Mathematics

AIMS Mathematics, 8(9): 21329–21358.
DOI: 10.3934/math.20231087
Received: 03 May 2023
Revised: 13 June 2023
Accepted: 16 June 2023
Published: 04 July 2023

http://www.aimspress.com/journal/Math

Research article

Numerical investigation of non-probabilistic systems using Inner Outer Direct Search optimization technique

Paresh Kumar Panigrahi[1] **and Sukanta Nayak**[2,*]

[1] Department of Mathematics, School of Advanced Sciences, VIT-AP University, Inavolu, Beside AP Secretariat, Amaravati, AP, India

[2] Department of Mathematics, School of Advanced Sciences, VIT-AP University, Inavolu, Beside AP Secretariat, Amaravati, AP, India

* **Correspondence:** Email: sukantgacr@gmail.com.

Abstract: Fuzzy systems of equations often appear while modeling physical systems with imprecisely defined parameters. Many mathematical methods are available to investigate them, but handling them is challenging due to the computational complexity and difficult implementation. As such, in this paper, the Inner-Outer Direct Search (IODS) optimization technique is extended in the fuzzy environment to solve a fuzzy system of nonlinear equations. The main purpose of the extension is to study the system variables in the presence of fuzzy information. To manage fuzziness, a fuzzy parametric form is employed in the uncertain system and controls the search process toward the optimal solution. The proposed approach of fuzzy IODS converts the fuzzy system of nonlinear equations to an unconstrained fuzzy optimization problem. Then, the unconstrained fuzzy optimization problem is studied through the IODS technique. To solve the unconstrained fuzzy optimization problem, the fuzzy objective function is minimized with the help of exploratory and pattern search approaches. These searches are performed with inner and outer computations. Then, the obtained united solution provides the desired solution which minimizes the objective function. From the same the uncertain system, variables are derived. To verify the solution and proposed algorithm, convergence analysis is performed. Three case studies are considered with only fuzzy and fully fuzzy systems, and various cases are discussed. A comparison with other methods is made to test the efficacy of the method. The proposed algorithm is coded with the help of MATLAB software, and the results are analyzed graphically. Finally, the simple procedure and computationally efficient approach may help to implement the same in many engineering and science problems that can be modeled into systems of equations.

Keywords: fuzzy set; triangular fuzzy number; unconstrained minimization problem; IODS technique; convergence
Mathematics Subject Classification: 03E72, 65K10

1. Introduction

Most science and engineering problems are governed by differential equations. Due to the complexity of the problems in real practice, often, numerical methods are adopted to study the field variables. Generally, the numerical methods convert the governing differential equations into algebraic equations. Systems of nonlinear equations are among the algebraic equations which commonly occur in mathematical models. Furthermore, the presence of uncertainty makes the system challenging to solve. As uncertainties play a vital role, it cannot be avoided, and the system needs to be investigated with uncertainty. As such, here we have considered fuzzy numbers as uncertainty.

The concept of fuzzy set theory, fuzzy numbers and their computation through fuzzy arithmetic operations were initially introduced in Dubois and Prade [1], Nahmias [2] and Zadeh [3]. As nonlinear systems of equations are common in mathematical in many engineering and science problems, the uncertain analysis of the same is essential. The uncertainties that are caused due to experimental errors, lack of knowledge and partial information of system parameters are taken as fuzzy. Hence, incorporation of these uncertainties in a system of nonlinear equations produces a fuzzy system of nonlinear equations in Jafari and Yu [4]. In this regard, many researchers presented different approaches to solve fuzzy equations. A few of the relevant research works are discussed here. In [5,6], Buckley and Qu presented an analytical approach to solve fuzzy linear and quadratic equations. In Buckley and Qu [5], the authors discussed the necessary and sufficient condition for linear and quadratic equations to have a solution, when the parameter is either real and complex. Furthermore, Buckley and Qu [6] introduced a new solution based on unified extension.

From analytical approaches, it is seen that when a fuzzy equation and/or fuzzy system of equations undergoes complicated boundary conditions and computational complexity, the numerical methods make it easier to handle. In this context, Abbasbandy and Asady [7], Abbasbandy and Ezzati [8] used Newton's method to investigate fuzzy nonlinear equations. In [9], Shokri introduced an idea of midpoint in Newton's method and then applied it to study the fuzzy nonlinear equation. Further, a fuzzy system of equations can be classified as only fuzzy and fully fuzzy system depending on the usage of fuzziness. In [10], Jafari et al. used an iterative approach to solve a fully fuzzy nonlinear system. In [11], Kajani et al. studied a dual fuzzy nonlinear system of equations, and an iterative algorithm was adapted to find the solution of the system.

