NEW MODELS AND ALGORITHMS FOR SOLUTIONS OF SINGLE-SIGNED FULLY FUZZY LR LINEAR SYSTEMS

R. EZZATI, S. KHEZERLOO, N. MAHDAVI-AMIRI AND Z. VALIZADEH

ABSTRACT. We present a model and propose an approach to compute an approximate solution of Fully Fuzzy Linear System ($FFLS$) of equations in which all the components of the coefficient matrix are either nonnegative or nonpositive. First, in discussing an $FFLS$ with a nonnegative coefficient matrix, we consider an equivalent $FFLS$ by using an appropriate permutation to simplify fuzzy multiplications. To solve the $m \times n$ permutated system, we convert it to three $m \times n$ real linear systems, one being concerned with the cores and the other two being related to the left and right spreads. To decide whether the core system is consistent or not, we use the modified Huang algorithm of the class of ABS methods. If the core system is inconsistent, an appropriate unconstrained least squares problem is solved for an approximate solution. The sign of each component of the solution is decided by the sign of its core. Also, to know whether the left and right spread systems are consistent or not, we apply the modified Huang algorithm again. Appropriate constrained least squares problems are solved, when the spread systems are inconsistent or do not satisfy fuzziness conditions. Then, we consider the $FFLS$ with a mixed single-signed coefficient matrix, in which each component of the coefficient matrix is either nonnegative or nonpositive. In this case, we break the $m \times n$ coefficient matrix up to two $m \times n$ matrices, one having only nonnegative and the other having only nonpositive components, such that their sum yields the original coefficient matrix. Using the distributive law, we convert each $m \times n$ $FFLS$ into two real linear systems where the first one is related to the cores with size $m \times n$ and the other is $2m \times 2n$ and is related to the spreads. Here, we also use the modified Huang algorithm to decide whether these systems are consistent or not. If the first system is inconsistent or the second system does not satisfy the fuzziness conditions, we find an approximate solution by solving a respective least squares problem. We summarize the proposed approach by presenting two computational algorithms. Finally, the algorithms are implemented and effectively tested by solving various randomly generated consistent as well as inconsistent numerical test problems.

1. Introduction

Systems of linear equations arise from various areas of science and engineering. Since many real-world systems are too complex to be defined in precise terms, imprecision is often involved. Analyzing such systems requires the use of fuzzy information. Therefore, the fuzzy concept proposed by Zadeh [29, 30] is deemed

Received: October 2010; Revised: May 2011; Accepted: August 2011

Key words and phrases: LR fuzzy numbers, Single-signed fuzzy numbers, Fully fuzzy linear systems, ABS algorithms, Least squares problems.

to be quite useful in many applications. Thus, the need for solving linear systems whose parameters are all or partially represented by fuzzy numbers, is apparent.
A general model for solving an arbitrary square fuzzy linear system whose coefficient matrix is crisp and the right-hand side column is an arbitrary fuzzy vector, was first proposed by Friedman et al. [21]. For computing a solution, they used the embedding method and replaced the original fuzzy $n \times n$ linear system by a $2n \times 2n$ crisp linear system.
Other numerical procedures for solving fuzzy linear systems such as Jacobi, Gauss-Seidel, Adomiam decomposition method and SOR iterative method can be seen in [6, 8, 9, 10].
In [13], Dehghan et al. also proposed some iterative techniques. Abbasbandy et al. [5] solved $m \times n$ $(m \leq n)$ original fuzzy linear system using a $2m \times 2n$ real system, and Allahviranloo et al. [7] studied finding least squares solution of an overdetermined $(m > n)$ fuzzy linear system.
In the more general system of fully fuzzy linear system of equations ($FFLSE$), both the coefficient matrix and the right hand sides are considered to be fuzzy and a fuzzy solution is desired. A conceivable method for solving an $FFLSE$ is to consider real systems corresponding to α-cuts whose coefficient matrices and right-hand side vectors, A and b (with corresponding positive membership degrees), are extracted from the fuzzy coefficient matrix $\widetilde{A} = \left(\widetilde{a}_{ij} \right)$ and fuzzy right hand side vector $\widetilde{b} = \left(\widetilde{b}_i \right)$, respectively. Buckley and Qu [12] gave a method to compute the set of solutions. For A nonsingular, $x = A^{-1}b$ with a positive membership, is a solution of the corresponding system and

$$\{x|\ x = A^{-1}b, A = \left(a_{ij} \right), b = \left(b_i \right), a_{ij} \in supp(\widetilde{a}_{ij}), b_i \in supp(\widetilde{b}_i)\ for\ all\ i, j\}$$

is the set of solutions where $supp(.)$ stands for the closure of the union of all α-cuts, $\alpha > 0$. Based on this fact, they discussed the theoretical aspects of this class of problems, extended several methods for this class and proved their equivalence. But their approach is not practical, since it may require trying a large number of α-cuts and solving large numbers of systems, without having a guarantee for establishing a solution for the fully fuzzy system.
In recent years, Dehghan et al. [16] paid some attention to solve such systems. They applied some well-known iterative methods such as Jacobi, Gauss-Seidel, SOR, AOR on the $FFLSE$. Moreover, they focused on the use of Adomian decomposition method for solving the $FFLSE$ [14]. They also used classical methods like Cramer's rule, Guassian elimination and LU decomposition to find the approximate solution [15]. To guarantee that the produced solution would be a fuzzy number, at times they applied linear programming models.
Here, we suppose that the components of the coefficient matrix of the system, $\widetilde{A}$, are single-signed LR fuzzy numbers. We intend to compute a fuzzy vector, $\widetilde{x}$, with single-signed components satisfying $\widetilde{A} \otimes \widetilde{x} = \widetilde{b}$, where $\widetilde{b}$ is a single-signed fuzzy column vector.
First, we order the elements in terms of their signs by using proper permutation matrices, and represent the original system by an equivalent fully fuzzy system in

which multiplying the coefficient matrix by the variable vector is simpler than the multiplications required in the original form. Then, to solve the equivalent system, first an appropriate sign is assigned to each variable considering the locus of its core; afterwards, by employing the Dubois and Prade's approximate arithmetic operators on LR fuzzy numbers [17], and multiplying the permutated coefficient matrix by the permuted vector, the system will be reformulated by three real linear systems, one relating to the cores that was solved before and the others corresponding to the left and right spreads.

There are several approaches to solve real linear systems of equations in the literature. These systems can be solved, although inefficiently, by using the inverse or pseudinverse of the coefficient matrices [16, 21]. They may also be solved by some direct methods, such as various algorithms in the ABS class [1, 2] and different decompositions [14] (note that decomposition methods also belong to the ABS class). Using ABS algorithms to solve linear systems have several advantages. Since by applying an ABS method one obtains all solutions of the system, then at the presence of more than one solution, a specific solution of interest may be accessible by using a particular member of the class. In addition, practical problems may contain dependent equations, and the ABS algorithms can readily determine wether the system is consistent or not, and when consistent, eliminate the dependent equations. Additionally, if at least one of the real systems lacks solution or its solution does not satisfy the fuzziness conditions, then we are able to find an appropriate approximate solution by solving a constrained least squares problem.

ABS class of algorithms was proposed in 1984 by Abbafy, Broyden and Spedicato [1] to solve linear systems and later extended to solve various problems such as linear least squares, nonlinear equations, optimization problems [2] and recently Diophantine equations [18, 25, 26]. The class of ABS methods unifies the existing direct methods for solving linear systems and supplies a variety of alternative ways for implementing a specific algorithm. Extensive computational experience has shown that the ABS methods are implementable in a stable way, being often more accurate than the corresponding traditional algorithms; for review on ABS algorithms and discussion on their computational results, see [28]. In our approach, we find an appropriate approximate solution by solving a constrained least squares problem using the ABS class of algorithms.

Some basic definitions and basic steps for the unscaled ABS algorithms are presented in sections 2 and 3, respectively. Section 4 investigates the solution for the case in which $\widetilde{A}$ is a positive fuzzy matrix and $\widetilde{b}$ is a fuzzy vector. In section 5, we consider general fully fuzzy systems in which the components of $\widetilde{A}$ are single-signed fuzzy numbers. In section 6, some numerical results are provided. Finally, section 7 gives our concluding remarks.

2. Preliminaries

The following basic definitions for fuzzy numbers can be found in [24, 31].

Definition 2.1. A fuzzy number is a fuzzy set like $\mu_{\widetilde{N}} : \Re \rightarrow [0, 1]$ which satisfies:

1. $\mu_{\widetilde{N}}$ is upper semi-continuous,

2. $\mu_{\widetilde{N}}(x) = 0$, outside some interval $[a, d] \subset \Re$,
3. There are real numbers b, c such that $a \leq b \leq c \leq d$ and
   (a) $\mu_{\widetilde{N}}(x)$ is monotonic increasing on $[a, b]$,
   (b) $\mu_{\widetilde{N}}(x)$ is monotonic decreasing on $[c, d]$,
   (c) $\mu_{\widetilde{N}}(x) = 1$, $b \leq x \leq c$.

We denote the set of all fuzzy numbers by F.

Definition 2.2. A fuzzy number $\widetilde{N}$ is called positive (negative), shown as $\widetilde{N} > 0$ ($\widetilde{N} < 0$), if its membership function $\mu_{\widetilde{N}}(x)$ satisfies $\mu_{\widetilde{N}}(x) = 0$, for all $x < 0$ (for all $x > 0$).

Remark 2.3. Of course, a fuzzy number could be neither positive nor negative.

Definition 2.4. A fuzzy number $\widetilde{N}$ is said to be a single-signed fuzzy number if $\widetilde{N}$ is either nonnegative or nonpositive.

Definition 2.5. A fuzzy number $\widetilde{N}$ is said to be an LR fuzzy number if

$$\mu_{\widetilde{N}}(x) = \begin{cases} L(\frac{n-x}{\alpha}), & x \leq n,\ \alpha > 0, \\ \\ R(\frac{x-n}{\beta}), & x \geq n,\ \beta > 0, \end{cases} \tag{1}$$

where n is the mean value of $\widetilde{N}$, α and β are left and right spreads, respectively, and the function $L(.)$, which is called left shape function, satisfies:

- $L(x) = L(-x)$,
- $L(0) = 1$ *and* $L(1) = 0$,
- $L(x)$ *is nonincreasing on* $[0, \infty)$.

The definition of a right shape function $R(.)$ is usually similar to that of $L(.)$.

