symmetry

MDPI

Article

Two-Stage Algorithm for Solving Arbitrary Trapezoidal Fully Fuzzy Sylvester Matrix Equations

Ahmed Abdel Aziz Elsayed [1,2], Bassem Saassouh [3,*], Nazihah Ahmad [2] and Ghassan Malkawi [4]

[1] Department of Mathematics, Institute of Applied Technology, Mohamed Bin Zayed City 33884, United Arab Emirates; Ahmed.elsayed@aths.ac.ae

[2] School of Quantitative Sciences, Universiti Utara Malaysia, Sintok 06010, Kedah, Malaysia; nazihah@uum.edu.my

[3] Academic Support Department, Abu Dhabi Polytechnic College, Abu Dhabi 111499, United Arab Emirates

[4] Faculty of Engineering, Math and Natural Science Division, Higher Colleges of Technology (HCT), Al Ain Campus, Abu Dhabi 17155, United Arab Emirates; gmalkawi@hct.ac.ae

* Correspondence: bassem.saassouh@adpoly.ac.ae

Abstract: Sylvester Matrix Equations (SME) play a central role in applied mathematics, particularly in systems and control theory. A fuzzy theory is normally applied to represent the uncertainty of real problems where the classical SME is extended to Fully Fuzzy Sylvester Matrix Equation (FFSME). The existing analytical methods for solving FFSME are based on Vec-operator and Kronecker product. Nevertheless, these methods are only applicable for nonnegative fuzzy numbers, which limits the applications of the existing methods. Thus, this paper proposes a new numerical method for solving arbitrary Trapezoidal FFSME (TrFFSME), which includes near-zero trapezoidal fuzzy numbers to overcome this limitation. The TrFFSME is converted to a system of non-linear equations based on newly developed arithmetic fuzzy multiplication operations. Then the non-linear system is solved using a newly developed two-stage algorithm. In the first stage algorithm, initial values are determined. Subsequently, the second stage algorithm obtains all possible finite fuzzy solutions. A numerical example is solved to illustrate the proposed method. Besides, this proposed method can solve other forms of fuzzy matrix equations and produces finite fuzzy and non-fuzzy solutions compared to the existing methods.

Keywords: fully fuzzy sylvester matrix equations; fuzzy matrix equation; numerical fuzzy solution; trapezoidal fuzzy multiplication

check for updates

Citation: Elsayed, A.A.A.; Saassouh, B.; Ahmad, N.; Malkawi, G. Two-Stage Algorithm for Solving Arbitrary Trapezoidal Fully Fuzzy Sylvester Matrix Equations. *Symmetry* **2022**, *14*, 446. https://doi.org/10.3390/sym14030446

Academic Editors: Jian Zhou, Ke Wang and Yuanyuan Liu

Received: 8 December 2021
Accepted: 13 February 2022
Published: 23 February 2022

1. Introduction

SME in the form $AX + XB = C$ plays a vital role in many fields such as in control systems [1], medical imaging data acquisition, model reduction [2] and stochastic control, in addition to image processing and filtering [3]. Considering any uncertainty problems, such as conflicting requirements during the system process, instability of environmental conditions and the distraction of any elements or noise, the classical SME is sometimes ill-equipped to handle uncertainty and vagueness in real-life situations; therefore, the crisp numbers need to be replaced by fuzzy numbers. Fuzzy logic has been studied since the 1920s, as infinite valued logic by Lukasiewicz and Tarski [4]. The fuzzy set theory was introduced by Lotfi Zadeh [5] in 1965, while the set theory was developed by Georg Cantor [6]. Fuzzy Relation Equations (FREs) with the max-min composition was first studied by Sanchez [7]. According to Sanchez's theorem, if a system of FREs has a solution, then the solution set is partially ordered. Therefore, the solution set of the given equations can be characterized by the maximum solution and all the minimal solutions. Due to the practicality of FREs, solving FREs has become one of the most extensively studied problems in the field of fuzzy sets and fuzzy logic [8]. The theory and applications of FREs can be found in Di Nola et al. [9], which indicated that if the solvability of max-continuous t-norm FREs is assumed, then the solution set for the FREs can be fully determined from a unique greatest solution and all minimal solutions, and the number of minimal solutions is always finite. Since then, FREs based on various compositions have been investigated. Some common compositions include max-min [9–14], max-product [15–18], max-Archimedean t-norm [19,20], u-norm [21], max t-norm [22] and max arithmetic mean [23,24]. The conditions for the existence of a solution to the inverse problem concerned with FRE are investigated in [25], a finite system of FREs with sup-T composition was studied in [26] and a system of FREs was investigated in [27,28]. It is worth mentioning that Kyosev Yordan [29] presented an original software for solving inverse problem resolution for the Fuzzy Linear Systems of Equations together with popular function for FREs, min-max, alpha, epsilon compositions of fuzzy matrices, min-max systems, systems of fuzzy intuitionistic equations, problems for finite fuzzy machines and fuzzy linear programming problem. However, this software needs to be modified to solve SME with Trapezoidal Fuzzy Numbers (TrFNS).

The SME can be extended to a fuzzy Sylvester matrix Equation (FSME) in the form $A\widetilde{X} + \widetilde{X}B = \widetilde{C}$ if the solution matrix $\widetilde{X}$ and the constant matrix $\widetilde{C}$ are in fuzzy form. The FSME was studied in [30,31], where the Kronecker product was applied to convert the FSME to a fuzzy linear system. However, this method was only applicable for small-sized FSME. In order to overcome this shortcoming, authors in [32] applied Dubois and Prade's multiplication operations [33] to convert the FSME to a crisp SME and then the fuzzy solution obtained by applying Bartle's Stewart method. In addition to the FSME, when all the SME's parameters are in fuzzy form, it is called a FFSME.

Definition 1. *The matrix equation that can be written as*

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

where, $\widetilde{A} = (\widetilde{a}_{ij})_{n\times n}$, $\widetilde{B} = \left(\widetilde{b}_{ij}\right)_{m\times m}$, $\widetilde{C} = (\widetilde{c}_{ij})_{m\times m}$ *and* $\widetilde{X} = (\widetilde{x}_{ij})_{n\times m}$ *are fuzzy matrices respectively is called FFSME.*

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

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

Triangular fully fuzzy Sylvester matrix equation (TFFSME) has been studied analytically by Shang, Guo and Bao [34], where the TFFSME is converted to a system of crisp linear matrix equations by applying Dubois and Prade's arithmetic operator for multiplication [33]. 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 [35] 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 [36]. 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 shortcomings in these methods, Daud, Ahmad and Malkawi [37] obtained a positive solution for singular TFFSME by applying an associated linear matrix system approach, where the solution was obtained by using the pseudoinverse method. Recently, authors in [10] 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. [38] 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, 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 obtaining the inverse of $mn \times mn$ matrices, which is not possible for large systems.

A study was conducted by [39] on the TFFSME in the form $\widetilde{A}\widetilde{X} - \widetilde{X}\widetilde{B} = \widetilde{C}$, 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 [40] 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, restricted only to positive fuzzy numbers with very small fuzziness [41]. Therefore, these 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, and these methods cannot detect all possible fuzzy solutions. Therefore, to deal with this shortcoming, this paper presents a new numerical method for solving TrFFSME with arbitrary TrFNs, where the TrFFSME is converted 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, the solution to the non-linear system can be obtained by a newly developed two-stage algorithm where the first stage algorithm reduces the search area for the fuzzy solution and the second stage algorithm finds 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 the basic arithmetic operations of fuzzy numbers [42].

