Solution of General Dual Fuzzy Linear Systems

Using ABS Algorithm

M. A. Fariborzi Araghi* and M. M. Hosseinzadeh

Department of Mathematics, Islamic Azad University
Central Tehran Branch, P.O. Box 13185.768, Tehran, Iran
*e-mail: fariborzi.araghi@gmail.com, m_fariborzi@iauctb.ac.ir

Abstract

The main aim of this paper intends to discuss the solution of general dual fuzzy linear system (GDFLS) $AX + F = BX + C$ where $A, B$ are real $m \times n$ matrix, $F$ and $C$ are fuzzy vectors, and the unknown vector $X$ is a vector consisting of $n$ fuzzy numbers, by a special algorithm based on a class of ABS algorithms called Huang algorithm. In special case, we apply the proposed algorithm for the solution of fuzzy linear system $AX = B$ where elements of $A$ is crisp and $B$ and $X$ are fuzzy vectors. Numerical examples show the efficiency of using ABS algorithm.

Keywords: Fuzzy linear systems, General dual fuzzy linear system, Fuzzy number, Fuzzy arithmetic, ABS algorithm, Huang algorithm.

1. Introduction

The concept of fuzzy numbers and fuzzy arithmetic operations were first introduced by Zadeh [16] and Dubois and Prade [5]. Some information on fuzzy numbers and fuzzy arithmetic can be found in Kanfmann [11]. One of the important topics in fuzzy set theory is to solve fuzzy linear systems of equations. Fuzzy Linear systems of equations play a major role in several applications in various areas such as economics, physics, statistics, engineering, finance and social sciences.

Friedman et al. in [8] proposed a general model for solving fuzzy linear systems by using the embedding approach. Then in [7], they used the parametric form of fuzzy numbers and replaced the original fuzzy $n \times n$ linear system by a crisp $2n \times 2n$ linear
system and studied duality in fuzzy linear systems $AX = BX + Y$ where $A, B$ are real $n \times n$ matrix, the unknown vector $X$ is vector consisting of $n$ fuzzy numbers and the constant $Y$ is vector consisting of $n$ fuzzy numbers. Wang et al. in [15] proposed an iterative algorithm for solving dual linear system of the form $X = AX + U$ where $A$ is real $n \times n$ matrix, the unknown vector $X$ and the constant $U$ is vectors consisting of fuzzy numbers. Muzziloi and Reynaerts [13] mentioned fuzzy linear systems of the form $A_1 x + b_1 = A_2 x + b_2$ with $A_1, A_2$ square matrices of fuzzy coefficients and $b_1, b_2$ fuzzy number vectors. Abbasbandy et al. [1], considered the existence of a minimal solution of general dual fuzzy linear systems of the form $AX + F = BX + C$ where $A, B$ are real $m \times n$ matrix, $F$ and $C$ are fuzzy vectors, and the unknown vector $X$ is vector consisting of $n$ fuzzy numbers. Xu-dong Sun and Si-zong [4], presented general fuzzy linear systems of the form $Y AX = 0$ and general dual fuzzy linear systems of the form $Z BX + Y AX = 0$ with $A, B$ matrices of crisp coefficients and $Z, Y$ fuzzy number vectors.

In section 2, we introduce some main definitions and theorems in fuzzy sets theory. In section 3, we recall the ABS algorithms. In section 4, we propose an algorithm based on the Huang algorithm for finding the solution of a duality fuzzy linear systems $AX + F = BX + C$ where $A$ and $B$ are real $m \times n$ matrices, $F$ and $C$ are fuzzy vectors, and the unknown vector $X$ is a vector consisting of $n$ fuzzy numbers. Then, by using the Hausdorff distance we compare the numerical solution obtained from the proposed algorithm with analytical solution in the sample example.

2. Preliminaries

**Definition 2.1.** [17]. A fuzzy number $a$ is of LR-type if there exist shape functions $L$ (for left), $R$ (for right) and scalars $\alpha > 0, \beta > 0$ with

$$
\mu_a(x) = \begin{cases} 
L \left( \frac{m - x}{\alpha} \right), & x \leq m \\
R \left( \frac{x - m}{\beta} \right), & x \geq m
\end{cases}
$$

$a = (m, \alpha, \beta)_{LR}$ is a triangular fuzzy number if $L = R = \max(0, 1 - x)$. A popular fuzzy number is a triangular fuzzy number $a = (m, \alpha, \beta)$ where $m$, is a real number, and $\alpha, \beta$ are called the left and right spreads, respectively. If $\alpha' = m - \alpha$ and $\beta' = m + \beta$ then we can also use the notation $a = (m, \alpha', \beta')$ in order to show the fuzzy number $a$.

