$$\tilde{X}_2 = \begin{pmatrix} (-30, -25, -23, -22) & (-3, 5, 6, 7) \\ (5, 6, 12, 15) & (-100, -98, -94, -92) \end{pmatrix},$$

$$\tilde{X}_3 = \begin{pmatrix} (-30, -25, -23, -21) & (-3, 5, 6, 7) \\ (5, 6, 12, 15) & (-100, -98, -94, -92) \end{pmatrix},$$

and

$$\tilde{X}_4 = \begin{pmatrix} (-30, -25, -23, -20) & (-3, 5, 6, 7) \\ (5, 6, 12, 15) & (-100, -98, -94, -92) \end{pmatrix}.$$

The following Figure 5 shows the arbitrary fuzzy solutions $\tilde{X}_1$, $\tilde{X}_2$, $\tilde{X}_3$ and $\tilde{X}_4$.
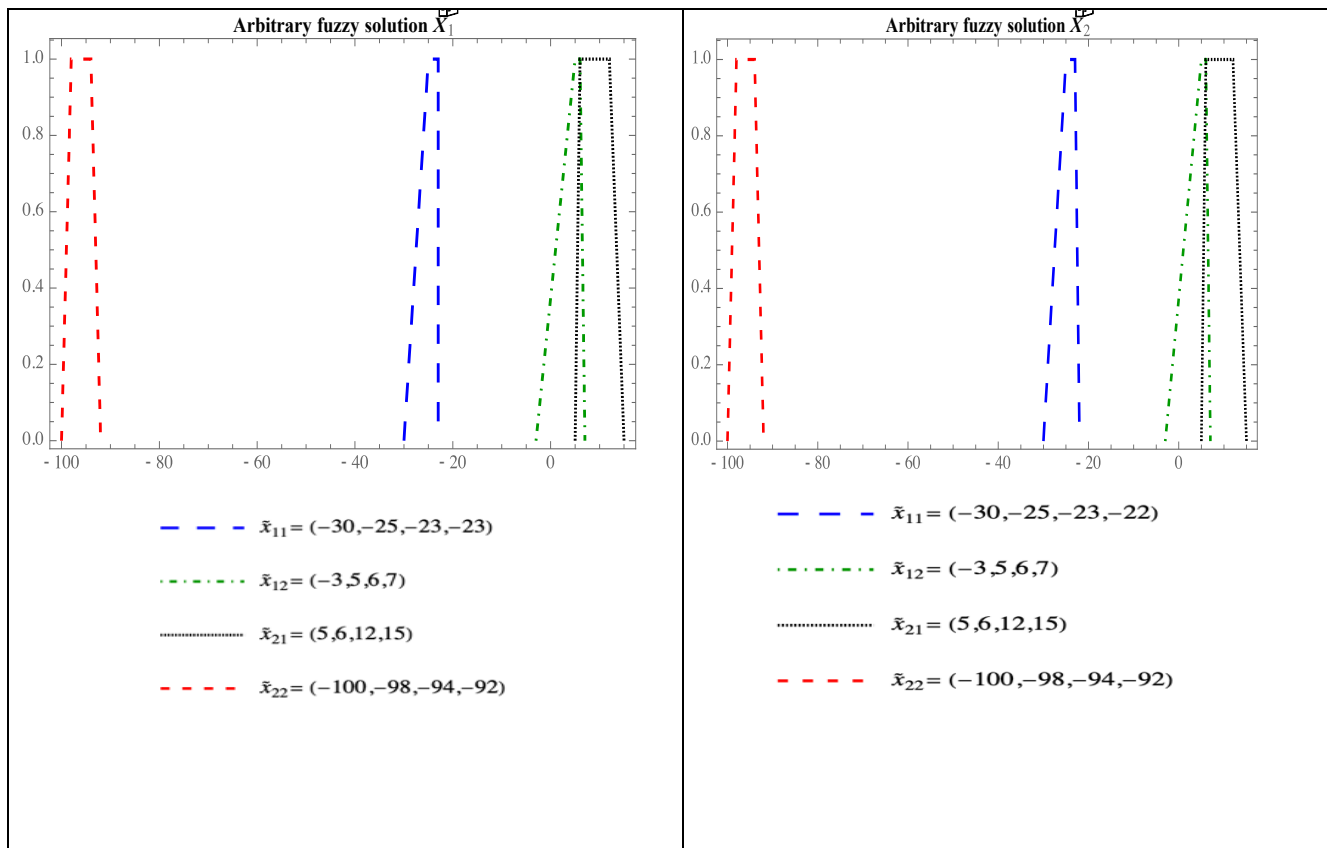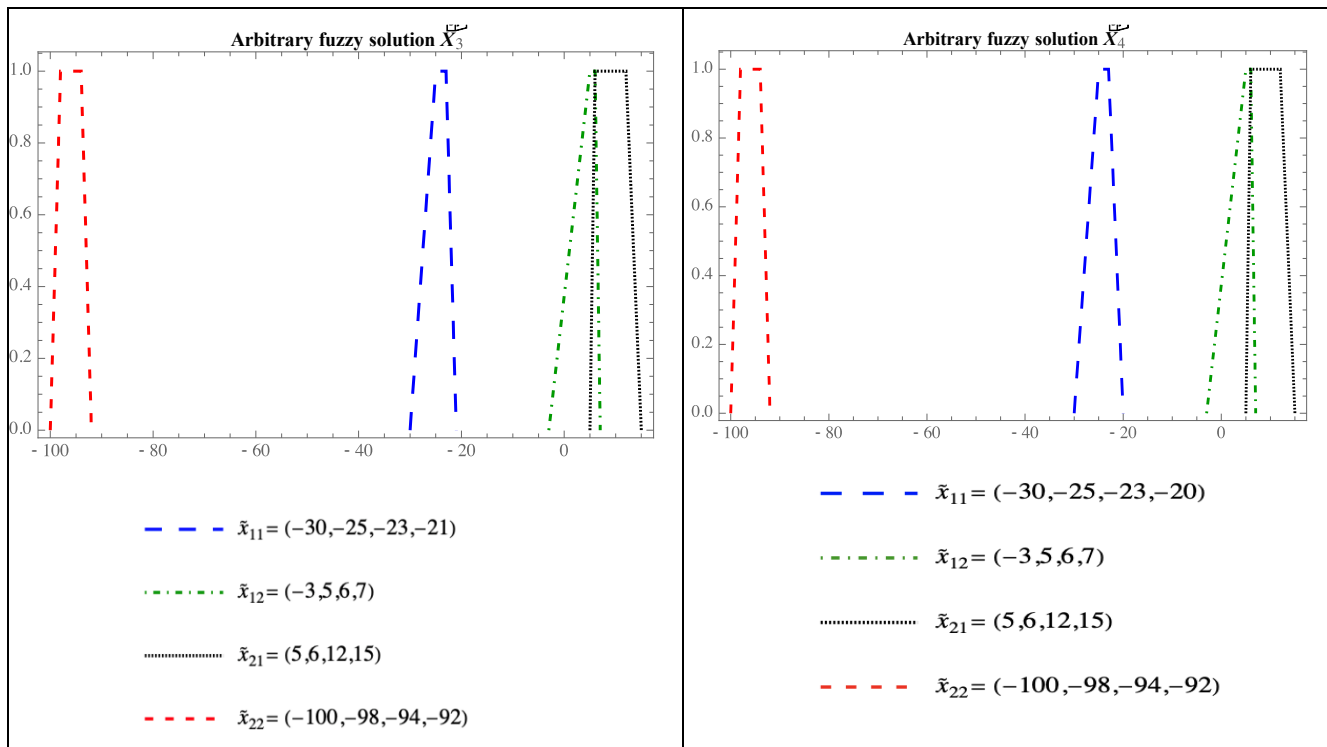
Figure 5. Arbitrary fuzzy solutions $\tilde{X}_1$, $\tilde{X}_2$, $\tilde{X}_3$ and $\tilde{X}_4$ of Example 4.