There are various types of fuzzy numbers that can be taken as uncertainty. One of the fuzzy numbers is the triangular fuzzy number (TFN), which is often used by researchers. We have also considered TFN for our present research work. Using TFN, Gani and Assarudeen [12] modeled a linear programming problem to deduce the fuzzy objective value as well as the fuzzy optimal solution. It is also noted that the linear programming problem is a special case of linear systems. Now, with the TFN environment, a general framework of a nonlinear system of equations can be developed. As such, here we have considered an $n \times n$ system of nonlinear equations with fuzzy environment for our investigation. In [13], Sherbrooke and Patrikalakis solved a nonlinear system of polynomial equations, and Garlof and Smit [14] obtained the solution of a polynomial equation, a recursive subdivision expressed on Bernstein's basis. Yazdi et al. [15] modeled risk assessments of different industrial and social systems with uncertainties. [16] provides many linguistic methods and case studies, demonstrating their applicability, which will enable readers to implement them in their own risk analysis process.

The above literature review reveals that the system is more challenging to understand and solve using gradient based techniques. Therefore, direct search algorithms can be implemented for easy understanding and solving of the same. In light of this, optimization techniques based on direct search approach can be introduced to obtain the solution of fuzzy systems. One of the ideas to proceed with an optimization technique for solving a system of nonlinear equations is to transform the system into an unconstrained minimization problem, as in Deb [17] and Nayak [18]. Moreover, many researchers presented various optimization techniques for the same. Luo et al. [19] suggested a hybrid technique for solving a system of nonlinear equations that combines a chaotic optimization algorithm with the quasi-Newton method. The concept behind this approach is to look for an initial estimate that will reach the convergent areas of the quasi-Newton method. Besides this, the computation cost may rise owing to the quasi-Newton approach, demanding the development of a new methodology. Recently, Ansorena [20] considered an optimization problem regarding the work strategy at the gate of the Barcelona container port. Authors provided a technique for solving fully fuzzy linear programming (FFLP) problems using the ranking function. In Jafarian and Jafari [21], a basic stability technique is used to modify the two-step method (TSM). Bibi et al. in [22] devised an approach for solving convex quadratic programming problems with limited variables. In Nayak and Chakraverty [23], the researchers took a fuzzy system of linear equations and then gave a united solution using a limit approach of fuzzy arithmetic. As the fuzzy set is the union of intervals with various α-cuts, in [24], Nayak and Pooja gave an optimization technique to solve an interval nonlinear system with the bounded parameter. A stochastic process is another way to handle uncertainty. In this regard, Yazdi [25] considered a stochastic game theory approach to solving a system safety and reliability decision-making problems under uncertainty. Hence, many direct search techniques have been used to solve the nonlinear systems of equations that can be managed for fuzzy nonlinear systems of equations.

In addition to the solution techniques, convergence analysis of the same is essential to study. The convergence analysis guarantees the occurrence of an optimal solution. In Hough et al. [26] and Lewi [27], authors provided a pattern search convergence for the mentioned optimization problem. The general theory of pattern search is extended to a local convergence of multidimensional search algorithms in Torczo [28]. In [29], Lewis and Torczon established global convergence of a pattern search technique.

As such, we shall extend the Inner Outer Direct Search (IODS) method [24] in a fuzzy environment. The search processes are inspired by the Hooke Jeeves pattern search method in Kirgat and Surde [30]. The proposed fuzzy IODS algorithm is used to solve fuzzy nonlinear systems of equations for both only fuzzy and fully fuzzy cases. Fuzzy IODS transforms a fuzzy system of nonlinear equations into an unconstrained multivariable optimization problem. It breaks the problem into a two-step calculation, that is, inner computation and outer computations. The obtained solutions through these two steps are assembled, and using regularity principle the final solution is obtained. Further, we have established the local convergence behavior of the fuzzy IODS algorithm in the neighborhood of the local minimizer x^*. Using two example problems, the analytical, numerical and graphical convergences of the obtained solutions are depicted for validation of the proposed algorithm. Finally, the nature of the uncertain solutions is discussed in various cases. The potential real-world application of this algorithm is found in modeling risk assessment problems, control systems, structural engineering problems and fluid mechanics.

This paper is structured in the following manner. Section 2 includes the preliminaries of fuzzy numbers and arithmetic operations using fuzzy numbers. Section 3 describes the classification of a fuzzy nonlinear system of equations and its different cases. Section 4 presents the Inner-Outer Direct Search method in fuzzy environment and fuzzy IODS algorithm. In Section 5, we have analyzed the local convergence of the fuzzy IODS algorithm. Then, in Section 6, three example problems are demonstrated through the fuzzy IODS algorithm, and the solutions are reported. Through convergence analysis, it is observed that the proposed fuzzy IODS method converges and guarantees the solution.