**Definition 2.2.** A fuzzy number $a$ is called positive (negative), denoted by $m > 0 (m < 0)$, if its membership function $\mu_a(x)$ satisfies $\mu_a(x) = 0, \forall x \leq 0 (\forall x \geq 0)$.
Solution of general dual fuzzy linear systems

**Theorem 2.3.** [17]. Let \( x = (m, \alpha, \beta) \), \( y = (n, \gamma, \delta) \) and \( \lambda \in IR \). Then

(a) \( \lambda \otimes (m, \alpha, \beta) = \begin{cases} \lambda m, \lambda \alpha, \lambda \beta & \lambda > 0 \\ \lambda m - \lambda \beta, -\lambda \alpha & \lambda < 0 \end{cases} \)

(b) \( x + y = (m + n, \alpha + \gamma, \beta + \delta) \)

(c) \( x - y = (m - n, \alpha + \delta, \beta + \gamma) \)

**Definition 2.4.** [10]. We represent an arbitrary fuzzy number by an ordered pair of functions \((u(r), \overline{u}(r))\), \(0 \leq r \leq 1\) which satisfy the following requirements:

1. \( u(r) \) is a bounded left continuous non-decreasing function over \([0,1]\),
2. \( \overline{u}(r) \) is a bounded left continuous non-increasing function over \([0,1]\),
3. \( u(r) \leq \overline{u}(r) \), \(0 \leq r \leq 1\).

The set of all these fuzzy numbers is denoted by \( E \) which is a complete metric space with Hausdorff distance. A crisp number \( a \) is simply represented by \( u(r) = \overline{u}(r) = a \), \(0 \leq r \leq 1\).

**Definition 2.5.** Let \( x = \left( \underline{x}(r), \overline{x}(r) \right) \) and \( y = \left( \underline{y}(r), \overline{y}(r) \right) \) be two given fuzzy numbers in \( E \) which is a complete metric space. Then the Hausdorff distance \( D(x, y) \) between \( x \) and \( y \) is defined as

\[
D(x, y) = \sup \max \left\{ \left| \underline{x}(r) - \underline{y}(r) \right|, \left| \overline{x}(r) - \overline{y}(r) \right| \right\}, \quad 0 \leq r \leq 1. \tag{1}
\]

**Definition 2.6.** The \( m \times n \) linear system

\[
\begin{align*}
    a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= y_1, \\
    a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= y_2, \\
    &\vdots \\
    a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= y_m,
\end{align*}
\]  

(2)

where the given matrix of coefficients \( A = (a_{ij}) \), \( 1 \leq i \leq m \) and \( 1 \leq j \leq n \) is a real \( m \times n \) matrix, the given \( y_j \in E, 1 \leq i \leq m \), with the unknowns \( x_j \in E, 1 \leq j \leq n \) is called a general fuzzy linear system (GFLS).

For arbitrary fuzzy numbers \( x = \left( \underline{x}(r), \overline{x}(r) \right) \), \( y = \left( \underline{y}(r), \overline{y}(r) \right) \) and real number \( k \), we may define the addition and the scalar multiplication of fuzzy numbers by using the extension principle as [12]

(a) \( x = y \) if and only if \( \underline{x}(r) = \underline{y}(r) \) and \( \overline{x}(r) = \overline{y}(r) \)

(b) \( x + y = (\underline{x}(r) + \underline{y}(r), \overline{x}(r) + \overline{y}(r)) \)

(c) \( x - y = (\underline{x}(r) - \underline{y}(r), \overline{x}(r) - \overline{y}(r)) \)
(d) $kx = \begin{cases} (k\overline{x}, k\overline{x}) & , k \geq 0 \\ (k\overline{x}, k\overline{x}) & , k < 0 \end{cases}$