II)   Solution using MATLAB and the function FSOLVE

By applying directly, the "FSOLVE" algorithm build in MATLAB for the non-linear system over the 16 equations, it gives the solution below:

$$\tilde{X}_{fsolve} = \begin{pmatrix} (-30.23\ , -25, -23,\quad 0.754) & (7.05,\quad 6,\quad 5,\quad 6.74) \\ (6.23,\quad 6,\quad 12,\quad 14.99) & (-100.11, -98, -94-92.13) \end{pmatrix}$$

Substituting this solution into Eq. (1) doesn't satisfy the equation with the proposed matrix $\tilde{C}$.

Adding the constraints $x_{ij}^{(1)} \leq x_{ij}^{(2)} \leq x_{ij}^{(3)} \leq x_{ij}^{(4)}$, it doesn't lead as well to a correct solution. However, this is not the case for all the examples tried. For some, applying "FSOLVE" leads to reasonable solutions.

Remark 1.

The two-stage algorithm found to be a useful alternative to the build-in function FSOLVE inherent to the MATLAB software, due to the following reasons, FSOLVE is based on the trust region method which can solve non-linear systems that are twice differentiable, it is not well suited to handle our min-max non-linear system of equations. In addition, the trust-region method needs a good initial guess to give a good approximation, with our assumption the obtained solution by FSOLVE does not always satisfy the system. Moreover, the FSOLVE function does not include any constraints; in case of adding constraints, the solution to the system of non-linear equations that satisfies the given constraints is not guaranteed to exist. Attempts to find an exact solution using FSOLVE when the fuzzy constraints are added to the min-max non-linear system always failed or produced warnings for possible error.

Remark 2. Extension of the proposed method to other fuzzy numbers and systems.

To our best knowledge, the proposed method is the first one applied to different fuzzy systems and fuzzy numbers without any amendments. For

example, it can be applied to arbitrary FFSME in the form $\tilde{A}\tilde{X} + \tilde{X}\tilde{B} = \tilde{C}$ with TrFNs and TFNs whenever, the mean values in the TrFNs used are equal. In addition, it can also be applied to the arbitrary FFME in the form $\tilde{A}\tilde{X} = \tilde{C}$ with TrFNs and TFNs if we allow $\tilde{B} = 0$, in $\tilde{A}\tilde{X} + \tilde{X}\tilde{B} = \tilde{C}$. Therefore, the proposed method can solve the following fuzzy systems: Arbitrary FFSME with TFNs and TrFNs and arbitrary FFME with TFNs and TrFNs without restriction.

6. Conclusion

In this paper, a new method is developed to solve the arbitrary TrFFSME $\tilde{A}\tilde{X} + \tilde{X}\tilde{B} = \tilde{C}$ based on a new fuzzy arithmetic operation between TrFNs. We obtained finite arbitrary solutions by developing a new arithmetic fuzzy multiplication operator for TrFNs based on the concept of interval arithmetic operations where a fully fuzzy Sylvester matrix equation is transformed into a system of equations that can be solved numerically. The proposed method is applicable for solving arbitrary FFSME with TFNs and TrFNs and arbitrary FFME with TFNs and TrFNs. The numerical algorithm presented in this paper is a first attempt towards solving unrestricted fully fuzzy Sylvester matrix equation with a TrFNs. More work is needed for optimizing the sampling and numerical methods to find all the solutions of the 16 non-linear equations.

Compliance with Ethical Standards

Conflict of interest The authors declare no conflict of interest, financial or otherwise.

Human and animals rights This article does not contain any studies with human participants or animals performed by any authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

[1]   B. Datta, Numerical methods for linear control systems, Academic Press, 2004. https://doi.org/10.1016/B978-0-12-203590-6.X5000-9.

[2]   D.C. Sorensen, A.C. Antoulas, The Sylvester equation and approximate balanced reduction, Linear Algebra Appl. 351–352 (2002) 671–700. https://doi.org/10.1016/S0024-3795(02)00283-5.

[3]   A. Bouhamidi, K. Jbilou, Sylvester Tikhonov-regularization methods in image restoration, J. Comput. Appl. Math. 206 (2007) 86–98. https://doi.org/http://dx.doi.org/10.1016/j.cam.2006.05.028.

[4]   X. Guo, D. Shang, Fuzzy Approximate Solution of Positive Fully Fuzzy Linear Matrix Equations, J. Appl. Math. 2013 (2013) 1–7. https://doi.org/10.1155/2013/178209.