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:

$$A \cdot B = [min(a_1b_1, a_1b_2, a_2b_1, a_2b_2), max(a_1b_1, a_1b_2, a_2b_1, a_2b_2)] \tag{4}$$

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

$$\begin{aligned} A/B &= [a_1, a_2]/[b_1, b_2] \\ &= [min(a_1/b_1, a_1/b_2, a_2/b_1, a_2/b_2), max(a_1/b_1, a_1/b_2, a_2/b_1, a_2/b_2)] \end{aligned}$$

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*

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

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* $A_\alpha = \left[a_1^\alpha, a_2^\alpha\right]$, $B_\alpha = \left[b_1^\alpha, b_2^\alpha\right]$ *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

$$A_\alpha + B_\alpha = [a_1^\alpha, a_2^\alpha] + [b_1^\alpha, b_2^\alpha] = [a_1^\alpha + b_1^\alpha, a_2^\alpha + b_2^\alpha]. \tag{10}$$

(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

$$\begin{aligned} A_\alpha \cdot B_\alpha &= \left[a_1^\alpha, a_2^\alpha\right] \cdot \left[b_1^\alpha, b_2^\alpha\right]. \\ &= \left[min\left(a_1^\alpha b_1^\alpha, a_1^\alpha b_2^\alpha, a_2^\alpha b_1^\alpha, a_2^\alpha b_2^\alpha\right), max\left(a_1^\alpha b_1^\alpha, a_1^\alpha b_2^\alpha, a_2^\alpha b_1^\alpha, a_2^\alpha b_2^\alpha\right)\right] \end{aligned} \tag{11}$$

(IV) Equality

Two intervals $A_\alpha = \left[a_1^\alpha, a_2^\alpha\right]$, *and* $B_\alpha = \left[b_1^\alpha, b_2^\alpha\right]$ *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 [33,43].

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

Definition 5. *A fuzzy set* $\widetilde{A}$, *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. $\widetilde{A}$ *is convex, i.e.,*

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

$\widetilde{A}$ *is normal, i.e.,*

2. $\exists\ x_0 \in R$ *such that* $\mu_{\widetilde{A}}(\ x_0) = 1$.
3. $\mu_{\widetilde{A}}$ *is piecewise continuous.*

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

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

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

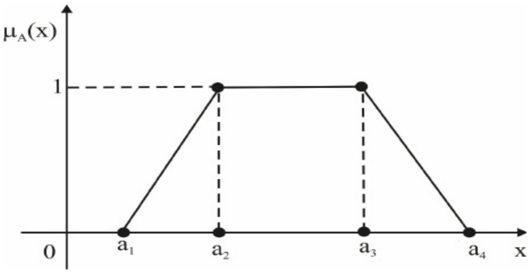

Figure 1. Representation of TrFN (a_1, a_2, a_3, a_4).

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

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

Definition 8. *Operation of TrFNs.*
The arithmetic operations of TrFNs are presented as follows, let
$\widetilde{A} = (a_1,\ a_2,\ a_3,\ a_4)$,
$\widetilde{B} = (b_1,\ b_2,\ b_3,\ b_4)$ *be two TrFNs then:*

1. *Subtraction*

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

2. *Addition*

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

3. *Symmetric image*

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

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

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

4. *Equality: The fuzzy numbers $\widetilde{A} = (a_1,\ a_2,\ a_3,\ a_4)$ and $\widetilde{B} = (b_1,\ b_2,\ b_3,\ b_4)$ are equal iff*

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

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 $\widetilde{A} = (a_1,\ a_2,\ a_3,\ a_4)$ can be written as:*

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

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

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

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,

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

□

The following propositions discuss new arithmetic multiplication operations between TrFNs, namely Ahmd Multiplication Operations (AMO).

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

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

where

$$\begin{aligned} a &= min(a_1b_1,\ a_1b_4,\ a_4b_1,\ a_4b_4) \\ h &= min(a_2b_2,\ a_2b_3,\ a_3b_2,\ a_3b_3) \\ m &= max(a_2b_2,\ a_2b_3,\ a_3b_2,\ a_3b_3) \\ d &= max(a_1b_1,\ a_1b_4,\ a_4b_1,\ a_4b_4) \end{aligned}$$

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

$$\widetilde{A}_\alpha = [a_1^\alpha, a_2^\alpha] = [(a_2 - a_1)\alpha + a_1, -(a_4 - a_3)\alpha + a_4]$$

$$\widetilde{B}_\alpha = [b_1^\alpha, b_2^\alpha] = [(b_2 - b_1)\alpha + b_1, -(b_4 - b_3)\alpha + b_4]$$

$\forall\alpha \in [0,1]$ respectively.
By applying the multiplication operations of $\alpha - cut$ interval in Definition 3 in Equation (11) on $\widetilde{A}_\alpha$ and $\widetilde{B}_\alpha$ we get:

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

where

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

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

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

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

$$A_0 \times B_0 = [e_1, e_2] = [min(a_1b_1,\ a_1b_4,\ a_4b_1,\ a_4b_4), max(a_1b_1,\ a_1b_4,\ a_4b_1,\ a_4b_4)]$$

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

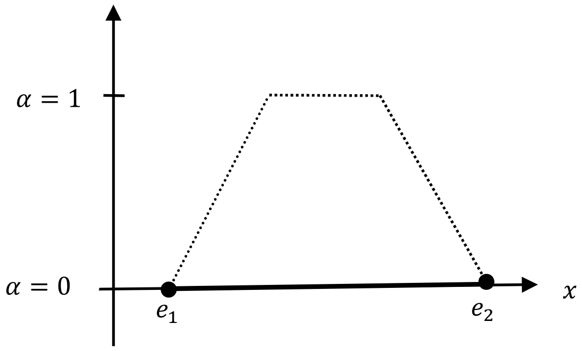

Figure 2. The product $A_0 \times B_0$ at α = 0.

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

$$A_1 \times B_1 = [e_1, e_2] = [min(a_2b_2,\ a_2b_3,\ a_3b_2,\ a_3b_3), max(a_2b_2,\ a_2b_3,\ a_3b_2,\ a_3b_3)]$$

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

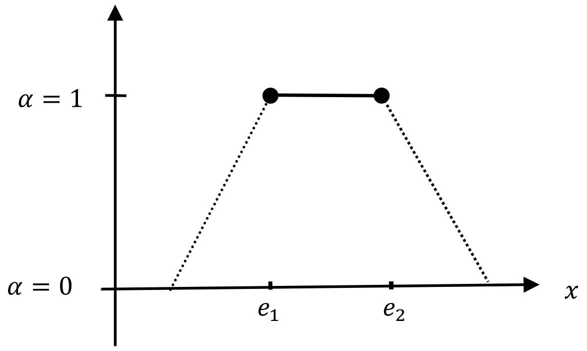

Figure 3. The product $A_1 \times B_1$ at α = 0.

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

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

where,

$$\begin{aligned} a &= min(a_1b_1,\ a_1b_4,\ a_4b_1,\ a_4b_4) \\ h &= min(a_2b_2,\ a_2b_3,\ a_3b_2,\ a_3b_3), \\ m &= max(a_2b_2,\ a_2b_3,\ a_3b_2,\ a_3b_3) \\ d &= max(a_1b_1,\ a_1b_4,\ a_4b_1,\ a_4b_4). \end{aligned}$$

□

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

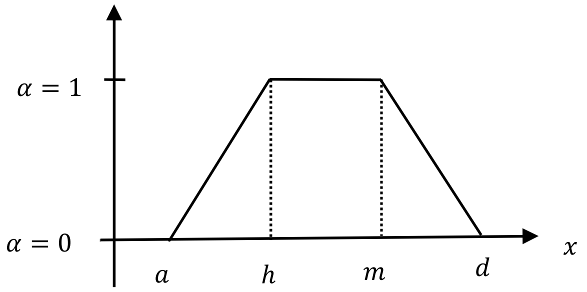

Figure 4. The product $\widetilde{A}\widetilde{B}$.

Definition 9. *If $\widetilde{A}$ and $\widetilde{B}$ are two arbitrary TrFNs respectively, then the multiplication $\widetilde{A}\widetilde{B} = (a, h, m, d)$ is called Ahmd Arithmetic Multiplication Operator (AMO) for Arbitrary TrFNs.*

The implementation of AMO is illustrated in the following example.