**Definition 2.7.** [8] A fuzzy number vector $(x_1, x_2, \ldots, x_n)'$ given by $x_j = (\underline{x}_j, \overline{x}_j)$; $1 \leq j \leq n$ , $0 \leq r \leq 1$, is called a solution of the GFLS (2) if

$$\sum_{j=1}^{n} a_{ij} x_j = \sum_{j=1}^{n} a_{ij} x_j = y_i , \quad 1 \leq i \leq m$$

$$\sum_{j=1}^{n} a_{ij} x_j = \sum_{j=1}^{n} a_{ij} x_j = \overline{y}_i .$$

**Definition 2.8.** [1] The general dual fuzzy linear system (GDFLS) is defined as

$$AX + F = BX + C ,$$

where $A = (a_{ij}), B = (b_{ij}), 1 \leq i \leq m, 1 \leq j \leq n$ , are crisp coefficient matrices with nonnegative elements, and $C$ and $F$ are fuzzy number vectors. From (3), we can write

$$\sum_{j=1}^{n} a_{ij} x_j + f_i = \sum_{j=1}^{n} b_{ij} x_j + c_i .$$

Since $a_{ij} \geq 0$ and $b_{ij} \geq 0$ for all $i, j$ then

$$\sum_{j=1}^{n} a_{ij} \overline{x}_j + \overline{f}_i = \sum_{j=1}^{n} b_{ij} \overline{x}_j + \overline{c}_i , \quad \sum_{j=1}^{n} a_{ij} \underline{x}_j + \underline{f}_i = \sum_{j=1}^{n} b_{ij} \underline{x}_j + \underline{c}_i .$$

It follows that

$$\sum_{j=1}^{n} (a_{ij} - b_{ij}) x_j = c_i - f_i$$

$$\sum_{j=1}^{n} (a_{ij} - b_{ij}) \overline{x}_j = \overline{c}_i - \overline{f}_i$$

$$\sum_{j=1}^{n} (a_{ij} - b_{ij}) \underline{x}_j = \underline{c}_i - \underline{f}_i$$

Therefore, the following (GFLS) is obtained

$$PX = Q ,$$

where the elements $P = (p_{ij})$ and $Q = (q_i)$ are:

$$p_{ij} = a_{ij} - b_{ij} \quad \text{and} \quad q_i = c_i - f_i \quad , \quad 1 \leq i \leq m, \quad 1 \leq j \leq n$$

3. **ABS algorithm for solving linear system of equations**

ABS methods were introduced by Abaffy et al. [2,3]. The ABS algorithm contains direct iterative methods for computing the general solution of linear systems, linear least
Solution of general dual fuzzy linear systems

167

squares, nonlinear equations, diophantine equations and optimization problems. ABS algorithm is used, for solving a $m$ linear equations and $n$ unknowns with $m \leq n$ [2,6,9].

Algorithm 3.1. ABS algorithm

**Step 1:** Let $x_i \in IR^n$ be arbitrary, and $H_i \in IR^{m \times n}$ be nonsingular arbitrary. Set $i = 1$, $r_i = 0$.

**Step 2:** Compute $\tau_i = a_i^T x_i - b_i$ and $s_i = H_i a_i$.

**Step 3:** If $s_i = 0$ and $r_i = 0$ then set $x_{i+1} = x_i$, $H_{i+1} = H_i$, $r_{i+1} = r_i$ and go to step 7 (the $i$-th equation is redundant). If $s_i = 0$ and $r_i \neq 0$ then stop (the $i$-th equation and hence the system is incompatible).

**Step 4:** If $s_i \neq 0$ Compute the search direction $p_i = H_i^T z_i$, where $z_i \in IR^n$ is an arbitrary vector satisfying $z_i^T H_i a_i = z_i^T s_i \neq 0$. Compute $\alpha_i = \tau_i / a_i^T p_i$ and set $x_{i+1} = x_i - \alpha_i p_i$.

**Step 5:** Update $H_i$ to $H_{i+1}$ by

\[ H_{i+1} = H_i - \frac{H_i a_i w_i^T H_i}{w_i^T H_i a_i}, \]

where $w_i \in IR^n$ is an arbitrary vector satisfying $w_i^T s_i \neq 0$.

**Step 6:** Set $r_{i+1} = r_i + 1$.

**Step 7:** If $i = m$ then stop ($x_{m+1}$ is a solution) else set $i = i + 1$ and go step 2.

We note that after the completion of the algorithm, the general solution of system, if compatible, is written as

\[ x = x_{m+1} + H_i^T q, \]

where $q \in IR^n$ is arbitrary.

One of the methods based upon the idea of solving at the $i$-th step the first $i$ equations is Huang method. The Huang algorithm is a special algorithm of the class of ABS algorithms, with choosing the special parameters [10].