[5]   D. Dubois, H. Prade, Operations on fuzzy numbers, Int. J. Syst. Sci. 9 (1978) 613–626.

[6]   Malkawi, Ahmad, Ibrahim, Solving the fully fuzzy sylvester matrix equation with triangular fuzzy number, Far East J. Math. Sci. 98 (2015) 37–55. https://doi.org/10.17654/FJMSSep2015_037_055.

[7]   G. Malkawi, N. Ahmad, H. Ibrahim, Solving fully fuzzy linear system with the necessary and sufficient condition to have a positive solution, Appl. Math. Inf. Sci. 8 (2014) 1003–1019. https://doi.org/10.12785/amis/080309.

[8]   S. Daud, N. Ahmad, G. Malkawi, Positive fuzzy minimal solution for positive singular fully fuzzy Sylvester matrix equation, in: AIP Conf. Proc., AIP Publishing LLC, 2018: p. 20084. https://doi.org/10.1063/1.5041615.

[9]   S. Daud, N. Ahmad, G. Malkawi, An algorithm for solving an arbitrary triangular fully fuzzy Sylvester matrix equations, in: AIP Conf. Proc., AIP Publishing LLC, 2017: p. 30012. https://doi.org/10.1063/1.5012158.

[10]   S. Daud, N. Ahmad, G. Malkawi, A new solution of pair matrix equations with arbitrary triangular fuzzy numbers, Turkish J. Math. 43 (2019) 1195–1217.

[11]   S. Daud, N. Ahmad, G. Malkawi, Positive Solution of Arbitrary Triangular Fully Fuzzy Sylvester Matrix Equations, Far East J. Math. Sci. 103 (2018) 271–298. https://doi.org/10.17654/ms103020271.

[12]   S. Daud, N. Ahmad, G. Malkawi, Solving arbitrary fully fuzzy Sylvester matrix equations and its theoretical foundation, AIP Conf. Proc. 2013 (2018) 20026. https://doi.org/10.1063/1.5054225.

[13]   K. Dookhitram, R. Lollchund, R.K. Tripathi, M. Bhuruth, Fully fuzzy Sylvester matrix equation, J. Intell. Fuzzy Syst. 28 (2015) 2199–2211. https://doi.org/10.3233/IFS-141502.

[14]   S. Daud, N. Ahmad, G. Malkawi, Positive Solution of Pair Fully Fuzzy Matrix Equations, Malaysian J. Math. Sci. 12 (2018) 383–400.

[15]   S. Daud, N. Ahmad, G. Malkawi, Positive Solution of Arbitrary Triangular Fully Fuzzy Sylvester Matrix Equations, Far East J. Math. Sci. 103 (2018) 271–298. https://doi.org/10.17654/ms103020271.

[16]   J. Fortin, D. Dubois, H. Fargier, Gradual numbers and their application to fuzzy interval analysis, IEEE Trans. Fuzzy Syst. 16 (2008) 388–402.

[17]   K.H. Lee, First course on fuzzy theory and applications, Springer Science & Business Media, 2005. https://doi.org/10.5860/choice.42-5917.

[18]   A. Kaufmann, M.M. Gupta, G. Bohlender, Introduction to Fuzzy Arithmetic, Theory and Applications., Math. Comput. 47 (1986) 762. https://doi.org/10.2307/2008199.

Appendix A: Example of solving system numerically using all possible integer combinations

In order to solve numerically a system of 4 equations with 4 unknowns $f(X_1,X_2,X_3,X_4) = 0$ under the assumptions of integer variables within a range of 10 numbers (i.e. $X_i \in [x_i, x_i + 9]$), below are to possible combinations.

The possible solutions can be any line of table 1 below. The table represents all the possible combinations of variables.