The core, left and right spreads, and the shape function of an LR fuzzy number $\widetilde{N}$ are symbolically shown as:

$$\widetilde{N} = (n, \alpha, \beta)_{LR}.$$

Clearly, $\widetilde{N} = (n, \alpha, \beta)_{LR}$ is positive (negative), if and only if $n - \alpha > 0$ ($n + \beta < 0$), and $\widetilde{N} = (n, \alpha, \beta)_{LR}$ is called a triangular fuzzy number if and only if the corresponding left and right shape functions, $L(x)$ and $R(x)$, are linear functions. In [17], Dubois and Prade designed the following exact formulas for the addition of LR fuzzy numbers and scalar multiplication. They also introduced an approximate formula for multiplication of the LR fuzzy numbers. For two LR fuzzy numbers $\widetilde{M} = (m, \alpha, \beta)_{LR}$ and $\widetilde{N} = (n, \gamma, \delta)_{LR}$, we have:

- Addition:

$$\widetilde{M} \oplus \widetilde{N} = (m, \alpha, \beta)_{LR} \oplus (n, \gamma, \delta)_{LR} = (m + n, \alpha + \gamma, \beta + \delta)_{LR}. \tag{2}$$

- Multiplication (approximate):

  If $\widetilde{M} \geq 0$ and $\widetilde{N} \geq 0$, then

$$\widetilde{M} \otimes \widetilde{N} = (m, \alpha, \beta)_{LR} \otimes (n, \gamma, \delta)_{LR} \cong (mn, m\gamma + n\alpha, m\delta + n\beta)_{LR}. \tag{3}$$

If $\widetilde{M} \geq 0$ and $\widetilde{N} < 0$, then

$$\widetilde{M} \otimes \widetilde{N} = (m, \alpha, \beta)_{LR} \otimes (n, \gamma, \delta)_{LR} \cong (mn, m\gamma - n\beta, m\delta - n\alpha)_{RL}. \tag{4}$$

If $\widetilde{M} < 0$ and $\widetilde{N} \geq 0$, then

$$\widetilde{M} \otimes \widetilde{N} = (m, \alpha, \beta)_{LR} \otimes (n, \gamma, \delta)_{LR} \cong (mn, -m\delta + n\alpha, -m\gamma + n\beta)_{RL}. \tag{5}$$

If $\widetilde{M} < 0$ and $\widetilde{N} < 0$, then

$$\widetilde{M} \otimes \widetilde{N} = (m, \alpha, \beta)_{LR} \otimes (n, \gamma, \delta)_{LR} \cong (mn, -m\delta - n\beta, -m\gamma - n\alpha)_{LR}. \tag{6}$$

- Scalar multiplication:

$$\lambda \otimes \widetilde{M} = \lambda \otimes (m, \alpha, \beta)_{LR} = \begin{cases} (\lambda m, \lambda\alpha, \lambda\beta)_{LR}, & \lambda > 0, \\ (\lambda m, -\lambda\beta, -\lambda\alpha)_{LR}, & \lambda < 0. \end{cases} \tag{7}$$

Let $\Re$ be the set of all real numbers. We assume that a fuzzy number $\widetilde{N}$ can be expressed for all $x \in \Re$ in the form,

$$\widetilde{N}(x) = \begin{cases} g(x), \ when \ x \in [a, b), \\ 1, \ when \ x \in [b, c], \\ h(x), \ when \ x \in (c, d], \\ 0, \ otherwise, \end{cases} \tag{8}$$

where a, b, c, d are real numbers such that $a < b \leq c < d$ and g is a real valued function that is increasing and right continuous and h is a real valued function that is decreasing and left continuous. Notice that (8) is an LR fuzzy number. Here, we use a fuzzy matrix, a fuzzy linear system and its solution in the sense defined by Dehghan et al. [14] and Dubios et al. [17]. From now on, we shall use single-signed LR fuzzy numbers.

Definition 2.6. A matrix $\widetilde{A} = \left(\ \widetilde{a}_{ij} \ \right)$ is called a fuzzy matrix if each component of $\widetilde{A}$ is a fuzzy number. $\widetilde{A}$ will be positive (negative) and denoted by $\widetilde{A} > 0$ ($\widetilde{A} < 0$) if for all i,j, $\widetilde{a}_{ij} > 0$ ($\widetilde{a}_{ij} < 0$). Nonnegative and nonpositive fuzzy matrices will be defined similarly.

Definition 2.7. Let $\widetilde{A} = \left(\ \widetilde{a}_{ij} \ \right)$ and $\widetilde{B} = \left(\ \widetilde{b}_{ij} \ \right)$ be two $m \times n$ and $n \times p$ fuzzy matrices. The product $\widetilde{A} \otimes \widetilde{B} = \widetilde{C} = \left(\ \widetilde{c}_{ij} \ \right)_{m \times p}$ is defined to be:

$$\widetilde{c}_{ij} = \sum_{k=1,\ldots,n}^{\oplus} \widetilde{a}_{ik} \otimes \widetilde{b}_{kj}. \tag{9}$$

Definition 2.8. The $m \times n$ linear system of equations,

$$\begin{cases} (\widetilde{a}_{11} \otimes \widetilde{x}_1) \oplus (\widetilde{a}_{12} \otimes \widetilde{x}_2) \oplus \cdots \oplus (\widetilde{a}_{1n} \otimes \widetilde{x}_n) = \widetilde{b}_1 \\ (\widetilde{a}_{21} \otimes \widetilde{x}_1) \oplus (\widetilde{a}_{22} \otimes \widetilde{x}_2) \oplus \cdots \oplus (\widetilde{a}_{2n} \otimes \widetilde{x}_n) = \widetilde{b}_2 \\ \vdots \\ (\widetilde{a}_{m1} \otimes \widetilde{x}_1) \oplus (\widetilde{a}_{m2} \otimes \widetilde{x}_2) \oplus \cdots \oplus (\widetilde{a}_{mn} \otimes \widetilde{x}_n) = \widetilde{b}_m, \end{cases} \tag{10}$$

or in its matrix form,

$$\widetilde{A} \otimes \widetilde{x} = \widetilde{b}, \tag{11}$$

where $\widetilde{a}_{ij}$, $1 \leq i \leq m$, $1 \leq j \leq n$, the coefficient matrix components, and $\widetilde{b}_i$, $1 \leq i \leq m$, components of $\widetilde{b}$, are fuzzy numbers, is called Fully Fuzzy Linear System (FFLS), and the vector $\widetilde{x}$ is the unknown to be found.

Here, we assume that all parameters in (10) are LR fuzzy numbers, that is,

$$\begin{aligned} \widetilde{a}_{ij} &= (a_{ij}, m_{ij}, n_{ij})_{LR}, \quad 1 \leq i \leq m, \ 1 \leq j \leq n, \\ \widetilde{b}_i &= (b_i, b_{iL}, b_{iR})_{LR}, \qquad 1 \leq i \leq m, \end{aligned} \tag{12}$$

where m_{ij}, b_{iL} and n_{ij}, b_{iR} denote left and right spreads of $\widetilde{a}_{ij}$ and $\widetilde{b}_i$, respectively. Besides, we represent the fuzzy matrix $\widetilde{A} = \left(\ \widetilde{a}_{ij} \ \right)_{m\times n}$ and the vector $\widetilde{b} = \left(\ \widetilde{b}_i \ \right)_{m\times 1}$ by the notations,

$$\begin{aligned} \widetilde{A} &= (A, M, N), \\ \widetilde{b} &= (b, b_L, b_R), \end{aligned} \tag{13}$$

where $A = \left(\ a_{ij} \ \right)$, $M = \left(\ m_{ij} \ \right)$ and $N = \left(\ n_{ij} \ \right)$ are three $m \times n$ real matrices, and $b = \left(\ b_i \ \right)$, $b_L = \left(\ b_{iL} \ \right)$ and $b_R = \left(\ b_{iR} \ \right)$ are three m dimensional real vectors.

Definition 2.9. Suppose $\widetilde{x} = (x, y, z)$ solves (11). If y and z are nonnegative vectors, then $\widetilde{x}$ is called a strong fuzzy solution of (11). Otherwise, if y or z contains a negative component, then $\widetilde{x}$ is called a weak fuzzy solution.

3. **Basic ABS Algorithms**

Here, we recall basic steps of the unscaled ABS algorithms for solving the system of linear equations,

$$Ax = b, \tag{14}$$

where $A \in \Re^{m\times n}$, $b \in \Re^m$ and $x \in \Re^n$.
Let $A = \left(\ a_1, \ \ \ldots \ \ , a_m \ \right)^T$, $a_i \in \Re^n$, $i = 1, ..., m$ and $b = \left(\ b_1, \ \ \ldots \ \ , b_m \ \right)^T$. The details of the following algorithm can be found in [2]. We give the steps as specified in [22]. In Algorithm 3.1, r_i gives the rank of the first $i-1$ rows of A; hence, r_{m+1} gives the rank of A.

Algorithm 3.1. The unscaled ABS algorithm [2, 22].

(A) Choose $x_1 \in \Re^n$, an arbitrary vector, and $H_1 \in \Re^{n\times n}$, an arbitrary nonsingular matrix. Set $i = 1$, $r_1 = 0$.

(B) Compute the vector $s_i = H_i a_i$ and the scalar $\tau_i = a_i^T x_i - b_i$.

(C) **If** $s_i = 0$ and $\tau_i = 0$ **then** let $x_{i+1} = x_i$, $H_{i+1} = H_i$, $r_{i+1} = r_i$ and **go to** (H) (in this case, the *ith equation is a linear combination of the first* $i - 1$ *equations and thus can be removed.)*

else if $s_i = 0$ *and* $\tau_i \neq 0$ ***then stop*** *(the ith equation is incompatible and hence the system has no solution).*

(D) $\{s_i \neq 0\}$ Compute the search direction $p_i = H_i^T z_i$, where $z_i \in \Re^n$ is an arbitrary vector that satisfies $z_i^T H_i a_i = z_i^T s_i \neq 0$.

(E) Compute $\alpha_i = \frac{\tau_i}{a_i^T p_i}$ and update the estimate of the solution by

$$x_{i+1} = x_i - \alpha_i p_i.$$

(F) Update the matrix H_i to H_{i+1} by

$$H_{i+1} = H_i - H_i a_i w_i^T H_i,$$

where $w_i \in \Re^n$ is an arbitrary vector satisfying $w_i^T H_i a_i = w_i^T s_i = 1$.

(G) Let $r_{i+1} = r_i + 1$.

(H) **If** $i = m$ **then stop** (x_{m+1} solves the system) **else** let $i = i + 1$ and **go to** (B).