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

$$\begin{aligned} a &= min(-4 \times -5, -4 \times 7,\ 3 \times -5,\ 3 \times 7) = -28 \\ h &= min(-2 \times 2, -2 \times 4,\ 1 \times 2,\ 1 \times 4) = -8 \\ m &= max(-2 \times 2, -2 \times 4,\ 1 \times 2,\ 1 \times 4) = 4 \\ d &= max(-4 \times -5, -4 \times 7,\ 3 \times -5,\ 3 \times 7) = 21 \end{aligned}$$

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

Corollary 1. *Positive TrFNs arithmetic multiplication operation.*
If $\widetilde{A} = (a_1,\ a_2,\ a_3,\ a_4)$, $\widetilde{B} = (b_1,\ b_2,\ b_3,\ b_4)$ be two positive TrFNs then:

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

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

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

Using Definition 2 by Equation (5) $\widetilde{A}\widetilde{B}$ can be reduced as follows:

$$\begin{aligned} a &= min(a_1b_1,\ a_1b_4,\ a_4b_1,\ a_4b_4) = a_1b_1 \\ h &= min(a_2b_2,\ a_2b_3,\ a_3b_2,\ a_3b_3) = a_2b_2, \\ m &= max(a_2b_2,\ a_2b_3,\ a_3b_2,\ a_3b_3) = a_3b_3 \\ d &= max(a_1b_1,\ a_1b_4,\ a_4b_1,\ a_4b_4) = a_4b_4. \end{aligned}$$

Thus,

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

□

Corollary 2. *Negative TrFNs arithmetic multiplication operation.*
If $\widetilde{A} = (a_1,\ a_2,\ a_3,\ a_4)$, $\widetilde{B} = (b_1,\ b_2,\ b_3,\ b_4)$ *let them be two negative TrFNs, then:*

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

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

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

Using Definition 2 by Equation (6) $\widetilde{A}\widetilde{B}$ can be reduced as follows:

$$\begin{aligned} a &= min(a_1b_1,\ a_1b_4,\ a_4b_1,\ a_4b_4) = a_4b_4 \\ h &= min(a_2b_2,\ a_2b_3,\ a_3b_2,\ a_3b_3) = a_3b_3, \\ m &= max(a_2b_2,\ a_2b_3,\ a_3b_2,\ a_3b_3) = a_2b_2 \\ d &= max(a_1b_1,\ a_1b_4,\ a_4b_1,\ a_4b_4) = a_1b_1. \end{aligned}$$

Thus,

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

□

Corollary 3. *Positive and negative TrFNs arithmetic multiplication operation.*
If $\widetilde{A} = (a_1,\ a_2,\ a_3,\ a_4) > 0$, $\widetilde{B} = (b_1,\ b_2,\ b_3,\ b_4) < 0$ *let them be two TrFNs, then:*

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

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

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

Using Definition 2 by Equation (7) $\widetilde{A}\widetilde{B}$ can be reduced as follows:

$$\begin{aligned} a &= min(a_1b_1,\ a_1b_4,\ a_4b_1,\ a_4b_4) = a_4b_1, \\ h &= min(a_2b_2,\ a_2b_3,\ a_3b_2,\ a_3b_3) = a_3b_2, \\ m &= max(a_2b_2,\ a_2b_3,\ a_3b_2,\ a_3b_3) = a_2b_3 \\ d &= max(a_1b_1,\ a_1b_4,\ a_4b_1,\ a_4b_4) = a_1b_4. \end{aligned}$$

Thus,

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

□

Corollary 4. *Negative and positive TrFNs arithmetic multiplication operation.*
If $\widetilde{A} = (a_1,\ a_2,\ a_3,\ a_4) < 0$, $\widetilde{B} = (b_1,\ b_2,\ b_3,\ b_4) > 0$ *let them be two TrFNs, then:*

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

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

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

Using Definition 2 by Equation (8) $\widetilde{A}\widetilde{B}$ can be reduced as follows:

$$\begin{aligned} a &= min(a_1b_1,\ a_1b_4,\ a_4b_1,\ a_4b_4) = a_1b_4, \\ h &= min(a_2b_2,\ a_2b_3,\ a_3b_2,\ a_3b_3) = a_2b_3, \\ m &= max(a_2b_2,\ a_2b_3,\ a_3b_2,\ a_3b_3) = a_3b_2 \\ d &= max(a_1b_1,\ a_1b_4,\ a_4b_1,\ a_4b_4) = a_4b_1. \end{aligned}$$

Thus,

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

□

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 AMO proposed in Proposition 2. Figure 5 displays the steps required for solving the arbitrary TrFFSME.

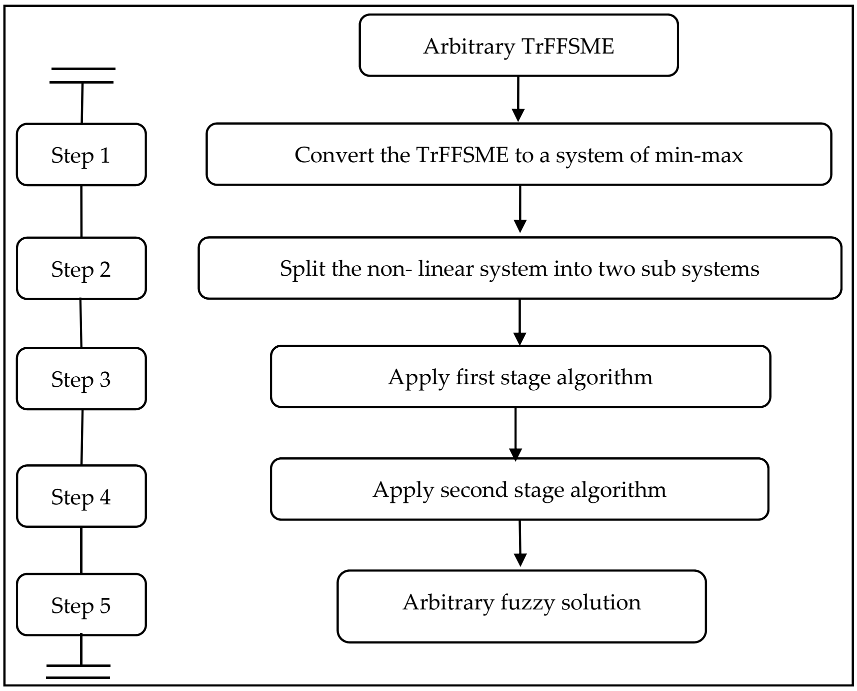
Arbitrary TrFFSME
Step 1
Convert the TrFFSME to a system of min-max
Step 2
Split the non- linear system into two sub systems
Step 3
Apply first stage algorithm
Step 4
Apply second stage algorithm
Step 5
Arbitrary fuzzy solution

Figure 5. Flow chart of the proposed method for solving the arbitrary TrFFSME.

In the following theorem, the FFSME Equation (1) is extended into systems matrix equations.

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

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

Proof. Let $\widetilde{A} = \left(\widetilde{a}_{ij}\right)_{n\times n} = (a_{ij}^{(1)}, a_{ij}^{(2)}, a_{ij}^{(3)}, a_{ij}^{(4)})$, $\widetilde{B} = \left(\widetilde{b}_{ij}\right)_{m\times m} = \left(\widetilde{b}_{ij}\right) = (b_{ij}^{(1)}, b_{ij}^{(2)}, b_{ij}^{(3)}, b_{ij}^{(4)})$, $\widetilde{X} = \left(\widetilde{x}_{ij}\right)_{n\times m} = (x_{ij}^{(1)}, x_{ij}^{(2)}, x_{ij}^{(3)}, x_{ij}^{(4)})$ and $\widetilde{C} = \left(\widetilde{c}_{ij}\right)_{n\times m} = (c_{ij}^{(1)}, c_{ij}^{(2)}, c_{ij}^{(3)}, c_{ij}^{(4)})$ be arbitrary TrFNs. Applying AMO in Proposition 2, $\widetilde{A}\widetilde{X}$ is obtained as follows:

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

where,

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

And,

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

where,

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

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

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

In the following Section 4.1, we propose a method for solving arbitrary TrFFSME. □

4.1. Arbitrary Solution to The Arbitrary TrFFSME

In this section, the arbitrary solution to the arbitrary TrFFSME is discussed. Without loss of generality, we will assume $\widetilde{A}$ to be 2×2. Then $\widetilde{A}$ can be written as:

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

Similarly,

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

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

and,

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

The steps of the proposed method are as follows:

Step 1. Multiplying $\widetilde{A}\widetilde{X}$ using AMO in Proposition 2, Equation (14) as follows:

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

Which can be written as,

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

where,

$$\begin{aligned}
m_1 &= min\left(a_{11}^{(1)}x_{11}^{(1)}, a_{11}^{(1)}x_{11}^{(4)}, a_{11}^{(4)}x_{11}^{(1)}, a_{11}^{(4)}x_{11}^{(4)}\right) + min\left(a_{12}^{(1)}x_{21}^{(1)}, a_{12}^{(1)}x_{21}^{(4)}, a_{12}^{(4)}x_{21}^{(1)}, a_{12}^{(4)}x_{21}^{(4)}\right), \\
n_1 &= min\left(a_{11}^{(1)}x_{12}^{(1)}, a_{11}^{(1)}x_{12}^{(4)}, a_{11}^{(4)}x_{12}^{(1)}, a_{11}^{(4)}x_{12}^{(4)}\right) + min\left(a_{12}^{(1)}x_{22}^{(1)}, a_{12}^{(1)}x_{22}^{(4)}, a_{12}^{(4)}x_{22}^{(1)}, a_{12}^{(4)}x_{22}^{(4)}\right), \\
\alpha_1 &= min\left(a_{21}^{(1)}x_{11}^{(1)}, a_{21}^{(1)}x_{11}^{(4)}, a_{21}^{(4)}x_{11}^{(1)}, a_{21}^{(4)}x_{11}^{(4)}\right) + min\left(a_{22}^{(1)}x_{21}^{(1)}, a_{22}^{(1)}x_{21}^{(4)}, a_{22}^{(4)}x_{21}^{(1)}, a_{22}^{(4)}x_{21}^{(4)}\right), \\
\beta_1 &= min\left(a_{21}^{(1)}x_{12}^{(1)}, a_{21}^{(1)}x_{12}^{(4)}, a_{21}^{(4)}x_{12}^{(1)}, a_{21}^{(4)}x_{12}^{(4)}\right) + min\left(a_{22}^{(1)}x_{22}^{(1)}, a_{22}^{(1)}x_{22}^{(4)}, a_{22}^{(4)}x_{22}^{(1)}, a_{22}^{(4)}x_{22}^{(4)}\right), \\
m_2 &= min\left(a_{11}^{(2)}x_{11}^{(2)}, a_{11}^{(2)}x_{11}^{(3)}, a_{11}^{(3)}x_{11}^{(2)}, a_{11}^{(3)}x_{11}^{(3)}\right) + min\left(a_{12}^{(2)}x_{21}^{(2)}, a_{12}^{(2)}x_{21}^{(3)}, a_{12}^{(3)}x_{21}^{(2)}, a_{12}^{(3)}x_{21}^{(3)}\right), \\
n_2 &= min\left(a_{11}^{(2)}x_{12}^{(2)}, a_{11}^{(2)}x_{12}^{(3)}, a_{11}^{(3)}x_{12}^{(2)}, a_{11}^{(3)}x_{12}^{(3)}\right) + min\left(a_{12}^{(2)}x_{22}^{(2)}, a_{12}^{(2)}x_{22}^{(3)}, a_{12}^{(3)}x_{22}^{(2)}, a_{12}^{(3)}x_{22}^{(3)}\right), \\
\alpha_2 &= min\left(a_{21}^{(2)}x_{11}^{(2)}, a_{21}^{(2)}x_{11}^{(3)}, a_{21}^{(3)}x_{11}^{(1)}, a_{21}^{(3)}x_{11}^{(3)}\right) + min\left(a_{22}^{(2)}x_{21}^{(2)}, a_{22}^{(2)}x_{21}^{(3)}, a_{22}^{(3)}x_{21}^{(2)}, a_{22}^{(3)}x_{21}^{(3)}\right), \\
\beta_2 &= min\left(a_{21}^{(2)}x_{12}^{(2)}, a_{21}^{(2)}x_{12}^{(3)}, a_{21}^{(3)}x_{12}^{(1)}, a_{21}^{(3)}x_{12}^{(3)}\right) + min\left(a_{22}^{(2)}x_{22}^{(2)}, a_{22}^{(2)}x_{22}^{(3)}, a_{22}^{(3)}x_{22}^{(2)}, a_{22}^{(3)}x_{22}^{(3)}\right), \\
m_3 &= max\left(a_{11}^{(2)}x_{11}^{(2)}, a_{11}^{(2)}x_{11}^{(3)}, a_{11}^{(3)}x_{11}^{(2)}, a_{11}^{(3)}x_{11}^{(3)}\right) + max\left(a_{12}^{(2)}x_{21}^{(2)}, a_{12}^{(2)}x_{21}^{(3)}, a_{12}^{(3)}x_{21}^{(2)}, a_{12}^{(3)}x_{21}^{(3)}\right), \\
n_3 &= max\left(a_{11}^{(2)}x_{12}^{(2)}, a_{11}^{(2)}x_{12}^{(3)}, a_{11}^{(3)}x_{12}^{(1)}, a_{11}^{(3)}x_{12}^{(3)}\right) + max\left(a_{12}^{(2)}x_{22}^{(2)}, a_{12}^{(2)}x_{22}^{(3)}, a_{12}^{(3)}x_{22}^{(2)}, a_{12}^{(3)}x_{22}^{(3)}\right), \\
\alpha_3 &= max\left(a_{21}^{(2)}x_{11}^{(2)}, a_{21}^{(2)}x_{11}^{(3)}, a_{21}^{(3)}x_{11}^{(1)}, a_{21}^{(3)}x_{11}^{(3)}\right) + max\left(a_{22}^{(2)}x_{21}^{(2)}, a_{22}^{(2)}x_{21}^{(3)}, a_{22}^{(3)}x_{21}^{(2)}, a_{22}^{(3)}x_{21}^{(3)}\right), \\
\beta_3 &= max\left(a_{21}^{(2)}x_{12}^{(2)}, a_{21}^{(2)}x_{12}^{(3)}, a_{21}^{(3)}x_{12}^{(1)}, a_{21}^{(3)}x_{12}^{(3)}\right) + max\left(a_{22}^{(2)}x_{22}^{(2)}, a_{22}^{(2)}x_{22}^{(3)}, a_{22}^{(3)}x_{22}^{(2)}, a_{22}^{(3)}x_{22}^{(3)}\right), \\
m_4 &= max\left(a_{11}^{(1)}x_{11}^{(1)}, a_{11}^{(1)}x_{11}^{(4)}, a_{11}^{(4)}x_{11}^{(1)}, a_{11}^{(4)}x_{11}^{(4)}\right) + max\left(a_{12}^{(1)}x_{21}^{(1)}, a_{12}^{(1)}x_{21}^{(4)}, a_{12}^{(4)}x_{21}^{(1)}, a_{12}^{(4)}x_{21}^{(4)}\right), \\
n_4 &= max\left(a_{11}^{(1)}x_{12}^{(1)}, a_{11}^{(1)}x_{12}^{(4)}, a_{11}^{(4)}x_{12}^{(1)}, a_{11}^{(4)}x_{12}^{(4)}\right) + max\left(a_{12}^{(1)}x_{22}^{(1)}, a_{12}^{(1)}x_{22}^{(4)}, a_{12}^{(4)}x_{22}^{(1)}, a_{12}^{(4)}x_{22}^{(4)}\right), \\
\alpha_4 &= max\left(a_{21}^{(1)}x_{11}^{(1)}, a_{21}^{(1)}x_{11}^{(4)}, a_{21}^{(4)}x_{11}^{(1)}, a_{21}^{(4)}x_{11}^{(4)}\right) + max\left(a_{22}^{(1)}x_{21}^{(1)}, a_{22}^{(1)}x_{21}^{(4)}, a_{22}^{(4)}x_{21}^{(1)}, a_{22}^{(4)}x_{21}^{(4)}\right)
\end{aligned}$$

and,

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

Step 2. Similarly, multiplying $\widetilde{X}\widetilde{B}$ using AMO in Proposition 2, Equation (14) as follows:

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

Which can be written as,

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

where,

$$\begin{aligned}
\gamma_1 &= min\left(x_{11}^{(1)}b_{11}^{(1)}, x_{11}^{(1)}b_{11}^{(4)}, x_{11}^{(4)}b_{11}^{(1)}, x_{11}^{(4)}b_{11}^{(4)}\right) + min\left(x_{12}^{(1)}b_{21}^{(1)}, x_{12}^{(1)}b_{21}^{(4)}, x_{12}^{(4)}b_{21}^{(1)}, x_{12}^{(4)}b_{21}^{(4)}\right), \\
\delta_1 &= min\left(x_{11}^{(1)}b_{12}^{(1)}, x_{11}^{(1)}b_{12}^{(4)}, x_{11}^{(4)}b_{12}^{(1)}, x_{11}^{(4)}b_{12}^{(4)}\right) + min\left(x_{12}^{(1)}b_{22}^{(1)}, x_{12}^{(1)}b_{22}^{(4)}, x_{12}^{(4)}b_{22}^{(1)}, x_{12}^{(4)}b_{22}^{(4)}\right), \\
\mu_1 &= min\left(x_{21}^{(1)}b_{11}^{(1)}, x_{21}^{(1)}b_{11}^{(4)}, x_{21}^{(4)}b_{11}^{(1)}, x_{21}^{(4)}b_{11}^{(4)}\right) + min\left(x_{22}^{(1)}b_{21}^{(1)}, x_{22}^{(1)}b_{21}^{(4)}, x_{22}^{(4)}b_{21}^{(1)}, x_{22}^{(4)}b_{21}^{(4)}\right), \\
\sigma_1 &= min\left(x_{21}^{(1)}b_{12}^{(1)}, x_{21}^{(1)}b_{12}^{(4)}, x_{21}^{(4)}b_{12}^{(1)}, x_{21}^{(4)}b_{12}^{(4)}\right) + min\left(x_{22}^{(1)}b_{22}^{(1)}, x_{22}^{(1)}b_{22}^{(4)}, x_{22}^{(4)}b_{22}^{(1)}, x_{22}^{(4)}b_{22}^{(4)}\right), \\
\gamma_2 &= min\left(x_{11}^{(2)}b_{11}^{(2)}, x_{11}^{(2)}b_{11}^{(3)}, x_{11}^{(3)}b_{11}^{(2)}, x_{11}^{(3)}b_{11}^{(3)}\right) + min\left(x_{12}^{(2)}b_{21}^{(2)}, x_{12}^{(2)}b_{21}^{(3)}, x_{12}^{(3)}b_{21}^{(2)}, x_{12}^{(3)}b_{21}^{(3)}\right), \\
\delta_2 &= min\left(x_{11}^{(2)}b_{12}^{(2)}, x_{11}^{(2)}b_{12}^{(3)}, x_{11}^{(3)}b_{12}^{(2)}, x_{11}^{(3)}b_{12}^{(3)}\right) + min\left(x_{12}^{(2)}b_{22}^{(2)}, x_{12}^{(2)}b_{22}^{(3)}, x_{12}^{(3)}b_{22}^{(2)}, x_{12}^{(3)}b_{22}^{(3)}\right), \\
\mu_2 &= min\left(x_{21}^{(2)}b_{11}^{(2)}, x_{21}^{(2)}b_{11}^{(3)}, x_{21}^{(3)}b_{11}^{(2)}, x_{21}^{(3)}b_{11}^{(3)}\right) + min\left(x_{22}^{(2)}b_{21}^{(2)}, x_{22}^{(2)}b_{21}^{(3)}, x_{22}^{(3)}b_{21}^{(2)}, x_{22}^{(3)}b_{21}^{(3)}\right), \\
\sigma_2 &= min\left(x_{21}^{(2)}b_{12}^{(2)}, x_{21}^{(2)}b_{12}^{(3)}, x_{21}^{(3)}b_{12}^{(2)}, x_{21}^{(3)}b_{12}^{(3)}\right) + min\left(x_{22}^{(2)}b_{22}^{(2)}, x_{22}^{(2)}b_{22}^{(3)}, x_{22}^{(3)}b_{22}^{(2)}, x_{22}^{(3)}b_{22}^{(3)}\right), \\
\gamma_3 &= max\left(x_{11}^{(2)}b_{11}^{(2)}, x_{11}^{(2)}b_{11}^{(3)}, x_{11}^{(3)}b_{11}^{(2)}, x_{11}^{(3)}b_{11}^{(3)}\right) + max\left(x_{12}^{(2)}b_{21}^{(2)}, x_{12}^{(2)}b_{21}^{(3)}, x_{12}^{(3)}b_{21}^{(2)}, x_{12}^{(3)}b_{21}^{(3)}\right), \\
\delta_3 &= max\left(x_{11}^{(2)}b_{12}^{(2)}, x_{11}^{(2)}b_{12}^{(3)}, x_{11}^{(3)}b_{12}^{(2)}, x_{11}^{(3)}b_{12}^{(3)}\right) + max\left(x_{12}^{(2)}b_{22}^{(2)}, x_{12}^{(2)}b_{22}^{(3)}, x_{12}^{(3)}b_{22}^{(2)}, x_{12}^{(3)}b_{22}^{(3)}\right), \\
\mu_3 &= max\left(x_{21}^{(2)}b_{11}^{(2)}, x_{21}^{(2)}b_{11}^{(3)}, x_{21}^{(3)}b_{11}^{(2)}, x_{21}^{(3)}b_{11}^{(3)}\right) + max\left(x_{22}^{(2)}b_{21}^{(2)}, x_{22}^{(2)}b_{21}^{(3)}, x_{22}^{(3)}b_{21}^{(2)}, x_{22}^{(3)}b_{21}^{(3)}\right), \\
\sigma_3 &= max\left(x_{21}^{(2)}b_{12}^{(2)}, x_{21}^{(2)}b_{12}^{(3)}, x_{21}^{(3)}b_{12}^{(2)}, x_{21}^{(3)}b_{12}^{(3)}\right) + max\left(x_{22}^{(2)}b_{22}^{(2)}, x_{22}^{(2)}b_{22}^{(3)}, x_{22}^{(3)}b_{22}^{(2)}, x_{22}^{(3)}b_{22}^{(3)}\right), \\
\gamma_4 &= max\left(x_{11}^{(1)}b_{11}^{(1)}, x_{11}^{(1)}b_{11}^{(4)}, x_{11}^{(4)}b_{11}^{(1)}, x_{11}^{(4)}b_{11}^{(4)}\right) + max\left(x_{12}^{(1)}b_{21}^{(1)}, x_{12}^{(1)}b_{21}^{(4)}, x_{12}^{(4)}b_{21}^{(1)}, x_{12}^{(4)}b_{21}^{(4)}\right), \\
\delta_4 &= max\left(x_{11}^{(1)}b_{12}^{(1)}, x_{11}^{(1)}b_{12}^{(4)}, x_{11}^{(4)}b_{12}^{(1)}, x_{11}^{(4)}b_{12}^{(4)}\right) + max\left(x_{12}^{(1)}b_{22}^{(1)}, x_{12}^{(1)}b_{22}^{(4)}, x_{12}^{(4)}b_{22}^{(1)}, x_{12}^{(4)}b_{22}^{(4)}\right), \\
\mu_4 &= max\left(x_{21}^{(1)}b_{11}^{(1)}, x_{21}^{(1)}b_{11}^{(4)}, x_{21}^{(4)}b_{11}^{(1)}, x_{21}^{(4)}b_{11}^{(4)}\right) + max\left(x_{22}^{(1)}b_{21}^{(1)}, x_{22}^{(1)}b_{21}^{(4)}, x_{22}^{(4)}b_{21}^{(1)}, x_{22}^{(4)}b_{21}^{(4)}\right)
\end{aligned}$$

and,

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

Step 3. Adding Equations (16) and (17), we obtain the following:

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

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

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

Step 5. The obtained Equation (18) can be converted into the following system of 16 equations. It is 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 obtain 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, Equation (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, Equation (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 most of the time are 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 a solution that is 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 $\widetilde{X}$ Equation (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 since 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.

Applying 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 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 aims to solve the system gradually by considering all the combinations within this narrow range.

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 search region for each of the four independent variables. This can be seen as searching for an average value x_{ij}^{Avg} of each of the main four vectors in Equation (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 Equation (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 four unknowns $\left(\widetilde{X}_{11}^{Avg}, \widetilde{X}_{21}^{Avg}, \widetilde{X}_{12}^{Avg}, \widetilde{X}_{22}^{Avg}\right)$. In other words, the fuzzy unknown numbers are considered crisp while keeping the same fuzzy multiplication operations.
3. Limiting those variables to four 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.

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:

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

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

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 systems; each is composed of 8 equations. 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 to significantly reduce the computational time for finding solutions. 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 two systems of six 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 solved using the two suggested methods. The first one uses the MATLAB built-in function "fsolve", and the second one uses the two sub-algorithms.

Example 2. *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) Solution using the newly developed algorithm.
The algorithm proposes the application of two 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

$$x_{11}^{avg} = -31\,;\; x_{21}^{avg} = 15\,;\; x_{12}^{avg} = 7\,;\; x_{22}^{avg} = -100$$

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:

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

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

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

And

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

The following Figure 6 shows the arbitrary fuzzy solutions $\widetilde{X}_1$, $\widetilde{X}_2$, $\widetilde{X}_3$ and $\widetilde{X}_4$.