Algorithm 3.2. Huang algorithm

**Step 1:** Let $x_i \in IR^n$ be arbitrary, and $H_i = I$, set $i = 1$, $r_i = 0$.

**Step 2:** Compute $\tau_i = a_i^T x_i - b_i$ and $s_i = H_i a_i$.

**Step 3:** If $s_i = 0$ and $r_i = 0$ then set $x_{i+1} = x_i$, $H_{i+1} = H_i$, $r_{i+1} = r_i$ and go to step 7 (the $i$-th equation is redundant). If $s_i = 0$ and $r_i \neq 0$ then stop (the $i$-th equation and hence the system is incompatible).

**Step 4:** Compute the search direction $p_i = H_i^T z_i$, with $z_i = a_i$ satisfying $z_i^T H_i a_i \neq 0$.

Compute $\alpha_i = \tau_i / a_i^T p_i$ and set $x_{i+1} = x_i - \alpha_i p_i$.

**Step 5:** Update $H_i$ to $H_{i+1}$ by
\[ H_{i+1} = H_i - H_ia_iw_i^TH_i, \]

where \( w_j = \frac{a_i}{a_i^TH_ia_i} \) such that \( w_i^TH_ia_i = 1 \).

**Step 6:** Set \( r_{i+1} = r_i + 1 \).

**Step 7:** If \( i = m \) then stop (\( x_{m+1} \) is a solution) else set \( i = i + 1 \) and go step 2.

### 4. The proposed algorithm and numerical examples

Now, we introduce the following algorithm in order to solve (3). In the algorithm, \( d \) is a positive given number like \( d = 0.1 \) and \( x_j = (m_j, \alpha_j, \beta_j), \ 1 \leq j \leq n \) are the unknowns of the (GFLS) in (5). Also, \( q_j = (q_{m_j}, q_{\alpha_j}, q_{\beta_j}), \ 1 \leq j \leq n \) are the element of \( q \).

**Algorithm 4.1.**

**Step 1:** Convert the given (GDFLS) \( AX + F = BX + C \) to (GLFS) \( PX = Q \) according to (4) and (5).

**Step 2:** For \( r = 0(d)1 \) do the following steps.

1. a) Solve the crisp \( m \times n \) system \( PM = q_m \) where \( M = (m_1, m_2, \ldots, m_n)^T \) and \( q_m = (q_{m_j}), \ 1 \leq j \leq n \) by using Huang algorithm to obtain \( M \).

2. b) Solve the crisp \( 2m \times 2n \) linear system \( PS = (q_{\alpha}, q_{\beta})^T \) where \( S = (\alpha_j, \beta_j)^T \), \( q_\alpha = (q_{\alpha_j}), \ q_\beta = (q_{\beta_j}), \ 1 \leq j \leq n \) by using Huang algorithm 3.2 to obtain \( S \).

**Step 3:** Find the Hausdorff distance \( D(x^{ABS}, x^{exact}) \) according to (1).

**Step 4:** Write \( x_j^{ABS} = (m_j, \alpha_j, \beta_j), \ 1 \leq j \leq n \) and \( D(x^{ABS}, x^{exact}) \).

The following example is solved by applying the algorithm 4.1.

**Example 4.1.** Consider the following \( 2 \times 3 \) general dual fuzzy linear system

\[
\begin{align*}
4x_1 + x_2 + 3x_3 + (2r + 1, 4 - r) &= x_1 + 2x_2 + x_3 + (3r + 1, 6 - 2r) \\
2x_1 + 3x_2 + 2x_3 + (r, 2 - r) &= x_1 + x_2 + x_3 + (2r + 1, 5 - 2r)
\end{align*}
\]

If we implement the first step of the algorithm 4.1, we have

\[
\begin{align*}
3x_1 - x_2 + 2x_3 &= (r, 2 - r) \\
x_1 + 2x_2 + x_3 &= (r + 1, 3 - r)
\end{align*}
\]

In this example, \( d = 0.1, \ m = 2, \ n = 3 \). In the Table 1, we consider the notation \( a = (m, \alpha', \beta') \), where \( \alpha' = m - \alpha \) and \( \beta' = m + \beta \) for fuzzy numbers.

The results of algorithm 4.1 are demonstrated in table 1.
Table 1: Results of example 4.1.