Remark 3.2. In step (B), $H_i a_i \neq 0$ if and only if a_i is linearly independent of $a_1, ..., a_{i-1}$; equivalently, $H_i a_i = 0$ if and only if a_i is linearly dependent on $a_1, ..., a_{i-1}$.

Remark 3.3. It can be shown that x_{i+1} is a solution for the first i equations and the columns of H_{i+1}^T span the null space of the first i rows of A; that is, $A_i^T H_{i+1}^T = 0$, where $A_i = \begin{pmatrix} a_1, & \dots & , a_i \end{pmatrix}$. Since r_{i+1} gives the rank of A_i, then the rank of H_{i+1} is equal to $n - r_{i+1}$.

Remark 3.4. At the beginning of the *ith iteration,* $i \geq 1$, *the general solution of the first* $i-1$ *equations is at hand. Note that with* x_i *as a solution of the first* $i-1$ *equations and with* $H_i^T \in \Re^{n\times n}$ *having* $rank(H_i) = n - r_i$, *as a generator of the null space of* A_{i-1}^T, *we have,*

$$x = x_i + H_i^T q,$$

with $q \in \Re^n$, *as the general solution of the first* $i-1$ *equations. Therefore, if the algorithm terminates at step* (H), *then the general solution of* (14) *is written as:*

$$x = x_{m+1} + H_{m+1}^T q,$$

where $q \in \Re^n$ *is arbitrary.*

Remark 3.5. Algorithm 3.1 gives a class of algorithms defined by specific choices of three parameters H_1, z_i, and w_i. For our problem, we use a subclass called modified Huang algorithm for its special properties. This solution is defined by the choices,

$$H_1 = I_n \ , \ z_i = H_i a_i \ , \ w_i = \frac{z_i}{z_i^T z_i}.$$

Below, we list certain properties of the modified Huang algorithm [2]:

(1) The direction search vectors $p_1, ..., p_i$ are pairwise orthogonal.

(2) In each iteration, the matrix H_i is symmetric. So, to save memory, only the diagonal and upper diagonal elements of H_i are needed to be stored.

(3) If we start with an arbitrary vector proportional to a_1, the sequence of iterates $x_2, ..., x_{m+1}$ generated by algorithm 3.1 has the minimal Euclidean norm among all solutions of the corresponding subsystems [2]. To simplify, choosing $x_1 = 0$, a 0-th multiple of a_1, is appropriate.

4. Solving the $FFLS$ with Nonnegative Coefficient Matrix

Here, we discuss the fully fuzzy systems of linear equations in which the components of the coefficient matrix are nonnegative LR fuzzy numbers. In the literature, several approaches have been discussed to find nonnegative fuzzy solutions of such systems. Here, we are to find a single-signed LR fuzzy solution (nonnegative or nonpositive). The sign of each fuzzy component is determined by the sign of the corresponding mean value. Note that according to the multiplication rules (3)-(6) for LR fuzzy numbers, the system of mean values corresponding to the solution of (11) is given by:

$$Ax = b, \tag{15}$$

independent of the signs of the components. Employing the modified Huang algorithm to solve system (15), we may face one of the following cases:

Case 1: The system has a unique solution x.

Case 2: The system has more than one solution (infinitely many solutions). We are able to find a specific solution by choosing proper parameters of the ABS method. For example, the solution with minimal Euclidean norm is obtained by choosing $x_1 = 0$ in the modified Huang algorithm (see section 3).

Case 3: The system is inconsistent. In this case, we will find an approximate solution for which the residual of the system has a minimal Euclidean norm by solving the following least squares problem,

$$\min_{x} \qquad \|Ax - b\|_2. \tag{16}$$

Here, we assume that if $x_i \geq 0$, for all i, then the corresponding fuzzy number is nonnegative; otherwise it is negative. Now, we will set appropriate crisp systems corresponding to the fully fuzzy system (11). To simplify, we use a proper permutation matrix to order the components of x in terms of their signs, so that the nonnegative and negative cores are settled as the beginning and ending components of the vector of mean values, respectively.

Proposition 4.1. *Suppose $\overline{x} = Px$, where P, $n \times n$, is a permutation matrix such that there exists $k \in \{1, ..., n\}$ with $x_i \geq 0$, for $i = 1, ..., k$, and $x_i < 0$, for $i = k + 1, ..., n$. The system* (11) *may be represented by using P as follows:*

$$\widetilde{\overline{A}} \otimes \widetilde{\overline{x}} = \widetilde{b}, \tag{17}$$

where $\widetilde{\overline{A}} = \widetilde{A} \otimes P^T$, $\widetilde{\overline{x}} = P \otimes \widetilde{x}$. Similarly, the system (10) *is reformulated as*

$$\begin{cases} (\widetilde{\overline{a}}_{11} \otimes \widetilde{\overline{x}}_1) \oplus (\widetilde{\overline{a}}_{12} \otimes \widetilde{\overline{x}}_2) \oplus \cdots \oplus (\widetilde{\overline{a}}_{1n} \otimes \widetilde{\overline{x}}_n) = \widetilde{b}_1 \\ (\widetilde{\overline{a}}_{21} \otimes \widetilde{\overline{x}}_1) \oplus (\widetilde{\overline{a}}_{22} \otimes \widetilde{\overline{x}}_2) \oplus \cdots \oplus (\widetilde{\overline{a}}_{2n} \otimes \widetilde{\overline{x}}_n) = \widetilde{b}_2 \\ \vdots \\ (\widetilde{\overline{a}}_{m1} \otimes \widetilde{\overline{x}}_1) \oplus (\widetilde{\overline{a}}_{m2} \otimes \widetilde{\overline{x}}_2) \oplus \cdots \oplus (\widetilde{\overline{a}}_{mn} \otimes \widetilde{\overline{x}}_n) = \widetilde{b}_m, \end{cases} \tag{18}$$

where $\widetilde{\overline{a}}_{ij} = (\overline{a}_{ij}, \overline{m}_{ij}, \overline{n}_{ij})_{LR}$, $1 \leq i \leq m$, $1 \leq j \leq n$ *and* $\widetilde{\overline{x}}_j = (\overline{x}_j, \overline{y}_j, \overline{z}_j)_{LR}$, $1 \leq j \leq n$.

Proof. Simple substitutions and rearrangements. □

Notations 4.2. Set $\overline{A} = AP^T = \left(\ \overline{a}_{ij} \ \right)_{m\times n}$, $\overline{M} = MP^T = \left(\ \overline{m}_{ij} \ \right)_{m\times n}$, $\overline{N} = NP^T = \left(\ \overline{n}_{ij} \ \right)_{m\times n}$, $\overline{x} = Px = \left(\ \overline{x}_j \ \right)_{n\times 1}$, $\overline{y} = Py = \left(\ \overline{y}_j \ \right)_{n\times 1}$, $\overline{z} = Pz = \left(\ \overline{z}_j \ \right)_{n\times 1}$, $\widetilde{\overline{A}} = (\overline{A}, \overline{M}, \overline{N})$ and $\widetilde{\overline{x}} = (\overline{x}, \overline{y}, \overline{z})$.

Now, applying (3)-(4) to (18) results in:

$$\Big(\textstyle\sum_{j=1}^n \overline{a}_{ij}\overline{x}_j \ , \ \sum_{j=1}^n \overline{a}_{ij}\overline{y}_j + \sum_{j=1}^k \overline{m}_{ij}\overline{x}_j + \sum_{j=k+1}^n (-\overline{n}_{ij}\overline{x}_j) \ , \ \sum_{j=1}^n \overline{a}_{ij}\overline{z}_j + \sum_{j=1}^k \overline{n}_{ij}\overline{x}_j + \sum_{j=k+1}^n (-\overline{m}_{ij}\overline{x}_j) \Big) = (b_i, b_{iL}, b_{iR}), \quad i = 1, \ldots, m. \tag{19}$$

By setting

$$\overline{u}_{ij} = \begin{cases} \overline{m}_{ij} & 1 \leq j \leq k \\ -\overline{n}_{ij} & k < j \leq n, \end{cases} \tag{20}$$

and

$$\overline{v}_{ij} = \begin{cases} \overline{n}_{ij} & 1 \leq j \leq k \\ -\overline{m}_{ij} & k < j \leq n, \end{cases} \tag{21}$$

we obtain:

$$\left(\ \textstyle\sum_{j=1}^n \overline{a}_{ij}\overline{x}_j \ , \ \sum_{j=1}^n (\overline{a}_{ij}\overline{y}_j + \overline{u}_{ij}\overline{x}_j) \ , \ \sum_{j=1}^n (\overline{a}_{ij}\overline{z}_j + \overline{v}_{ij}\overline{x}_j) \ \right) = (b_i, b_{iL}, b_{iR}), \tag{22}$$

or in matrix form,

$$\left(\ \overline{A}\overline{x} \ , \ \overline{A}\overline{y} + \overline{U}\overline{x} \ , \ \overline{A}\overline{z} + \overline{V}\overline{x} \ \right) \ = \ (b, b_L, b_R), \tag{23}$$

where $\overline{U} = \left(\ \overline{u}_{ij} \ \right)_{m\times n}$ and $\overline{V} = \left(\ \overline{v}_{ij} \ \right)_{m\times n}$. So, the left and right spreads of the solution (18) are found by solving the following two real systems,

$$\overline{A}\overline{y} = b_L - \overline{U}\overline{x}, \tag{24}$$

and

$$\overline{A}\overline{z} = b_R - \overline{V}\overline{x}. \tag{25}$$

Since the formulas (3)-(6) give approximate values of the product of LR fuzzy numbers, we are to find an approximate solution of the system (17). Now, we have the following result.

Theorem 4.3. *A fuzzy vector* $\widetilde{\overline{x}} = (\widetilde{\overline{x}}_1, \ldots, \widetilde{\overline{x}}_n)^T$ *given by* $\widetilde{\overline{x}}_j = (\overline{x}_j, \overline{y}_j, \overline{z}_j)$, $j = 1, \ldots, n$, *is an approximate solution of* (17) *if and only if* $\overline{x} = Px$, *where* x *satisfies* (15), *and* $\overline{y}$ *and* $\overline{z}$ *are the solutions of* (24) *and* (25), *respectively.*

Remark 4.4. To compute an approximate solution of (17), in addition to the mean-value vector $\overline{x}$, two nonnegative vectors involving the left and right spreads are needed. By solving the system (15), the mean-value vector x is obtained and by Proposition 4.1 we have $\overline{x} = Px$. The spread vectors $\overline{y}$ and $\overline{z}$ are obtained by solving the systems (24) and (25), respectively. The obtained results would form a fuzzy solution for (17) if $\overline{y}$ and $\overline{z}$ satisfy $\overline{x}_j - \overline{y}_j \geq 0$, for $j = 1, ..., k$ and $\overline{x}_j + \overline{z}_j \leq 0$, for $j = k+1, ..., n$. Otherwise, we can find approximate solutions for (17).