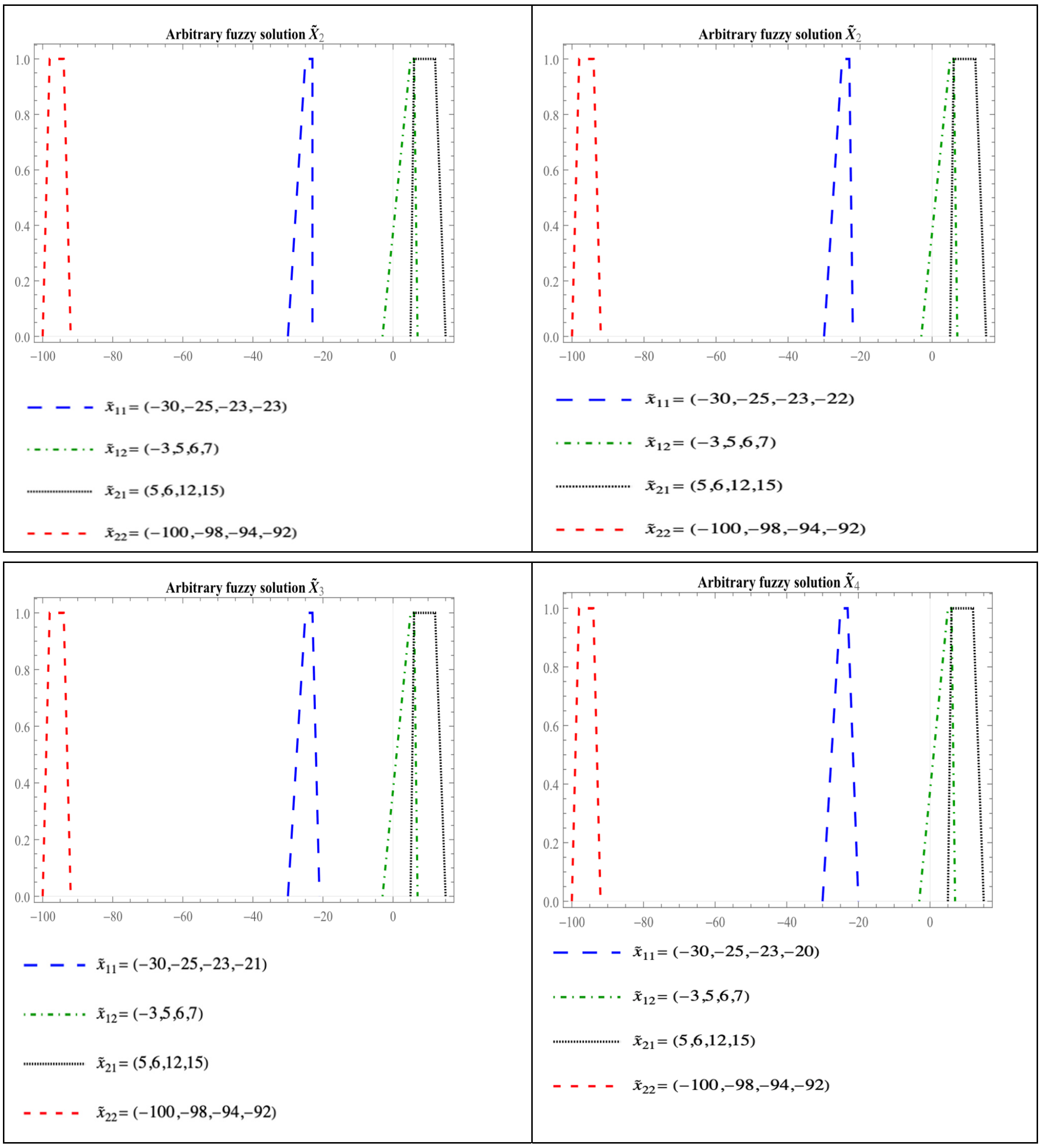

Figure 6. Arbitrary fuzzy solutions $\widetilde{X}_1$, $\widetilde{X}_2$, $\widetilde{X}_3$ and $\widetilde{X}_4$ of Example 2.

(II) Solution using MATLAB and the function FSOLVE

There exists a package in MATLAB by Zalatko Zahariev [29]. However, by applying the "FSOLVE" algorithm built-in MATLAB directly for the non-linear system over the 16 equations, it gives the solution below:

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

Substituting this solution into Equation (1) does not satisfy the equation with the proposed matrix $\widetilde{C}$.

Adding the constraints $x_{ij}^{(1)} \leq x_{ij}^{(2)} \leq x_{ij}^{(3)} \leq x_{ij}^{(4)}$, it also does not lead 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 was found to be a useful alternative to the build-in function FSOLVE inherent to the MATLAB software. However, due to the reason that 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 the 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* $\widetilde{A}\widetilde{X} + \widetilde{X}\widetilde{B} = \widetilde{C}$ *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* $\widetilde{A}\widetilde{X} = \widetilde{C}$ *with TrFNs and TFNs if we allow* $\widetilde{B} = 0$*, in* $\widetilde{A}\widetilde{X} + \widetilde{X}\widetilde{B} = \widetilde{C}$*. 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. In the following Example 3, we resolve Example 1 in [44].*

Example 3. *Consider the following Arbitrary FFME*:

$$\begin{pmatrix} (-2,3,3,4) & (-2,2,2,3) \\ (1,2,2,2) & (4,4,4,5) \end{pmatrix} \begin{pmatrix} \left(x_{11}^{(1)}, x_{11}^{(2)}, x_{11}^{(3)}, x_{11}^{(4)}\right) \\ \left(x_{21}^{(1)}, x_{21}^{(2)}, x_{21}^{(3)}, x_{21}^{(4)}\right) \end{pmatrix} = \begin{pmatrix} (-13,8,8,14) \\ (-14,8,8,14) \end{pmatrix}$$

Solution: The fuzzy solution to the given arbitrary FFME is obtained by Kumar et al. [44] as follows:

$$\widetilde{X} = \begin{pmatrix} (1,\ 2,\ 2,\ 2) \\ (-3,\ 1,\ 1,\ 2) \end{pmatrix}.$$

However, Malkawi et al. [45] were able to obtain two fuzzy solutions as follows:

$$\widetilde{X} = \left\{ \begin{pmatrix} (1,\ 2,\ 2,\ 2) \\ (-3,\ 1,\ 1,\ 2) \end{pmatrix}, \begin{pmatrix} (-23/14,\ 2,\ 2,\ 2) \\ (-15/7, 1,\ 1,\ 2) \end{pmatrix} \right\}.$$

Applying the two-stage algorithm to this arbitrary FFME gives strong fuzzy solutions. Table 1 gives all possible feasible fuzzy solutions to Example 3.

Table 1. Fuzzy solutions of Example 3.

Fuzzy Solutions	Absolute Error
$\widetilde{X}_1 = \begin{pmatrix} (1,\ 2,\ 2,\ 2) \\ (-3,\ 1,\ 1,\ 2) \end{pmatrix}$	0
$\widetilde{X}_2 = \begin{pmatrix} (-1,\ 2,\ 2,\ 2) \\ (-3,\ 1,\ 1,\ 2) \end{pmatrix}$	0
$\widetilde{X}_3 = \begin{pmatrix} (2,\ 2,\ 2,\ 2) \\ (-3,\ 1,\ 1,\ 2) \end{pmatrix}$	5.5×10^{-27}
$\widetilde{X}_4 = \begin{pmatrix} (-2,\ 2,\ 2,\ 2) \\ (0,\ 1,\ 1,\ 2) \end{pmatrix}$	1.57×10^{-8}
$\widetilde{X}_5 = \begin{pmatrix} (0,\ 2,\ 2,\ 2) \\ (-3,\ 1,\ 1,\ 2) \end{pmatrix}$	0

The two-stage algorithm method is not only able to detect all possible fuzzy solutions but also able to find all possible non-fuzzy solutions. The following Table 2 gives the weak fuzzy solution to Example 3.

Table 2. Non-Fuzzy solutions of Example 3.