<table>
<thead>
<tr>
<th>r</th>
<th>$x_1^{\text{ABS}}$</th>
<th>$x_2^{\text{ABS}}$</th>
<th>$x_3^{\text{ABS}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0.3333,0.2190,0.4476)</td>
<td>(0.6667,0.2952,1.0381)</td>
<td>(0.3333,0.1905,0.4762)</td>
</tr>
<tr>
<td>0.1</td>
<td>(0.3333,0.2305,0.4362)</td>
<td>(0.6667,0.3324,1.0010)</td>
<td>(0.3333,0.2048,0.4619)</td>
</tr>
<tr>
<td>0.2</td>
<td>(0.3333,0.2419,0.4248)</td>
<td>(0.6667,0.3695,0.9638)</td>
<td>(0.3333,0.2190,0.4476)</td>
</tr>
<tr>
<td>0.3</td>
<td>(0.3333,0.2533,0.4133)</td>
<td>(0.6667,0.4067,0.9267)</td>
<td>(0.3333,0.2333,0.4333)</td>
</tr>
<tr>
<td>0.4</td>
<td>(0.3333,0.2648,0.4019)</td>
<td>(0.6667,0.4438,0.8895)</td>
<td>(0.3333,0.2476,0.4190)</td>
</tr>
<tr>
<td>0.5</td>
<td>(0.3333,0.2762,0.3905)</td>
<td>(0.6667,0.4810,0.8524)</td>
<td>(0.3333,0.2619,0.4048)</td>
</tr>
<tr>
<td>0.6</td>
<td>(0.3333,0.2876,0.3790)</td>
<td>(0.6667,0.5181,0.8152)</td>
<td>(0.3333,0.2762,0.3905)</td>
</tr>
<tr>
<td>0.7</td>
<td>(0.3333,0.2990,0.3676)</td>
<td>(0.6667,0.5552,0.7781)</td>
<td>(0.3333,0.2905,0.3762)</td>
</tr>
<tr>
<td>0.8</td>
<td>(0.3333,0.3105,0.3562)</td>
<td>(0.6667,0.5924,0.7410)</td>
<td>(0.3333,0.3048,0.3619)</td>
</tr>
<tr>
<td>0.9</td>
<td>(0.3333,0.3219,0.3448)</td>
<td>(0.6667,0.6295,0.7038)</td>
<td>(0.3333,0.3190,0.3476)</td>
</tr>
<tr>
<td>1</td>
<td>(0.3333,0.3333,0.3333)</td>
<td>(0.6667,0.6667,0.6667)</td>
<td>(0.3333,0.3333,0.3333)</td>
</tr>
</tbody>
</table>

In this example, $D(x^{\text{ABS}}, x^{\text{strong}}) = 0.1524$, here $x^{\text{strong}}$ is the strong solution of system mentioned in [4].

The following example at first was solved by Fridman, et al. in [8]. Then, Ghanbari and Mahdavi Amiri in [9] proposed a method based on ranking function and ABS algorithm for solving this example. In order to find the solution of this system, they presented a nonlinear programming which was solved by applying the LINGO package in order to find $\alpha_i$, $1 \leq i \leq n$. Then, they found $\beta_i$ from the given ranking function. We can find the same solution with a simpler and faster method by applying theorem 2.3 and algorithms 3.2 and 4.1.
Example 4.2 [8,9]. Consider
\[
\begin{align*}
&x_1 - x_2 = (1,1,1) \\
&x_1 + 3x_2 = (5,1,2)
\end{align*}
\]
Using theorem 2.3, we have:
\[
\begin{align*}
&m_1 - m_2 = 1 \\
&m_1 + 3m_2 = 5 \quad \text{,} \\
&\alpha_1 + \beta_2 = 1 \\
&\alpha_1 + 3\alpha_2 = 1 \\
&\beta_1 + \alpha_2 = 1 \\
&\beta_1 + 3\beta_2 = 2
\end{align*}
\]
Using algorithm 3.2, we obtain \( m_1 = 2, m_2 = 1 \) and \( \alpha_1 = 0.625, \alpha_2 = 0.125, \beta_1 = 0.875, \beta_2 = 0.375 \). Therefore, \( x = (x_1, x_2)^T \), with \( x_1 = (2,0.625,0.875) \), \( x_2 = (1,0.125,0.375) \) is the strong or exact solution obtained in [8,9].

5. Conclusion

In this paper, we found a numerical fuzzy solution of a general dual fuzzy linear system by introducing a new algorithm based on the Huang algorithm. This new idea is able to solve a (GDFLS) by a less complexity and simply way. One can use the proposed algorithm to solve a nonlinear fuzzy system of equations by converting to a (GDFLS) or (FLS).

References


Solution of general dual fuzzy linear systems


Received: August, 2011