The real systems of spreads are also solved by applying the modified Huang algorithm. But, the obtained solutions do not guarantee that $\widetilde{\overline{x}}_1, \ldots, \widetilde{\overline{x}}_k$ be nonnegative fuzzy numbers and $\widetilde{\overline{x}}_{k+1}, \ldots, \widetilde{\overline{x}}_n$ be negative. Indeed, three cases may occur in each of the real systems (24) and (25). First, we consider the system (24):

Case 1: The system has a unique nonnegative solution satisfying

$$\overline{x}_j - \overline{y}_j \geq 0, \qquad j = 1, ..., k. \tag{26}$$

Case 2: The system has infinitely many nonnegative solutions, $\overline{y}$, satisfying $\overline{x}_j - \overline{y}_j \geq 0$, $j = 1, ..., k$. In this case, we are able to find a specific solution by choosing proper parameters of the ABS method. For instance, the solution with minimal Euclidean norm is obtained by choosing the starting point proportional to the first row of $\overline{A}$ in the modified Huang algorithm (e.g., the zero vector), as pointed out in section 3.

Case 3: The systems lacks a solution. Here, we will find an approximate solution for which the residual of the system has a minimal Euclidean norm by solving the following constrained optimization problem,

$$\begin{array}{lll} \underset{\overline{y}}{min} & s_{left} = \|\overline{A}\overline{y} + \overline{U}\overline{x} - b_L\|_2^2 & \\ s.t. & \overline{y} \geq 0 & \\ & \overline{x}_j - \overline{y}_j \geq 0, \qquad j = 1, \ldots, k. & \end{array} \tag{27}$$

Also, the following three cases may happen in considering the system (25):

Case 1: The system has a unique nonnegative solution satisfying

$$\overline{x}_j + \overline{z}_j \leq 0, \qquad j = k+1, ..., n. \tag{28}$$

Case 2: The system has infinitely many nonnegative solutions, $\overline{z}$, satisfying $\overline{x}_j + \overline{z}_j \leq 0$, $j = k+1, ..., n$. Here, we are able to find a specific solution by choosing proper parameters of the ABS method. For instance, the solution with minimal Euclidean norm is obtained by choosing the starting point proportional to the first row of $\overline{A}$ in the modified Huang algorithm (e.g., the zero vector).

Case 3: The systems lacks a solution. In this case, we will find an approximate solution for which the residual of the system has a minimal Euclidean norm by solving the following constrained optimization problem,

$$\begin{array}{ll} \min\limits_{\overline{z}} & s_{right} = \|\overline{A}\overline{z} + \overline{V}\overline{x} - b_R\|_2^2 \\ s.t. & \\ & \overline{z} \geq 0 \\ & \overline{x}_j + \overline{z}_j \leq 0, \; j = k+1, \ldots, n. \end{array} \tag{29}$$

By defining vectors $h = \begin{pmatrix} h_1, & \ldots & , h_n \end{pmatrix}^T$ and $g = \begin{pmatrix} g_1, & \ldots & , g_n \end{pmatrix}^T$ as

$$h_j = \begin{cases} \overline{x}_j & 1 \leq j \leq k \\ +\infty & k < j \leq n, \end{cases} \tag{30}$$

and

$$g_j = \begin{cases} +\infty & 1 \leq j \leq k \\ -\overline{x}_j & k < j \leq n, \end{cases} \tag{31}$$

the optimization problems (27) and (29) may be reformulated as

$$\begin{array}{ll} \min\limits_{\overline{y}} & s_{left} = \|\overline{A}\overline{y} + \overline{U}\overline{x} - b_L\|_2^2 \\ s.t. & \\ & 0 \leq \overline{y} \leq h \end{array} \tag{32}$$

and

$$\begin{array}{ll} \min\limits_{\overline{z}} & s_{right} = \|\overline{A}\overline{z} + \overline{V}\overline{x} - b_R\|_2^2 \\ s.t. & \\ & 0 \leq \overline{z} \leq g, \end{array} \tag{33}$$

respectively. Note that problems (32) and (33) always have solutions, since non-negative objective functions are being minimized on convex feasible spaces.

Remark 4.5. First, we solve the real system (15) by the modified Huang algorithm. If it lacks a solution, then we solve the least squares problem (16) in which case the value of the objective function in the minimizing problem is non-zero. Then, we solve the real left spread system (24) by the modified Huang algorithm. If a solution with the conditions (26) does not exist, then the least squares problem (32) is solved in which case the value of the objective function is non-zero. Similarly, we solve the real right spread system (25) by the modified Huang algorithm. Unless a solution with the conditions (28) exists, the least squares problem (33) is solved in which case the value of objective function is non-zero.

For solving nonlinear constrained optimization problems, there are various approaches in the literature such as Lagrangian coefficient method, trust region method, different interior point methods like gradient projection and conjugate gradient methods [11, 27], etc. However, one can also find the optimum solution of the constrained least squares problem using the Matlab program ***lsqlin***.

Remark 4.6. As proposed by Ghanbari et al. [22], the two optimization problems (32) and (33) may conveniently be combined into:

$$\begin{array}{ll} \underset{\overline{y},\overline{z}}{min} & s = \|\overline{A}\overline{y} + \overline{U}\overline{x} - b_L\|_2^2 + \|\overline{A}\overline{z} + \overline{V}\overline{x} - b_R\|_2^2 \\ s.t. & \\ & 0 \le \overline{y} \le h \\ & 0 \le \overline{z} \le g. \end{array} \tag{34}$$

Consequently, we can consider either two optimization problems of size n or one of size $2n$.

In [22], the authors proposed an approach to guarantee that the obtained solution be fuzzy numbers. Here, we follow the same approach and show the corresponding appropriate changes in the optimization problems (32)-(34).

Remark 4.7. In the least squares problems (32)-(34), we may consider the numbers $\overline{y}_j$, $\overline{z}_j$, $j = 1, \dots, n$, having small positive values, say 10^{-20}. In this case, the LR fuzzy numbers $(\overline{x}_j, \overline{y}_j, \overline{z}_j)$ are very close to the crisp numbers $\overline{x}_j$. Thus, one can define a positive parameter ε such that $\overline{y}_j, \overline{z}_j \ge \varepsilon$, for all j. By this choice, it is guaranteed that each fuzzy number has positive spreads at least as much as ε. Therefore, instead of problems (32) and (33), we consider the problems,

$$\begin{array}{ll} \underset{\overline{y}}{min} & s_{left} = \|\overline{A}\overline{y} + \overline{U}\overline{x} - b_L\|_2^2 \\ s.t. & \\ & \varepsilon.u \le \overline{y} \le h \end{array} \tag{35}$$

and

$$\begin{array}{ll} \underset{\overline{z}}{min} & s_{right} = \|\overline{A}\overline{z} + \overline{V}\overline{x} - b_R\|_2^2 \\ s.t. & \\ & \varepsilon.u \le \overline{z} \le g, \end{array} \tag{36}$$

respectively, and instead of the least squares problem (34), we may consider the following optimization problem,

$$\begin{array}{ll} \underset{\overline{y},\overline{z}}{min} & s = \|\overline{A}\overline{y} + \overline{U}\overline{x} - b_L\|_2^2 + \|\overline{A}\overline{z} + \overline{V}\overline{x} - b_R\|_2^2 \\ s.t. & \\ & \varepsilon.u \le \overline{y} \le h \\ & \varepsilon.u \le \overline{z} \le g, \end{array} \tag{37}$$

where $u = \begin{pmatrix} 1, & \dots & ,1 \end{pmatrix}^T \in \Re^n$ and $\varepsilon > 0$ is a user-defined parameter.

Remark 4.8. Since the solution of the system (17) provides a permuted solution for (11), the solution for (11) is thus turned to

$$(x, y, z) = (P^T\overline{x}, P^T\overline{y}, P^T\overline{z}).$$

We are now ready to give an algorithm for solving an $FFLS$ with nonnegative coefficient matrix.

Algorithm 4.9. Find an approximate solution for $FFLS$.

(A) Give $\varepsilon > 0$ to be used for (35) and (36).

(B) Find the solution of (15), x, by the modified Huang algorithm. **If** the system (15) has no solution **then** solve the optimization problem (16).

(C) Let k be the number of nonnegative components of x.

(D) Let $\overline{x} = Px$, where P is an appropriate permutation matrix such that $\overline{x}_j \geq 0$, for $j \leq k$ and $\overline{x}_j < 0$, for $j > k$.

(E) Determine $\overline{A}$, $\overline{M}$ and $\overline{N}$ as specified in Notation 4.2.

(F) Define $\overline{U}$ and $\overline{V}$ by (20) and (21), and the vectors h and g by (30) and (31), respectively.

(G) Find the solution of (24), $\overline{y}$, by the modified Huang algorithm.
If the system (24) has no solution **or** $\overline{y}$ has at least one negative component **or** there exists an index j, $1 \leq j \leq k$, such that $\overline{x}_j - \overline{y}_j < 0$ **then** solve the optimization problem (35).

(H) Find the solution of (25), $\overline{z}$, by the modified Huang algorithm.
If the system (25) has no solution **or** $\overline{z}$ has at least one negative component **or** there exists an index j, $k < j \leq n$, such that $\overline{x}_j + \overline{z}_j > 0$ **then** solve the optimization problem (36).

(I) The approximate solution of (11) is $\widetilde{x} = (x, y, z)$ where $x = P^T\overline{x}$, $y = P^T\overline{y}$ and $z = P^T\overline{z}$.

(J) **Stop**.

Remark 4.10. Note that Algorithm 4.9 terminates, since the modified Huang algorithm is used a finite number of times (three times, in steps (B), (G) and (H)) and the convergence of Huaung's algorithm as a special case of ABS algorithms is already established (see Remark 3.4).

5. Solution of the $FFLS$ with Single-signed Coefficient Matrix

Here, we consider the fully fuzzy systems for which the components of the coefficient matrix are single-signed LR fuzzy numbers whether nonnegative or negative and search for its single-signed solution. Consider the system (11) and suppose that all the components of $\widetilde{A}$ and $\widetilde{b}$ are single-signed LR fuzzy numbers. We break up $\widetilde{A}$ into two $m \times n$ matrices such that their addition is $\widetilde{A}$, and show that in this special case, the distributive law is confirmed.