Non-Fuzzy Solutions	Absolute Error
$\widetilde{X}_1 = \begin{pmatrix} (2,\ 2,\ 2,-1) \\ (2,\ 1,\ 1,-3) \end{pmatrix}$	7.04×10^{-68}
$\widetilde{X}_2 = \begin{pmatrix} (-1,\ 2,\ 2,\ 2) \\ (2,\ 1,\ 1,-3) \end{pmatrix}$	0
$\widetilde{X}_3 = \begin{pmatrix} (2,\ 2,\ 2,\ 0) \\ (-3,\ 1,\ 1,\ 2) \end{pmatrix}$	0
$\widetilde{X}_4 = \begin{pmatrix} (2,\ 2,\ 2,\ 0) \\ (2,\ 1,\ 1,-3) \end{pmatrix}$	0
$\widetilde{X}_5 = \begin{pmatrix} (0,\ 2,\ 2,\ 2) \\ (2,\ 1,\ 1,-3) \end{pmatrix}$	0

6. Conclusions

In this paper, the solution of the arbitrary TrFFSME $\widetilde{A}\widetilde{X} + \widetilde{X}\widetilde{B} = \widetilde{C}$ is obtained numerically using newly developed two-stage algorithm. In solving TrFFSME, fuzzy arithmetic multiplication for TrFNs is constructed. This method is superior compared to the existing methods since it eliminates the restrictions to the parameters of the TrFFSME and is able to obtain all possible finite fuzzy and non-fuzzy solutions. Furthermore, the proposed method is applicable for solving arbitrary TrFFSME and can also solve arbitrary FFME with TFNs and TrFNs. The main limitation of the proposed method is that it needs a long computational time and, therefore, large memory storage in order to find all possible solutions. In future work, a further modification to the AMO is needed in order to reduce the complexity of the min-max non-linear system. In addition, optimization techniques need to be developed to overcome the limitation of this method.

Author Contributions: Conceptualization, A.A.A.E. and B.S.; methodology, A.A.A.E. and B.S.; software, B.S.; validation, N.A. and G.M.; formal analysis, A.A.A.E. and B.S.; investigation, A.A.A.E. and B.S.; resources, A.A.A.E.; data curation, A.A.A.E. and B.S.; writing—original draft preparation, A.A.A.E. and B.S.; writing—review and editing, A.A.A.E. and B.S.; visualization, A.A.A.E. and B.S.; supervision, N.A. and G.M.; project administration, N.A. and G.M.; funding acquisition, B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Abu Dhabi Polytechnic College.

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

Conflicts of Interest: The authors declare no conflict of interest, financial or otherwise.

Appendix A Example of Solving System Numerically Using All Possible Integer Combinations

In order to solve numerically a system of four equations with four unknowns $f(X1, X2, X3, X4) = 0$ under the assumptions of integer variables within a range of 10 numbers (i.e., $X_i \in [X_i, Xi + 9]$), below are the possible combinations.

The possible solutions can be any line of Table A1 below. The table represents all the possible combinations of variables.

Table A1. Permutations of four variables with 10 possible integer values each.

*X*1	*X*2	*X*3	*X*4
$x1$	$x2$	$x3$	$x4$
$x1$	$x2$	$x3$	$x4 + 1$
...	...	...	...
$x1$	$x2$	$x3$	$x4 + 9$
$x1$	$x2$	$x3 + 1$	$x4$
...	...	$x3 + 1$	$x4 + 1$
...	...	...	...
$x1$	$x2$	$x3 + 9$	$x4 + 9$
...	$x2 + 1$	$x3$	$x4$
...	...	...	...
$x1$	$x2 + 9$	$x3 + 9$	$x4 + 9$
$x1 + 1$	$x2$	$x3$	$x4$
...			
$x1 + 9$	$x2 + 9$	$x3 + 9$	$x4 + 9$

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 gradually considering the equations and their potential solutions. For example, the first equation in the first-stage algorithm has three 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 fourth 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)).

$$\begin{cases} m_1 + \gamma_1 = c_{11}^{(1)} & (A1a) \\ n_1 + \delta_1 = c_{12}^{(1)} & (A1b) \\ \alpha_1 + \mu_1 = c_{21}^{(1)} & (A1c) \\ \beta_1 + \sigma_1 = c_{22}^{(1)} & (A1d) \\ \\ m_4 + \gamma_4 = c_{11}^{(4)} & (A1e) \\ n_4 + \delta_4 = c_{12}^{(4)} & (A1f) \\ \alpha_4 + \mu_4 = c_{21}^{(4)} & (A1g) \\ \beta_4 + \sigma_4 = c_{22}^{(4)} & (A1h) \end{cases}$$

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

Appendix C Relaxing Constraints of Equations Into Inequations

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 a perfectly normal occurrence due to considering the fuzzy variables as crisp. Therefore, the system of equations to be solved by the first-stage algorithm is relaxed into inequalities. In other words, the equations in Appendix B are transformed into the eight inequalities below:

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

The value of ε is calibrated in a way to obtain one or a 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 εwill not allow finding solutions in most cases.

Appendix D Solving the 8 Equations with Superscripts (1) and (4)

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

$$\begin{cases} m_1 + \gamma_1 = c_{11}^{(1)} & (A2a) \\ n_1 + \delta_1 = c_{12}^{(1)} & (A2b) \\ \alpha_1 + \mu_1 = c_{21}^{(1)} & (A2c) \\ \beta_1 + \sigma_1 = c_{22}^{(1)} & (A2d) \\ m_4 + \gamma_4 = c_{11}^{(4)} & (A2e) \\ n_4 + \delta_4 = c_{12}^{(4)} & (A2f) \\ \alpha_4 + \mu_4 = c_{21}^{(4)} & (A2g) \\ \beta_4 + \sigma_4 = c_{22}^{(4)} & (A2h) \end{cases}$$

The eight 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 displayed in Table A2 as follows:

Table A2. Combinations under constraints $x_{11}^{(1)} \leq x_{11}^{(4)}$.

$\mathbf{X_{11}^{(1)}}$	$\mathbf{X_{11}^{(4)}}$
$x_{11}^{(1)}$	$x_{11}^{(1)}$
$x_{11}^{(1)}$	$x_{11}^{(1)} + 1$
…	…
$x_{11}^{(1)}$	$x_{11}^{(1)} + 9$
$x_{11}^{(1)} + 1$	$x_{11}^{(1)} + 1$
$x_{11}^{(1)} + 1$	$x_{11}^{(1)} + 2$
…	…
$x_{11}^{(1)} + 1$	$x_{11}^{(1)} + 9$
$x_{11}^{(1)} + 2$	$x_{11}^{(1)} + 2$
…	…
$x_{11}^{(1)} + 7$	$x_{11}^{(1)} + 9$
$x_{11}^{(1)} + 8$	$x_{11}^{(1)} + 8$
$x_{11}^{(1)} + 8$	$x_{11}^{(1)} + 9$
$x_{11}^{(1)} + 9$	$x_{11}^{(1)} + 9$

1. The sub-algorithm considers Equations (A2a) and (A2e), which include the six 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 Equation (A2a).
2. The retained combinations are passed to Equation (A2a). Only those combinations satisfying Equations (A2a) and (A2e) are saved in a set called S_{15}, having N_{15} combinations.

3. The set S_{15} is combined with all combinations of variables $x_{22}^{(1)}, x_{22}^{(4)}$. This set called $S_{8\text{var}}$ contains all potential solutions of the system of eight equations. The number of combinations in this case becomes: $N_{8var} = 55 \times N_{15}$.
4. Those combinations are then filtered by Equation (A2b) then Equation (A2f).
5. Passed and filtered after by Equation (A2c) and Equation (A2g)
6. And lastly passed to Equation (A2d) and Equation (A2h).