$X1$	$X2$	$X3$	$X4$
$x1$	$x2$	$x3$	$x4$
$x1$	$x2$	$x3$	$x4 + 1$
...	...	...	...
$x1$	$x2$	$x3$	$x4 + 9$
$x1$	$x2$	$x3 + 1$	$x4$
...	...	$x3 + 1$	$x4 + 1$
...	...	...	...
$x1$	$x2$	$x3 + 9$	$x4 + 9$
...	$x2 + 1$	$x3$	$x4$
...	...	...	...
$x1$	$x2 + 9$	$x3 + 9$	$x4 + 9$
$x1 + 1$	$x2$	$x3$	$x4$
...			

$x1 + 9$	$x2 + 9$	$x3 + 9$	$x4 + 9$

Table 1. Permutations of 4 variables with 10 possible integer values each

For this system with independent variables, all the possible integer combinations are 10^4.

In the general case, where a system of n variables X_i, each having r_i values, the number of combinations to be tested as potential solutions, always for independent variables, is:

$$N = \prod_{i=1}^{n} r_i$$

If the range for all variables is the same ($r_i = r$), the number N would become:

$$N = r^n = 10^4$$

If the range $r = 200$ as for the one used in the first-stage algorithm, the number of combinations would be:

$N = 200^4 = 16 * 10^8$.

However, this number was reduced significantly by considering gradually the equations and their potential solutions. For example, the first equation in the first-stage algorithm has 3 unknowns, which reduce the number N of combinations to $N = 200^3 = 8 * 10^6$ instead of $N = 200^4 = 16 * 10^8$.

The retained potential solution is used to be combined with potential values of the 4th unknowns (in the case of first algorithm it is the variable $X_{22}^{(avg)}$).

Appendix B: Finding seeds for the range of main variables (Average values)

The equations considered in the first stage algorithm are those containing the lower and upper bounds of matrices A, B, x and C (i.e. the equations containing superscripts (1) and (4)).

$$
\begin{cases}
m_1 + \gamma_1 = c_{11}^{(1)} & (B.1) \\
n_1 + \delta_1 = c_{12}^{(1)} & (B.2) \\
\alpha_1 + \mu_1 = c_{21}^{(1)} & (B.3) \\
\beta_1 + \sigma_1 = c_{22}^{(1)} & (B.4) \\
\\
m_4 + \gamma_4 = c_{11}^{(4)} & (B.5) \\
n_4 + \delta_4 = c_{12}^{(4)} & (B.6) \\
\alpha_4 + \mu_4 = c_{21}^{(4)} & (B.7) \\
\beta_4 + \sigma_4 = c_{22}^{(4)} & (B.8)
\end{cases}
$$

They are 8 equations and 8 unknowns. By considering $X_{ij}^{(1)} = X_{ij}^{(4)} = X_{ij}^{avg}$, the number of unknowns is reduced to 4. Starting with the first and the fifth equation only 3 unknowns can be considered (as explained in Appendix *A* above).

Appendix C: Relaxing constraints of equations into inequations

Since solving the first stage algorithm (finding the seeds X_{ij}^{avg}) might not lead to a solution even when the original system has one; and this is perfectly normal to happen because of considering the fuzzy variables as crisp. Therefore, the system of equation to be solved by the first-stage

algorithm is relaxed into inequalities. In other words, the equations in Appendix *B* are transformed into the 8 inequalities below:

$$\begin{cases} Abs(m_1 + \gamma_1 - c_{11}^{(1)}) < \varepsilon \\ Abs(n_1 + \delta_1 - c_{12}^{(1)}) < \varepsilon \\ Abs(\alpha_1 + \mu_1 - c_{21}^{(1)}) < \varepsilon \\ Abs(\beta_1 + \sigma_1 - c_{22}^{(1)}) < \varepsilon \\ Abs(m_4 + \gamma_4 - c_{11}^{(4)}) < \varepsilon \\ Abs(n_4 + \delta_4 - c_{12}^{(4)}) < \varepsilon \\ Abs(\alpha_4 + \mu_4 - c_{21}^{(4)}) < \varepsilon \\ Abs(\beta_4 + \sigma_4 - c_{22}^{(4)}) < \varepsilon \end{cases}$$

The value of ε is calibrated in a way to get one or few solutions$\left(x_{ij}^{Avg}\right)$. The algorithm attempts to find values for x_{ij}^{Avg} that makes the inequations above the closest possible solutions in Appendix *B*. It is worth mentioning that small values of ε will not allow finding solutions in most cases.