Proposition 5.1. *Suppose that components of* $\widetilde{A} = \left(\widetilde{a}_{ij} \right)_{m\times n}$ *and* $\widetilde{x} = \left(\widetilde{x}_j \right)_{n\times 1}$ *are single-signed LR fuzzy numbers. Let* $\widetilde{A}^+ = \left(\widetilde{a}^+_{ij} \right)_{m\times n}$ *and* $\widetilde{A}^- = \left(\widetilde{a}^-_{ij} \right)_{m\times n}$, *where,*

$$\widetilde{a}^+_{ij} = \begin{cases} \widetilde{a}_{ij} & \widetilde{a}_{ij} \geq 0 \\ 0 & \widetilde{a}_{ij} < 0 \end{cases}$$

and

$$\widetilde{a}_{ij}^- = \begin{cases} 0 & \widetilde{a}_{ij} \geq 0 \\ \widetilde{a}_{ij} & \widetilde{a}_{ij} < 0. \end{cases}$$

Then,

$$(\widetilde{A}^+ \oplus \widetilde{A}^-) \otimes \widetilde{x} = (\widetilde{A}^+ \otimes \widetilde{x}) \oplus (\widetilde{A}^- \otimes \widetilde{x}). \tag{38}$$

Proof. Evidently,

$$(\widetilde{A}^+ \oplus \widetilde{A}^-) \otimes \widetilde{x} = \widetilde{A} \otimes \widetilde{x}, \tag{39}$$

and by Definition 2.7,

$$\left(\widetilde{A} \otimes \widetilde{x}\right)_i = \sum_{j, \widetilde{a}_{ij} \geq 0}^{\oplus} (\widetilde{a}_{ij} \otimes \widetilde{x}_j) \ \oplus \sum_{j, \widetilde{a}_{ij} < 0}^{\oplus} (\widetilde{a}_{ij} \otimes \widetilde{x}_j), \ i = 1, ..., m. \tag{40}$$

Consider the definition for $\widetilde{A}^+$ and $\widetilde{A}^-$ as above. We can easily establish:

$$\begin{aligned} \left(\widetilde{A} \otimes \widetilde{x}\right)_i &= \textstyle\sum_{j=1}^n (\widetilde{a}_{ij}^+ \otimes \widetilde{x}_j) \oplus \sum_{j=1}^n (\widetilde{a}_{ij}^- \otimes \widetilde{x}_j) \\ &= (\widetilde{A}^+ \otimes \widetilde{x})_i \oplus (\widetilde{A}^- \otimes \widetilde{x})_i, \qquad i = 1, ..., m. \end{aligned} \tag{41}$$

Therefore, by (39) and (41), the proof is complete. □

Note that this decomposition simplifies the fuzzy multiplications in the system (10), since the produced matrices, $\widetilde{A}^+$ and $\widetilde{A}^-$, are single-signed matrices.

Remark 5.2. According to Proposition 5.1, the system (11) is reformulated as:

$$(\widetilde{A}^+ \otimes \widetilde{x}) \oplus (\widetilde{A}^- \otimes \widetilde{x}) = \widetilde{b}. \tag{42}$$

Let

$$\begin{aligned} \widetilde{A}^+ &= (A^+, M^+, N^+), \\ \widetilde{A}^- &= (A^-, M^-, N^-), \end{aligned} \tag{43}$$

or equivalently,

$$\begin{aligned} \widetilde{A}^+ &= \left(\ \widetilde{a}_{ij}^+ \ \right)_{m \times n}, \ \widetilde{a}_{ij}^+ = (a_{ij}^+, m_{ij}^+, n_{ij}^+), \quad i = 1, \ldots, m, \ j = 1, \ldots, n, \\ \widetilde{A}^- &= \left(\ \widetilde{a}_{ij}^- \ \right)_{m \times n}, \ \widetilde{a}_{ij}^- = (a_{ij}^-, m_{ij}^-, n_{ij}^-), \quad i = 1, \ldots, m, \ j = 1, \ldots, n. \end{aligned} \tag{44}$$

Proposition 5.3. *Definition* (43) *results in:*

$$\begin{aligned} M^+ + M^- &= M, \\ N^+ + N^- &= N. \end{aligned} \tag{45}$$

Proof. Note that $\widetilde{A}^+ + \widetilde{A}^- = \widetilde{A}$. □

Applying (3)-(6) to multiply $\widetilde{A}^+$ and $\widetilde{A}^-$ by $\widetilde{x}$ in (42), we obtain:

$$\begin{aligned} &\textstyle\sum_{j, \widetilde{x}_j \geq 0}^{\oplus} (\widetilde{a}_{ij}^+ \otimes \widetilde{x}_j) \ \oplus \sum_{j, \widetilde{x}_j < 0}^{\oplus} (\widetilde{a}_{ij}^+ \otimes \widetilde{x}_j) \\ \oplus &\textstyle\sum_{j, \widetilde{x}_j \geq 0}^{\oplus} (\widetilde{a}_{ij}^- \otimes \widetilde{x}_j) \ \oplus \sum_{j, \widetilde{x}_j < 0}^{\oplus} (\widetilde{a}_{ij}^- \otimes \widetilde{x}_j) \ = \widetilde{b}_i, \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad i = 1, \ \ldots, m, \end{aligned} \tag{46}$$

and thus,

$$\begin{aligned}
&\textstyle\sum^{\oplus}_{j,\widetilde{x}_j\geq 0}(a^+_{ij}x_j, a^+_{ij}y_j + m^+_{ij}x_j, a^+_{ij}z_j + n^+_{ij}x_j)\\
&\oplus \textstyle\sum^{\oplus}_{j,\widetilde{x}_j<0}(a^+_{ij}x_j, a^+_{ij}y_j - n^+_{ij}x_j, a^+_{ij}z_j - m^+_{ij}x_j)\\
&\oplus \textstyle\sum^{\oplus}_{j,\widetilde{x}_j\geq 0}(a^-_{ij}x_j, -a^-_{ij}z_j + m^-_{ij}x_j, -a^-_{ij}y_j + n^-_{ij}x_j)\\
&\oplus \textstyle\sum^{\oplus}_{j,\widetilde{x}_j<0}(a^-_{ij}x_j, -a^-_{ij}z_j - n^-_{ij}x_j, -a^-_{ij}y_j - m^-_{ij}x_j)\\
&= (b_i, b_{iL}, b_{iR}), \qquad\qquad i = 1,\ \dots\ , m. \qquad (47)
\end{aligned}$$

By setting

$$u^+_{ij} = \begin{cases} m^+_{ij} & \widetilde{x}_j \geq 0 \\ -n^+_{ij} & \widetilde{x}_j < 0 \end{cases}, \quad u^-_{ij} = \begin{cases} m^-_{ij} & \widetilde{x}_j \geq 0 \\ -n^-_{ij} & \widetilde{x}_j < 0 \end{cases} \qquad (48)$$

and

$$v^+_{ij} = \begin{cases} n^+_{ij} & \widetilde{x}_j \geq 0 \\ -m^+_{ij} & \widetilde{x}_j < 0 \end{cases}, \quad v^-_{ij} = \begin{cases} n^-_{ij} & \widetilde{x}_j \geq 0 \\ -m^-_{ij} & \widetilde{x}_j < 0, \end{cases} \qquad (49)$$

we get

$$\begin{aligned}
&\Big(\textstyle\sum^n_{j=1} a^+_{ij}x_j, \sum^n_{j=1}(a^+_{ij}y_j + u^+_{ij}x_j), \sum^n_{j=1}(a^+_{ij}z_j + v^+_{ij}x_j)\Big)\\
&\oplus \Big(\textstyle\sum^n_{j=1} a^-_{ij}x_j, \sum^n_{j=1}(-a^-_{ij}z_j + u^-_{ij}x_j), \sum^n_{j=1}(-a^-_{ij}y_j + v^-_{ij}x_j)\Big)\\
&= (b_i, b_{iL}, b_{iR}), \qquad\qquad i = 1,\ \dots\ , m. \qquad (50)
\end{aligned}$$

Suppose that $u_{ij} = u^+_{ij} + u^-_{ij}$ and $v_{ij} = v^+_{ij} + v^-_{ij}$. According to Proposition 5.3, if $\widetilde{x}_j \geq 0$, then we have,

$$u_{ij} = m^+_{ij} + m^-_{ij} = m_{ij}, \qquad v_{ij} = n^+_{ij} + n^-_{ij} = n_{ij},$$

and if $\widetilde{x}_j < 0$, then we have,

$$u_{ij} = -n^+_{ij} - n^-_{ij} = -n_{ij} \quad \mathit{and} \quad v_{ij} = -m^+_{ij} - m^-_{ij} = -m_{ij}\ .$$

So,

$$u_{ij} = \begin{cases} m_{ij} & \widetilde{x}_j \geq 0 \\ -n_{ij} & \widetilde{x}_j < 0, \end{cases}$$

$$v_{ij} = \begin{cases} n_{ij} & \widetilde{x}_j \geq 0 \\ -m_{ij} & \widetilde{x}_j < 0. \end{cases} \qquad (51)$$

The relation (47) is represented in the matrix form as:

$$\begin{pmatrix} Ax, & A^+y - A^-z + Ux, & -A^-y + A^+z + Vx \end{pmatrix} = \begin{pmatrix} b, & b_L, & b_R \end{pmatrix}, \qquad (52)$$

where,

$$\begin{aligned} U &= \begin{pmatrix} u_{ij} \end{pmatrix}_{m\times n},\\ V &= \begin{pmatrix} v_{ij} \end{pmatrix}_{m\times n}. \end{aligned} \qquad (53)$$

We get the mean values of the solution by equating the primary component of the ordered 3-tuple. Thus, we may state the following result.

Theorem 5.4. *The vector of the mean values of the solution* (42) *is found by solving the system,*

$$Ax = b. \tag{54}$$

Three cases may occur after employing the modified Huang algorithm to solve the system (54):

Case 1: The system has a unique solution.

Case 2: The system has more than one solution. By choosing appropriate parameters of the ABS method, we are able to find a specific solution. For example, the solution with a minimal Euclidean norm is obtained by choosing $x_1 = 0$ in the modified Huang algorithm.

Case 3: The system lacks a solution. In this case, we will find an approximate solution x, for which the residual of the system has a minimal Euclidean norm. In other words, we solve,

$$\|Ax - b\|_2 \;=\; \min_{t \in \Re^n} \|At - b\|_2. \tag{55}$$

Remark 5.5. Define the sets I^+ and I^-, to include the indices of nonnegative and negative components of x as follows:

$$\begin{aligned} I^+ &= \{j \in Z_n \mid x_j \geq 0\}, \\ I^- &= \{j \in Z_n \mid x_j < 0\}, \end{aligned} \tag{56}$$

where $Z_n = \{1, 2, ..., n\}$. We will use these sets in setting up the proper optimization problems.