2. Preliminaries

This section includes the fundamentals of fuzzy numbers and their arithmetic (Zimmermann [31]).

A fuzzy number $\tilde{A}$ is a convex normalized piecewise continuous fuzzy set $\tilde{A}$ on the real line R with the membership function

$$\mu_{\tilde{A}}(x): R \to [0,1], \forall x \in R. \tag{1}$$

One of the fuzzy numbers, a triangular fuzzy number (TFN), is written as

$$\tilde{A} = [a_1, a_2, a_3], \tag{2}$$

where a_1, a_2 and a_3 are the left, center and right values of the TFN $\tilde{A}$, and $a_1 \leq a_2 \leq a_3$.

Two TFNs are said to be equal if the left endpoint, center and right endpoints for both the TFNs are the same, respectively.

For example, if we take two TFNs, viz. $\tilde{A} = [a_1, a_2, a_3]$ and $\tilde{B} = [b_1, b_2, b_3]$, then they are said to be equal if $a_1 = b_1$, $a_2 = b_2$, and $a_3 = b_3$. The width of the TFN $\tilde{A}$ is defined as $w = a_3 - a_1$.

To compute TFNs, one may use the traditional arithmetic in Nayak [32] operations given below. $\tilde{A} = [a_1, a_2, a_3]$ may be transformed into an interval by using $\alpha -$ cut as follows:

$$\tilde{A} = [a_1, a_2, a_3] = [a_1 + (a_2 - a_1)\alpha, a_3 - (a_3 - a_2)\alpha], \text{ where } \alpha \in [0,1].$$

Suppose that $\tilde{A} = [\underline{A}(\alpha), \overline{A}(\alpha)]$ and $\tilde{B} = [\underline{B}(\alpha), \overline{B}(\alpha)]$ are two fuzzy numbers. Then, we have the following.

Addition:

$$\tilde{A} + \tilde{B} = \left[\underline{A}(\alpha) + \underline{B}(\alpha), \overline{A}(\alpha) + \overline{B}(\alpha)\right].$$

Subtraction:

$$\tilde{A} - \tilde{B} = \left[\underline{A}(\alpha) - \overline{B}(\alpha), \overline{A}(\alpha) - \underline{B}(\alpha)\right].$$

Multiplication:

$$\tilde{A} \times \tilde{B} = [\min\{\underline{A}(\alpha) \times \underline{B}(\alpha), \underline{A}(\alpha) \times \overline{B}(\alpha), \overline{A}(\alpha) \times \underline{B}(\alpha), \overline{A}(\alpha) \times \overline{B}(\alpha)\}, \max\{\underline{A}(\alpha) \times \underline{B}(\alpha), \underline{A}(\alpha) \times \overline{B}(\alpha), \overline{A}(\alpha) \times \underline{B}(\alpha), \overline{A}(\alpha) \times \overline{B}(\alpha)\}].$$

Division:

$$\tilde{A} \div \tilde{B} = [\min\{\underline{A}(\alpha) \div \underline{B}(\alpha), \underline{A}(\alpha) \div \overline{B}(\alpha), \overline{A}(\alpha) \div \underline{B}(\alpha), \overline{A}(\alpha) \div \overline{B}(\alpha)\}, \max\{\underline{A}(\alpha) \div \underline{B}(\alpha), \underline{A}(\alpha) \div \overline{B}(\alpha), \overline{A}(\alpha) \div \underline{B}(\alpha), \overline{A}(\alpha) \div \overline{B}(\alpha)\}],$$

$$\text{where } \underline{B}(\alpha) \neq 0 \text{ and } \overline{B}(\alpha) \neq 0.$$

A TFN is a union of intervals, that is, for each membership value $\alpha \in [0, 1]$, one gets an interval, and the union of all these intervals gives the same TFN. Hence, we may conclude that during computation, the operation takes place pointwise. Therefore, one needs to construct a function using the variable α and then compute the TFNs (Chakraborty and Guh [33]). In the next section, we will study how to solve a fuzzy system of nonlinear equations using the mentioned arithmetic operations.

3. Fuzzy system of nonlinear equations

In the following, we may rewrite the definition of a TFN.

A fuzzy number $\tilde{A} = [x_L, x_N, x_R]$ is said to be a TFN if the membership values are defined as

$$\mu_A(x) = \begin{cases} \mathrm{f_L}, & x_L \leq x \leq x_N \\ \mathrm{f_R}, & x_N \leq x \leq x_R \\ 0, & otherwise \end{cases}, \tag{3}$$

where $\mathrm{f_L}$ is the left monotonically non-decreasing function, and $\mathrm{f_R}$ is the right monotonically non-increasing function. These functions can be represented as $\mathrm{f_L} = \frac{x - x_L}{x_N - x_L}$ and $\mathrm{f_R} = \frac{x_R - x}{x_R - x_N}$.