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

References

1. Datta, B. *Numerical Methods for Linear Control Systems*; Academic Press: London, UK, 2004; Volume 1, ISBN 0122035909.
2. Sorensen, D.C.; Antoulas, A.C. The Sylvester equation and approximate balanced reduction. *Linear Algebra Appl.* **2002**, *351*, 671–700. [CrossRef]
3. Bouhamidi, A.; Jbilou, K. Sylvester Tikhonov-regularization methods in image restoration. *J. Comput. Appl. Math.* **2007**, *206*, 86–98. [CrossRef]
4. Lukasiewicz, J. On three-valued logic. *Ruch Filoz.* **1920**, *5*, 170–171.
5. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
6. Johnson, P.E. The Genesis and Development of Set Theory. *Two-Year Coll. Math. J.* **1972**, *3*, 55. [CrossRef]
7. Sanchez, E. Resolution of composite fuzzy relation equations. *Inf. Control* **1976**, *30*, 38–48. [CrossRef]
8. De Baets, B. *Analytical Solution Methods for Fuzzy Relational Equations*; Springer: Boston, MA, USA, 2000; pp. 291–340.
9. Di Nola, A.; Sessa, S.; Pedrycz, W.; Sanchez, E. *Fuzzy Relation Equations and Their Applications to Knowledge Engineering*; Springer: Dordrecht, The Netherlands, 1989; ISBN 9789048140503.
10. Cechlárová, K. A note on unsolvable systems of max-min (fuzzy) equations. *Linear Algebra Appl.* **2000**, *310*, 123–128. [CrossRef]
11. Chen, L.; Wang, P.P. Fuzzy relation equations (I): The general and specialized solving algorithms. *Soft Comput. A Fusion Found. Methodol. Appl.* **2002**, *6*, 428–435. [CrossRef]
12. Peeva, K. Resolution of fuzzy relational equations—Method, algorithm and software with applications. *Inf. Sci.* **2013**, *234*, 44–63. [CrossRef]
13. Yeh, C.-T. On the minimal solutions of max–min fuzzy relational equations. *Fuzzy Sets Syst.* **2008**, *159*, 23–39. [CrossRef]
14. Shieh, B.-S. New Resolution of Finite Fuzzy Relation Equations with Max-Min Composition. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2008**, *16*, 19–33. [CrossRef]
15. Bourke, M.M.; Fisher, D.G. Solution algorithms for fuzzy relational equations with max-product composition. *Fuzzy Sets Syst.* **1998**, *94*, 61–69. [CrossRef]
16. Loetamonphong, J.; Fang, S.-C. An efficient solution procedure for fuzzy relation equations with max-product composition. *IEEE Trans. Fuzzy Syst.* **1999**, *7*, 441–445. [CrossRef]
17. Loetamonphong, J.; Fang, S.-C. Optimization of fuzzy relation equations with max-product composition. *Fuzzy Sets Syst.* **2001**, *118*, 509–517. [CrossRef]
18. Peeva, K.; Kyosev, Y. Algorithm for Solving Max-product Fuzzy Relational Equations. *Soft Comput.* **2007**, *11*, 593–605. [CrossRef]
19. Wu, Y.-K.; Guu, S.-M. An Efficient Procedure for Solving a Fuzzy Relational Equation With Max–Archimedean t-Norm Composition. *IEEE Trans. Fuzzy Syst.* **2008**, *16*, 73–84. [CrossRef]
20. Lin, J.-L. On the relation between fuzzy max-Archimedean t-norm relational equations and the covering problem. *Fuzzy Sets Syst.* **2009**, *160*, 2328–2344. [CrossRef]
21. Lin, J.-L.; Wu, Y.-K.; Guu, S.-M. On fuzzy relational equations and the covering problem. *Inf. Sci.* **2011**, *181*, 2951–2963. [CrossRef]
22. Shieh, B.-S. Solutions of fuzzy relation equations based on continuous t-norms. *Inf. Sci.* **2007**, *177*, 4208–4215. [CrossRef]
23. Khorram, E.; Ghodousian, A. Linear objective function optimization with fuzzy relation equation constraints regarding max–av composition. *Appl. Math. Comput.* **2006**, *173*, 872–886. [CrossRef]
24. Wu, Y.-K. Optimization of fuzzy relational equations with max-av composition. *Inf. Sci.* **2007**, *177*, 4216–4229. [CrossRef]
25. Pappis, C.P.; Sugeno, M. Fuzzy relational equations and the inverse problem. *Fuzzy Sets Syst.* **1985**, *15*, 79–90. [CrossRef]
26. Li, P.; Fang, S.-C. On the resolution and optimization of a system of fuzzy relational equations with sup-T composition. *Fuzzy Optim. Decis. Mak.* **2008**, *7*, 169–214. [CrossRef]
27. Loetamonphong, J.; Fang, S.-C.; Young, R.E. Multi-objective optimization problems with fuzzy relation equation constraints. *Fuzzy Sets Syst.* **2002**, *127*, 141–164. [CrossRef]
28. Perfilieva, I.; Nosková, L. System of fuzzy relation equations with inf- Composition: Complete set of solutions. *Fuzzy Sets Syst.* **2008**, *159*, 2256–2271. [CrossRef]

29. Zlatko Zahariev. Fuzzy Calculus Core FC2ore-File Exchange-MATLAB Central. Available online: https://www.mathworks.com/matlabcentral/fileexchange/27046-fuzzy-calculus-core-fc2ore?s_tid=srchtitle_fuzzy-calculuscore-fc2ore._1 (accessed on 29 January 2022).
30. Guo, X. Bin Approximate solution of fuzzy Sylvester matrix equations. In Proceedings of the 7th International Conference on Computational Intelligence and Security, Sanya, China, 3–4 December 2011; pp. 52–56. [CrossRef]
31. Keyanpour, M.; Salkuyeh, D.K.; Moosaei, H.; Ketabchi, S. On the solution of the fully fuzzy Sylvester matrix equation. *Int. J. Model. Simul.* **2020**, *40*, 80–85. [CrossRef]
32. He, Q.; Hou, L.; Zhou, J. The solution of fuzzy Sylvester matrix equation. *Soft Comput.* **2018**, *22*, 6515–6523. [CrossRef]
33. Dubois, D.; Prade, H. Operations on fuzzy numbers. *Int. J. Syst. Sci.* **1978**, *9*, 613–626. [CrossRef]
34. Guo, X.; Shang, D. Fuzzy Approximate Solution of Positive Fully Fuzzy Linear Matrix Equations. *J. Appl. Math.* **2013**, *2013*, 178209. [CrossRef]
35. Malkawi, G.; Ahmad, N.; Ibrahim, H. Solving the fully fuzzy sylvester matrix equation with triangular fuzzy number. *Far East J. Math. Sci.* **2015**, *98*, 37–55. [CrossRef]
36. Malkawi, G.; Ahmad, N.; Ibrahim, H. Solving fully fuzzy linear system with the necessary and sufficient condition to have a positive solution. *Appl. Math. Inf. Sci.* **2014**, *8*, 1003–1019. [CrossRef]
37. Daud, S.; Ahmad, N.; Malkawi, G. Positive fuzzy minimal solution for positive singular fully fuzzy Sylvester matrix equation. *AIP Conf. Proc.* **2018**, *1974*, 020084. [CrossRef]
38. Daud, S.; Ahmad, N.; Malkawi, G. An algorithm for solving an arbitrary triangular fully fuzzy Sylvester matrix equations. *AIP Conf. Proc.* **2017**, *1905*, 030012. [CrossRef]
39. Dookhitram, K.; Lollchund, R.; Tripathi, R.K.; Bhuruth, M. Fully fuzzy Sylvester matrix equation. *J. Intell. Fuzzy Syst.* **2015**, *28*, 2199–2211. [CrossRef]
40. Daud, W.S.W.; Ahmad, N.; Malkawi, G. Positive Solution of Arbitrary Triangular Fully Fuzzy Sylvester Matrix Equations. *Far East J. Math. Sci.* **2018**, *103*, 271–298. [CrossRef]
41. Fortin, J.; Dubois, D.; Fargier, H. Gradual numbers and their application to fuzzy interval analysis. *IEEE Trans. Fuzzy Syst.* **2008**, *16*, 388–402. [CrossRef]
42. Lee, K.H. *First Course on Fuzzy Theory and Applications*; Springer Science & Business Media: Berlin, Germany, 2005; Volume 42, ISBN 3540229884.
43. Kaufmann, A.; Gupta, M.M.; Bohlender, G. Introduction to Fuzzy Arithmetic, Theory and Applications. *Math. Comput.* **1986**, *47*, 762. [CrossRef]
44. Kumar, A.; Neetu; Bansal, A. A new approach for solving fully fuzzy linear systems. *Adv. Fuzzy Syst.* **2011**, *2011*, 943161. [CrossRef]
45. Malkawi, G.; Ahmad, N.; Ibrahim, H. Finite solutions of fully fuzzy linear system. *AIP Conf. Proc.* **2014**, *1635*, 447–454. [CrossRef]