Now, the left and right spreads of the solution of (11) are obtained by solving the following two real systems of equations,

$$\begin{aligned} A^+ y - A^- z &= b_L - Ux, \\ -A^- y + A^+ z &= b_R - Vx. \end{aligned} \tag{57}$$

Theorem 5.6. *A fuzzy vector* $\widetilde{x} = (\widetilde{x}_1, \ldots, \widetilde{x}_n)^T$ *given by* $\widetilde{x}_j = (x_j, y_j, z_j)$, $j = 1, \ldots, n$, *is an approximate solution of* (11) *if and only if* x *satisfies* (54) *and* $w = \begin{pmatrix} y_1, & \ldots & , y_n & , z_1, & \ldots & , z_n \end{pmatrix}^T$ *is the solution of*

$$\begin{cases} Cw = b_L - Ux \\ Dw = b_R - Vx, \end{cases} \tag{58}$$

where,

$$\begin{aligned} C &= \begin{pmatrix} A^+ & -A^- \end{pmatrix}_{m \times 2n}, \\ D &= \begin{pmatrix} -A^- & A^+ \end{pmatrix}_{m \times 2n}. \end{aligned} \tag{59}$$

Remark 5.7. As pointed out in section 4, some properties for the solutions of the systems are sought to gain strong fuzzy solutions.

Remark 5.8. Suppose that when $x_j \geq 0$ ($x_j < 0$) the corresponding fuzzy number is nonnegative (negative). So, according to Remark 5.5, the fuzziness of solutions may be guaranteed by solving the following optimization problem,

$$\begin{array}{lll} \min\limits_{w\in\Re^{2n}} & s = \|Cw + Ux - b_L\|_2^2 + \|Dw + Vx - b_R\|_2^2 & \\ s.t. & & \\ & w \geq 0 & \\ & w_j \leq x_j, & j \in I^+ \\ & w_{n+j} \leq -x_j, & j \in I^-. \end{array} \tag{60}$$

Ghanbari et al. [22] proposed an approach to guarantee that the obtained solution be fuzzy numbers. Here, we also follow their approach and describe the corresponding changes in the optimization problem (60).

Remark 5.9. In the least squares problem (60), we may consider the numbers w_j, $j = 1, \ldots, 2n$, to have small positive values. In this case, the LR fuzzy numbers (x_j, y_j, z_j) are very close to the crisp numbers x_j. Thus, one can define a positive parameter ε such that $w_j \geq \varepsilon$ for all j. By this choice, it is guaranteed that each fuzzy number has positive spreads at least as much as ε. Therefore, instead of the least squares problem (60), we solve the following optimization problem,

$$\begin{array}{lll} \min\limits_{w\in\Re^{2n}} & s = \|Cw + Ux - b_L\|_2^2 + \|Dw + Vx - b_R\|_2^2 & \\ s.t. & & \\ & w \geq \varepsilon.u & \\ & w_j \leq x_j, & j \in I^+ \\ & w_{n+j} \leq -x_j, & j \in I^-, \end{array} \tag{61}$$

where $u = \begin{pmatrix} 1, & \ldots & ,1 \end{pmatrix}^T \in \Re^{2n}$ and $\varepsilon > 0$ is a user-defined parameter.

Algorithm 5.10. Find an approximate solution for an $FFLS$.

(A) Give $\varepsilon > 0$ for (61).

(B) Find the solution of (54), x, by the modified Huang algorithm. **If** the system (54) has no solution **then** solve the optimization problem (55).

(C) Define the sets I^+ and I^-, as specified in Remark 5.5 and the two matrices U and V, as given by (51).

(D) Find the solution of (58), $w = \begin{pmatrix} y_1, & \ldots & ,y_n & ,z_1, & \ldots & ,z_n \end{pmatrix}^T$, by the modified Huang algorithm. **If** the system (58) has no solution **or** w is not feasible for the constrains of the problem (61) **then** solve the optimization problem (61).

(E) The approximate solution of (11) is $\widetilde{x} = (x, y, z)$.

(F) **Stop**.

Remark 5.11. Similar to Remark 4.10, the convergence of Algorithm 5.10 is obviously confirmed.

6. **Numerical Testing**

Here, we apply our proposed algorithms to several randomly generated test problems and show the effectiveness of our approach in computing an approximate single-signed solution of the fully fuzzy linear systems. In each example, we first produce the coefficient matrix $\widetilde{A} = \left(\widetilde{a}_{ij} \right)_{m\times n}$, and the exact solution $\widetilde{x} = \left(\widetilde{x}_j \right)_{n\times 1}$, randomly by the function ***rand*** in Matlab. Let $\widetilde{a}_{ij} = (a_{ij}, m_{ij}, n_{ij})$ and $\widetilde{x}_j = (x_j, y_j, z_j)$, for $i = 1, ..., m$, $j = 1, ..., n$. In the examples for Algorithm 4.9, the core values a_{ij} and x_j are arbitrary values in $[0, k]$, $k > 0$ for all i and j, and in the examples for Algorithm 5.10, they are arbitrary values in $[-k, k]$, $k > 0$, for all i and j. The spreads are proportional to the absolute values of the cores with random positive multiples not greater than one, that is,

$$m_{ij} = \alpha|a_{ij}|,\ n_{ij} = \alpha|a_{ij}|,\ y_j = \alpha|x_j|,\ z_j = \alpha|x_j|,$$

where $\alpha = \rho.\mathbf{rand}$, $\rho \in (0, 1]$. We used $\rho = 0.25$ and $k = 10$. Then, the right hand side vector is specified by multiplying the coefficient matrix and the exact solution. The first two test examples are used for Algorithm 4.9 and the third one is used for Algorithm 5.10. In all the examples, we set $\varepsilon = 10^{-5}$.

Example 6.1. Here, the spread systems do not have nonnegative solutions, and so the least squares problems (35) and (36) are solved to find an approximate fuzzy solution of the system,

$$\begin{cases} (7.9995, 0.1169, 0.1077) \otimes \widetilde{x}_1 \oplus (2.5928, 0.1233, 0.1199) \otimes \widetilde{x}_2 \oplus \\ (7.3328, 0.5358, 0.0183) \otimes \widetilde{x}_3 \oplus (1.5238, 0.1931, 0.1768) \otimes \widetilde{x}_4 \oplus \\ \qquad\qquad = (-55.9021, 3.9565, 7.9603) \\ \\ (2.8943, 0.4454, 0.0093) \otimes \widetilde{x}_1 \oplus (7.1323, 1.6974, 0.0338) \otimes \widetilde{x}_2 \oplus \\ (6.2233, 1.2862, 0.1223) \otimes \widetilde{x}_3 \oplus (2.0332, 0.4307, 0.0228) \otimes \widetilde{x}_4 \oplus \\ \qquad\qquad = (-12.6929, 8.4970, 8.9149) \\ \\ (6.9510, 1.1282, 0.0230) \otimes \widetilde{x}_1 \oplus (7.2037, 0.8526, 0.1366) \otimes \widetilde{x}_2 \oplus \\ (9.8978, 0.3298, 0.0928) \otimes \widetilde{x}_3 \oplus (8.1932, 0.1459, 0.1162) \otimes \widetilde{x}_4 \oplus \\ \qquad\qquad = (-43.2516, 7.4950, 14.4877). \end{cases}$$

The exact solution of this system is:

$$\widetilde{x} = \begin{bmatrix} (-5.6598, 0.1515, 0.5444) \\ (2.5526, 0.3670, 0.0816) \\ (-2.3849, 0.1024, 0.1167) \\ (0.1595, 0.0285, 0.1735) \end{bmatrix}.$$

Solving the corresponding system (15), by the modified Huang algorithm, results in $x = \begin{pmatrix} -5.5609, & 2.6283, & -2.5230, & 0.1759 \end{pmatrix}^T$. To obtain the spreads, we first solve the systems (24) and (25) by the modified Huang algorithm. The results are

$$\overline{y} = \begin{pmatrix} 0.3087, & -0.0161, & 0.1979, & 0.0427 \end{pmatrix}^T$$

and

$$\overline{z} = \begin{pmatrix} -0.2054, & 1.1892, & 0.2283, & 0.2794 \end{pmatrix}^T,$$

respectively. There exists a negative component in each spread vector. So, we solve the optimization problem (35) and (36). Therefore, the obtained approximate solution is:

$$\widetilde{x} = \begin{bmatrix} (-5.5609, 0.1845, 0.4203) \\ (2.6283, 0.4173, 0) \\ (-2.5230, 0, 0) \\ (0.1759, 0.0072, 1.1546) \end{bmatrix},$$

for which the Hausdorff distance from the exact solution is 0.2547.

Example 6.2. In this example, the systems (24) and (25) have solutions, and thus there is no need to solve the optimization problems (35) and (36). Consider,

$$\begin{cases}
(3.2589, 1.3812, 0.0866) \otimes \widetilde{x}_1 \oplus (1.5448, 0.2413, 0.1269) \otimes \widetilde{x}_2 \oplus \\
(7.9193, 0.3802, 0.0863) \otimes \widetilde{x}_3 \oplus (5.1007, 0.1490, 0.0864) \otimes \widetilde{x}_4 \oplus \\
\qquad\qquad = (120.2749, 26.4283, 3.4467) \\
\\
(6.8876, 0.5626, 0.1025) \otimes \widetilde{x}_1 \oplus (4.0028, 0.3829, 0.0368) \otimes \widetilde{x}_2 \oplus \\
(4.1420, 1.1067, 0.0974) \otimes \widetilde{x}_3 \oplus (1.9735, 0.5069, 0.0986) \otimes \widetilde{x}_4 \oplus \\
\qquad\qquad = (94.2114, 26.3155, 4.0641) \\
\\
(9.2328, 2.2363, 0.1654) \otimes \widetilde{x}_1 \oplus (0.6457, 0.0207, 0.1685) \otimes \widetilde{x}_2 \oplus \\
(8.1640, 1.6113, 0.0690) \otimes \widetilde{x}_3 \oplus (2.0989, 0.3875, 0.0119) \otimes \widetilde{x}_4 \oplus \\
\qquad\qquad = (149.6216, 48.3791, 3.8137) \\
\\
(7.5713, 1.0574, 0.1565) \otimes \widetilde{x}_1 \oplus (1.2035, 0.0424, 0.1115) \otimes \widetilde{x}_2 \oplus \\
(3.9532, 0.2030, 0.1721) \otimes \widetilde{x}_3 \oplus (5.7998, 0.3732, 0.0555) \otimes \widetilde{x}_4 \oplus \\
\qquad\qquad = (112.5987, 20.9014, 5.0149) \\
\\
(4.0111, 0.2890, 0.1470) \otimes \widetilde{x}_1 \oplus (0.2934, 0.0365, 0.0109) \otimes \widetilde{x}_2 \oplus \\
(8.2595, 0.3950, 0.1449) \otimes \widetilde{x}_3 \oplus (9.7921, 2.3836, 0.1409) \otimes \widetilde{x}_4 \oplus \\
\qquad\qquad = (147.8950, 30.2200, 5.0744) \\
\\
(19611, 0.3930, 0.0412) \otimes \widetilde{x}_1 \oplus (8.6519, 1.6713, 0.1745) \otimes \widetilde{x}_2 \oplus \\
(3.3762, 0.6815, 0.1513) \otimes \widetilde{x}_3 \oplus (4.6712, 1.0564, 0.0842) \otimes \widetilde{x}_4 \oplus \\
\qquad\qquad = (65.4385, 19.9329, 4.5168).
\end{cases}$$