Appendix D: Solving the 8 equations with superscripts (1) and (4)

Below are the 8 equations with 8 unknowns containing all the variables with superscripts (1) and (4).

$$\begin{cases} m_1 + \gamma_1 = c_{11}^{(1)} & (D.1) \\ n_1 + \delta_1 = c_{12}^{(1)} & (D.2) \\ \alpha_1 + \mu_1 = c_{21}^{(1)} & (D.3) \\ \beta_1 + \sigma_1 = c_{22}^{(1)} & (D.4) \\ m_4 + \gamma_4 = c_{11}^{(4)} & (D.5) \\ n_4 + \delta_4 = c_{12}^{(4)} & (D.6) \\ \alpha_4 + \mu_4 = c_{21}^{(4)} & (D.7) \\ \beta_4 + \sigma_4 = c_{22}^{(4)} & (D.8) \end{cases}$$

The 8 unknowns are: $X_{11}^{(1)}, X_{11}^{(4)}, X_{21}^{(1)}, X_{21}^{(4)}, X_{12}^{(1)}, X_{12}^{(4)}, X_{22}^{(1)}, X_{22}^{(4)}$.

Each of these variables can take any value in its range (e.g. for 10 values, r = 10), with the constraint $x_{ij}^{(1)} \leq x_{ij}^{(4)}$. The number of combinations for each couple of variables $(x_{ij}^{(1)}, x_{ij}^{(4)})$ is equal to

$$\frac{r(r+1)}{2} = \frac{10 \times 11}{2} = 55$$

Those combinations are as follows:

$X_{11}^{(1)}$	$X_{11}^{(4)}$
$x_{11}^{(1)}$	$x_{11}^{(1)}$
$x_{11}^{(1)}$	$x_{11}^{(1)}+1$
...	...
$x_{11}^{(1)}$	$x_{11}^{(1)}+9$
$x_{11}^{(1)}+$	$x_{11}^{(1)}+1$
$x_{11}^{(1)}+$	$x_{11}^{(1)}+2$
...	...
$x_{11}^{(1)}+$	$x_{11}^{(1)}+9$

$x_{11}^{(1)} +$	$x_{11}^{(1)} + 2$
...	...
$x_{11}^{(1)} +$	$x_{11}^{(1)} + 9$
$x_{11}^{(1)} +$	$x_{11}^{(1)} + 8$
$x_{11}^{(1)} +$	$x_{11}^{(1)} + 9$
$x_{11}^{(1)} +$	$x_{11}^{(1)} + 9$

Table 2. Combinations under constraints $x_{11}^{(1)} \leq x_{11}^{(4)}$

1-   The sub-algorithm considers Eq. (D.1) and Eq. (D.5) which include the 6 variables: $x_{11}^{(1)}, x_{11}^{(4)}, x_{21}^{(1)}, x_{21}^{(4)}, x_{12}^{(1)}, x_{12}^{(4)}$. Those variables lead to $55^3 = 166,375$ combinations. Gradually, the algorithm discards combinations that do not satisfy Eq. (D.1).

2-   The retained combinations are passed to Eq. (D.5). Only those combinations   satisfying Eq. (D.1) and Eq. (D.5) are saved in a set called S15, having N15 combinations.

3-   The set S15 is combined with all combinations of variables $x_{22}^{(1)}, x_{22}^{(4)}$. This set called S8var contains all potential solutions of the system of 8 equations. The number of combinations in this case becomes:

$N_{8var} = 55 \times N_{15}$.

4-   Those combinations are then filtered by Eq. (D.2) then Eq. (D.6).

5-   Passed and filtered after by Eq.(D.3) and Eq.(D.7)

6-   And lastly passed to Eq.(D.4) and Eq.(D.8).

The remaining combinations are solutions for the system of 8 equations with the superscripts (1) and (4). In order to find the solution for the remaining 8 equations, the same 2nd stage algorithm is repeated in order to find $x_{ij}^{(2)}$ and $x_{ij}^{(3)}$ by considering the region of search the intervals $[x_{ij}^{(1)}, x_{ij}^{(4)}]$.