The exact solution of this system is:

$$\widetilde{x} = \begin{bmatrix} (6.5840, 0.5721, 0.1336) \\ (0.0130, 0.0014, 0.1680) \\ (9.8200, 1.3000, 0.0514) \\ (4.1231, 0.1611, 0.1116) \end{bmatrix}.$$

Solving the system (15) by the modified Huang algorithm results in

$$x = \begin{pmatrix} 6.5840, & 0.0130, & 9.8200, & 4.1231 \end{pmatrix}^T.$$

Solving the systems (24) and (25) by the modified Huang algorithm results in

$$\overline{y} = \begin{pmatrix} 0.5721, & 0.0014, & 1.3000, & 0.1611 \end{pmatrix}^T$$

and

$$\overline{z} = \begin{pmatrix} 0.1336, & 0.1680, & 0.0514, & 0.1116 \end{pmatrix}^T,$$

respectively. Here, solving the optimization problems is not required, since all the components of $\overline{y}$ and $\overline{z}$ are positive. Obviously, the almost exact solution of the fully fuzzy linear system is:

$$\widetilde{x} = \begin{bmatrix} (6.5840, 0.5721, 0.1336) \\ (0.0130, 0.0014, 0.1680) \\ (9.8200, 1.3000, 0.0514) \\ (4.1231, 0.1611, 0.1116) \end{bmatrix},$$

for which the Hausdorff distance from the exact solution is $4.9544e - 15$.

Example 6.3. This example is used to test Algorithm 5.10. It is found that the corresponding spread system (58) is inconsistent, and so the constrained minimization problem (61) is solved. Consider,

$$\begin{cases} (-1.4840, 0.1168, 0.1244) \otimes \widetilde{x}_1 \oplus (-5.2718, 0.1003, 1.2035) \otimes \widetilde{x}_2 \oplus \\ (8.0951, 0.8449, 0.1599) \otimes \widetilde{x}_3 = (-13.2664, 10.2687, 13.2664) \\ \\ (-6.2145, 0.1027, 1.3837) \otimes \widetilde{x}_1 \oplus (3.1244, 0.3044, 0.1861) \otimes \widetilde{x}_2 \oplus \\ (-7.2331, 0.1540, 0.5113) \otimes \widetilde{x}_3 = (-49.5008, 6.3265, 19.5593) \\ \\ (7.5437, 1.3722, 0.1829) \otimes \widetilde{x}_1 \oplus (1.8174, 0.0482, 0.1778) \otimes \widetilde{x}_2 \oplus \\ (0.4531, 0.0674, 0.0024) \otimes \widetilde{x}_3 = (64.4898, 14.1803, 2.9570) \\ \\ (-2.8060, 0.0247, 0.1672) \otimes \widetilde{x}_1 \oplus (7.7925, 1.7159, 0.1626) \otimes \widetilde{x}_2 \oplus \\ (6.1330, 1.1704, 0.1621) \otimes \widetilde{x}_3 = (52.1180, 25.1771, 4.7417). \end{cases}$$

The exact solution of this system is:

$$\widetilde{x} = \begin{bmatrix} (6.8225, 0.3459, 0.0550) \\ (6.2480, 0.7522, 0.0800) \\ (3.6808, 0.6463, 0.0643) \end{bmatrix}.$$

Solving the system (54) by the modified Huang algorithm results in

$$x = \begin{pmatrix} 6.8225, & 6.2480, & 3.6808 \end{pmatrix}^T.$$

Applying the modified Huang algorithm to solve the system (58), shows that the system is inconsistent. So, we solve the optimization problem (61) to obtain the approximate solution. The result is:

$$\widetilde{x} = \begin{bmatrix} (6.8225, 0.3469, 0.0785) \\ (6.2480, 0.7426, 0.1050) \\ (3.6808, 0.6341, 0.0173) \end{bmatrix},$$

for which the Hausdorff distance from the exact solution is 0.0470.

Range of dimensions	$[1, 10]$	$[10, 100]$	$[100, 500]$
Number of problems	1000	300	50
Number of times the core and spread systems were solved by the modified Huang algorithm	1000	300	50
Average Hausdorff distance	$2.1703e-12$	$2.0147e-10$	$1.2771e-11$
Number of times the core systems were solved by the modified Huang algorithm and spread vectors were obtained by solving the least squares problems	0	0	0
Average Hausdorff distance	$-$	$-$	$-$
Number of times where the core and spread vectors were obtained by solving the least squares problems	0	0	0
Average Hausdorff distance	$-$	$-$	$-$
Total average Hausdorff distance	$2.1703e-12$	$2.0147e-10$	$1.2771e-11$

Table 1. The Results for Algorithm 4.9 on Some Small and Medium Size Fully Fuzzy Linear Systems with Random Exact Solutions and $n = m$

Range of dimensions	$[1, 10]$	$[10, 100]$	$[100, 500]$
Number of problems	1000	300	50
Number of times the core and spread systems were solved by the modified Huang algorithm	999	298	50
Average Hausdorff distance	$2.9416e-13$	$7.8389e-12$	$5.2574e-8$
Number of times the core systems were solved by the modified Huang algorithm and spread vectors were obtained by solving the least squares problems	1	0	0
Average Hausdorff distance	$5.8400e-10$	$-$	$-$
Number of times where the core and spread vectors were obtained by solving the least squares problems	0	2	0
Average Hausdorff distance	$-$	$1.5543e-8$	$-$
Total average Hausdorff distance	$8.7786e-13$	$1.1140e-10$	$5.2574e-8$

Table 2. The Results for Algorithm 4.9 on Some Small and Medium Size Fully Fuzzy Linear Systems with Random Exact Solutions and $n \leq m$

Finally, we show some results for Algorithms 4.9 and 5.10 applied to medium size systems. We generate the systems in two different ways. First, we produce the coefficient matrices and the exact solutions of fully fuzzy linear systems, randomly as mentioned before, to have consistent systems. Tables $1-4$ show the average Hausdorff distances between the exact and approximate solutions of some small and medium size consistent systems. Second, we generate random fully fuzzy linear systems by producing the coefficient matrices and the right hand side vectors, randomly (Tables $5-8$). In this case, it is expected that randomly generated systems be inconsistent. For both problem types, the size of the systems are also set randomly; that is, m and n, the number of rows and columns, respectively, are two random numbers. The first row in the tables shows the range of dimensions of the systems and the second row represents the number of systems randomly generated and solved in each case. The next six rows display the number of problems having exact solutions followed by the number of problems having approximate solutions (their solutions were obtained by solving least squares problems) with the corresponding average Hausdorff distances, separately. The total average Hausdorff distance is displayed in the last row.

Range of dimensions	$[1, 10]$	$[10, 100]$	$[100, 500]$
Number of problems	1000	300	50
Number of times the core and spread systems were solved by the modified Huang algorithm	1000	299	50
Average Hausdorff distance	$5.8681e-13$	$6.6483e-12$	$2.0482e-7$
Number of times the core systems were solved by the modified Huang algorithm and spread vectors were obtained by solving the least squares problems	0	0	0
Average Hausdorff distance	$-$	$-$	$-$
Number of times where the core and spread vectors were obtained by solving the least squares problems	0	1	0
Average Hausdorff distance	$-$	$4.0547e-5$	$-$
Total average Hausdorff distance	$5.8681e-13$	$1.3516e-9$	$2.0482e-7$

TABLE 3. The Results for Algorithm 5.10 on Some Small and Medium Size Fully Fuzzy Linear Systems with Random Exact Solutions and $n = m$

Range of dimensions	$[1, 10]$	$[10, 100]$	$[100, 500]$
Number of problems	1000	300	50
Number of times the core and spread systems were solved by the modified Huang algorithm	998	299	50
Average Hausdorff distance	$5.1441e-13$	$1.4660e-12$	$1.7220e-11$
Number of times the core systems were solved by the modified Huang algorithm and spread vectors were obtained by solving the least squares problems	1	0	0
Average Hausdorff distance	$1.0549e-9$	$-$	$-$
Number of times where the core and spread vectors were obtained by solving the least squares problems	1	1	0
Average Hausdorff distance	$8.3251e-10$	$2.0350e-11$	$-$
Total average Hausdorff distance	$2.4007e-12$	$1.5289e-12$	$1.7220e-11$

TABLE 4. The Results for Algorithm 5.10 on Some Small and Medium Size Fully Fuzzy Linear Systems with Random Exact Solutions and $n \leq m$

Range of dimensions	$[1, 10]$	$[10, 100]$	$[100, 500]$
Number of problems	1000	300	50
Number of times the core and spread systems were solved by the modified Huang algorithm	23	0	0
Average Hausdorff distance	$5.6959e-16$	$-$	$-$
Number of times the core systems were solved by the modified Huang algorithm and spread vectors were obtained by solving the least squares problems	976	0	0
Average Hausdorff distance	26.1178	$-$	$-$
Number of times where the core and spread vectors were obtained by solving the least squares problems	1	300	50
Average Hausdorff distance	$5.2779e+3$	98.1446	243.5114
Total average Hausdorff distance	30.7689	98.1446	243.5114

TABLE 5. The Results for Algorithm 4.9 on Some Small and Medium Size Fully Fuzzy Linear Systems with Random Right Hand Side Vectors and $n = m$

Range of dimensions	$[1, 10]$	$[10, 100]$	$[100, 500]$
Number of problems	1000	300	50
Number of times the core and spread systems were solved by the modified Huang algorithm	29	0	0
Average Hausdorff distance	$3.5795e-16$	–	–
Number of times the core systems were solved by the modified Huang algorithm and spread vectors were obtained by solving the least squares problems	264	17	0
Average Hausdorff distance	12.4078	42.5604	–
Number of times where the core and spread vectors were obtained by solving the least squares problems	707	283	50
Average Hausdorff distance	7.2524	14.7983	31.1396
Total average Hausdorff distance	8.4031	16.3715	31.1396

Table 6. The Results for Algorithm 4.9 on Some Small and Medium Size Fully Fuzzy Linear Systems with Random Right Hand Side Vectors and $n \leq m$

Range of dimensions	$[1, 10]$	$[10, 100]$	$[100, 500]$
Number of problems	1000	300	50
Number of times the core and spread systems were solved by the modified Huang algorithm	23	0	0
Average Hausdorff distance	$5.8890e-16$	–	–
Number of times the core systems were solved by the modified Huang algorithm and spread vectors were obtained by solving the least squares problems	976	0	0
Average Hausdorff distance	10.3472	–	–
Number of times where the core and spread vectors were obtained by solving the least squares problems	1	300	50
Average Hausdorff distance	$2.2464e+3$	26.1545	104.8408
Total average Hausdorff distance	12.3453	26.1545	104.8408

Table 7. The Results for Algorithm 5.10 on Some Small and Medium Size Fully Fuzzy Linear Systems with Random Right Hand Side Vectors and $n = m$

Range of dimensions	$[1, 10]$	$[10, 100]$	$[100, 500]$
Number of problems	1000	300	50
Number of times the core and spread systems were solved by the modified Huang algorithm	22	0	0
Average Hausdorff distance	$3.2297e-16$	–	–
Number of times the core systems were solved by the modified Huang algorithm and spread vectors were obtained by solving the least squares problems	252	13	1
Average Hausdorff distance	6.2394	53.4677	128.3529
Number of times where the core and spread vectors were obtained by solving the least squares problems	726	287	49
Average Hausdorff distance	6.6142	10.8949	20.3055
Total average Hausdorff distance	6.3742	12.7397	22.4664

Table 8. The Results for Algorithm 5.10 on Some Small and Medium Size Fully Fuzzy Linear Systems with Random Right Hand Side Vectors and $n \leq m$

7. Conclusions

We proposed an approach to compute an approximate solution of fully fuzzy linear system ($FFLS$) of equations, where all the components of the coefficient matrix were either nonnegative or nonpositive. First, we considered an $FFLS$ with a nonnegative coefficient matrix. For this case, we obtained an equivalent $FFLS$ by using an appropriate permutation to simplify fuzzy multiplications. To solve the $m \times n$ permutated system, we converted it to three $m \times n$ crisp real linear systems, one being concerned with the cores and the other two being related to the left and right spreads. As an effective method of solution, we proposed the modified Huang algorithm, a special approach within the ABS class of algorithms, to decide whether the systems were consistent or not. If the core system was inconsistent, an appropriate unconstrained least squares problem was solved for an approximate solution. We identified the signs of the components of the fuzzy solution with respect to the signs of the obtained cores. To know whether the left and right spread systems were consistent or not, we also applied the modified Huang algorithm. Appropriate constrained least squares problems were solved, when the spread systems were inconsistent or did not satisfy fuzziness conditions. Then, we considered the $FFLS$ with a mixed single-signed coefficient matrix, where each component of the coefficient matrix was either nonnegative or nonpositive. In this case, we broke the $m \times n$ coefficient matrix up into two $m \times n$ matrices, one having only nonnegative and the other having only nonpositive components, such that their sum yielded the original coefficient matrix. Using the distributive law, we converted each $m \times n$ $FFLS$ to two crisp real linear systems where the first one was related to the cores with size $m \times n$ and the other was $2m \times 2n$ and was related to the spreads. We again considered solving these systems by the modified Huang algorithm, and when the first system was inconsistent or the second system did not satisfy the fuzziness conditions, we found an approximate solution by solving a respective least squares problem. We summarized the proposed approach by presenting two computational algorithms. Finally, the algorithms were implemented and effectively tested by solving various randomly generated consistent as well as inconsistent numerical test problems.

References

[1] J. Abaffy, C. G. Broyden and E. Spedicato, *A class of direct methods for linear systems*, Numerische Mathematik, **45** (1984), 361-376.

[2] J. Abaffy and E. Spedicato, *ABS projection alegorithms: mathematical techniques for linear and nonlinear equations*, John Wiley and Sons, 1989.

[3] S. Abbasbandy, A. Jafarian and R. Ezzati, *Conjugate gradient method for fuzzy symmetric positive definite system of linear equations*, Applied Mathematics and Computation, **171** (2005), 1184-1191.

[4] S. Abbasbandy and M. Alavi, *A method for solving fuzzy linear systems*, Iranian Journal of Fuzzy Systems, **2(2)** (2005), 37-43.

[5] S. Abbasbandy, B. Asady and M. Alavi, *Fuzzy general linear systems*, Applied Mathematics and Computation, **169** (2005), 34-40.

[6] T. Allahviranloo, *Numerical methods for fuzzy system of linear equations*, Applied Mathematics and Computation, **155** (2004), 493-502.

[7] T. Allahviranloo, *Revised solution of an overdetermined fuzzy linear system of equations*, International Journal of Computational Cognition, **6(3)** (2008), 66-70.

[8] T. Allahviranloo, *Successive over relaxation iterative method for fuzzy system of linear equations*, Applied Mathematics and Computation, **162** (2005), 189-196.

[9] T. Allahviranloo, *The Adomian decomposition method for fuzzy system of linear equations*, Applied Mathematics and Computation, **163** (2005), 553-563.

[10] T. Allahviranloo, E. Ahmady, N. Ahmady and K. Shams Alketaby, *Block Jacobi two-stage method with Gauss-Sidel inner iterations for fuzzy system of linear equations*, Applied Mathematics and Computation, **175** (2006), 1217-1228.

[11] M. S. Bazaraa, H. D. Sherali and C. M. Shetty, *Nonlinear programming: theory and algorithms*, John Wiley and Sons, 2006.

[12] J. J. Buckley and Y. Qu, *Solving system of linear fuzzy equations*, Fuzzy Sets and Systems, **43** (1991), 33-43.

[13] M. Dehghan and B. Hashemi, *Iterative the solution of fuzzy linear systems*, Applied Mathematics and Computation, **175** (2006), 645-674.

[14] M. Dehghan and B. Hashemi, *Solution of the fully fuzzy linear systems using the decomposition procedure*, Applied Mathematics and Computation, **182** (2006), 1568-1580.

[15] M. Dehghan, B. Hashemi and M. Ghatee, *Computational methods for solving fully fuzzy linear systems*, Applied Mathematics and Computation, **179** (2006), 328-343.

[16] M. Dehghan, B. Hashemi and M. Ghatee, *Solution of the fully fuzzy linear systems using iterative techniques*, Chaos Solutions and Fractals, **34** (2007), 316-336.

[17] D. Dubois and H. Prade, *Fuzzy sets and systems: theory and applications*, Academic Press, New York, 1980.

[18] H. Esmaeili, N. Mahdavi-Amiri and E. Spedicato, *A class of ABS algorithms for Diophantine linear systems*, Numerische Mathematik, **90** (2001), 101-115.

[19] R. Ezzati, *Approximate symmetric least square solutions of general fuzzy linear systems*, Applied and Computational Mathematics, **9(2)** (2010), 220-225.

[20] R. Ezzati, *Solving fuzzy linear systems*, Soft Computing, **15** (2011), 193-197.

[21] M. Friedman, M. Ming and A. Kandel, *Fuzzy linear systems*, Fuzzy Sets and Systems, **96** (1998), 201-209.

[22] R. Ghanbari, N. Mahdavi-Amiri and R. Yousefpour, *Exact and approximate solutions of fuzzy LR linear systems: new algorithms using a least squares model and ABS approach*, Iranian Journal of Fuzzy Systems, **7(2)** (2010), 1-18.

[23] M. S. Hashemi, M. K. Mirnia and S. Shahmorad, *Solving fuzzy linear systems by using the schur complement when cofficent matrix is an M-matrix*, Iranian Journal of Fuzzy Systems, **15(3)** (2008), 15-29.

[24] A. Kauffman and M. M. Gupta, *Introduction to fuzzy arithmetic: theory and application*, Van Nostrand Reinhold, New York, 1991.

[25] M. Khorramizadeh and N. Mahdavi-Amiri, *Integer extended ABS algorithms and possible control of intermediate results for linear Diophantine systems*, 4OR: A Quarterly Journal of Operations Research, **7(2)** (2009), 145-167.

[26] M. Khorramizadeh and N. Mahdavi-Amiri, *On solving linear Diophantine systems using generalized Rosser's algorithm*, Bulletin of Iranian Mathematical Society, **34(2)** (2008), 1-25.

[27] J. Nocedal and S. J. Wright, *Numerical optimization,* Springer, 2006.

[28] E. Spedicato, E. Bodon, A. Del Popolo and N. Mahdavi-Amiri, *ABS methods and ABSPACK for linear systems and optimization: a review*, 4OR, **1** (2003), 51-66.

[29] L. A. Zadeh, *A fuzzy-set-theoretic interpretation of linguistic hedges*, Journal of Cybernetics, **2** (1972), 4-34.

[30] L. A. Zadeh, *The concept of the linguistic variable and its application to approximate reasoning*, Information Sciences, **8** (1975), 199-249.

[31] H. J. Zimmermann, *Fuzzy set theory and its applications*, Kluwer Academic Press, Dordrecht, 1991.

R. Ezzati, Department of Mathematics, Karaj Branch, Islamic Azad University, 31485 − 313, Karaj, Iran
E-mail address: ezati@kiau.ac.ir

S. Khezerloo, Department of Mathematics, Karaj Branch, Islamic Azad University, 31485 − 313, Karaj, Iran
E-mail address: S_khezerloo@yahoo.com

Z. Valizadeh, Department of Mathematics, Karaj Branch, Islamic Azad University, 31485 − 313, Karaj, Iran
E-mail address: z_valizadeh@kiau.ac.ir

N. Mahdavi-Amiri*, Department of Mathematical Sciences, Sharif University of Technology, 1458 − 889694, Tehran, Iran
E-mail address: nezamm@sina.sharif.edu

∗Corresponding author