For the computational utility, an arbitrary TFN $\tilde{A} = [x_L, x_N, x_R]$ can be transformed to a two-variable form through the following steps.

Step 1: TFN to interval form,

Step 2: Interval to crisp form.

Using α-cuts the TFN can be written as

$$\tilde{A} = [x_L, x_N, x_R] \approx [\xi_L, \xi_R], \tag{4}$$

where

$$\xi_L = x_L + \alpha(x_N - x_L) \text{ and } \xi_R = x_R + \alpha(x_N - x_R);\ \alpha \in [0, 1].$$

The crisp representation of TFN $\tilde{A}$ becomes

$$\xi_L - \beta(\xi_L - \xi_R), \beta \in [0,1]. \tag{5}$$

The above representation may be used in the following system of equations to study a fuzzy system of equations.

Consider a system of equations in matrix form

$$\widetilde{M}\tilde{x} = \tilde{b}, \tag{6}$$

where $\widetilde{M}$ is a coefficient matrix, $\tilde{b}$ is the right-side vector, and $\tilde{x}$ is the unknown vector that needs to be quantified. The compact form of the same is defined as

$$\widetilde{M} = [\tilde{a}_{ij}], i,j = 1,2, \dots n, \tilde{b} = [\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_n]^T, \text{ and } \tilde{x} = [\tilde{x}_1, \tilde{x}_2, \dots \tilde{x}_n]^T. \tag{7}$$

Here, the entries $\tilde{a}_{ij}$ and $\tilde{b}_j$ are the TFNs.

Consider a system of equations having n variables, that is,

$$\widetilde{M}_{n\times n}\widetilde{X}_{n\times 1} = \tilde{b}_{n\times 1}, \tag{8}$$

where

$$\widetilde{M}_{n\times n} = \begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ \tilde{a}_{21} & \tilde{a}_{22} & \cdots & \tilde{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \cdots & \tilde{a}_{nn} \end{bmatrix}, \widetilde{X}_{n\times n} = \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{bmatrix}, \text{ and } \tilde{b}_{n\times n} = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_n \end{bmatrix}. \tag{9}$$

The expanded forms of TFNs $\tilde{a}_{ij}$ and $\tilde{b}_j$ are

$$\tilde{a}_{ij} = [a_{L_{ij}}, a_{N_{ij}}, a_{R_{ij}}] \text{ and } \tilde{b}_j = [b_{L_j}, b_{N_j}, b_{R_j}];\ i,j = 1,2,\cdots,n.$$

Now, the fully fuzzy system of the equations can be defined as

$$\begin{bmatrix} [\tilde{a}_{L_{11}}, \tilde{a}_{N_{11}}, \tilde{a}_{R_{11}}] & [\tilde{a}_{L_{12}}, \tilde{a}_{N_{12}}, \tilde{a}_{R_{12}}] & \cdots & [\tilde{a}_{L_{1n}}, \tilde{a}_{N_{1n}}, \tilde{a}_{R_{1n}}] \\ [\tilde{a}_{L_{21}}, \tilde{a}_{N_{21}}, \tilde{a}_{R_{21}}] & [\tilde{a}_{L_{22}}, \tilde{a}_{N_{22}}, \tilde{a}_{R_{22}}] & \cdots & [\tilde{a}_{L_{1n}}, \tilde{a}_{N_{1n}}, \tilde{a}_{R_{1n}}] \\ \vdots & \vdots & \ddots & \vdots \\ [\tilde{a}_{L_{n1}}, \tilde{a}_{N_{n1}}, \tilde{a}_{R_{11}}] & [\tilde{a}_{L_{n2}}, \tilde{a}_{N_{n2}}, \tilde{a}_{R_{n2}}] & \cdots & [\tilde{a}_{L_{nn}}, \tilde{a}_{N_{nn}}, \tilde{a}_{R_{nn}}] \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{bmatrix} = \begin{bmatrix} [\tilde{b}_{L_1}, \tilde{b}_{N_1}, \tilde{b}_{R_1}] \\ [\tilde{b}_{L_2}, \tilde{b}_{N_2}, \tilde{b}_{R_2}] \\ \vdots \\ [\tilde{b}_{L_n}, \tilde{b}_{N_n}, \tilde{b}_{R_n}] \end{bmatrix}. \tag{10}$$

Case 1. In Eq (10), the entries of the coefficient matrix are TFNs. The entry $\tilde{a}_{ij}$ is defined as $[a_{L_{ij}}, a_{N_{ij}}, a_{R_{ij}}]$, and the right-side vector is b_j (crisp in nature).

The matrix representation of the Case 1 type of fuzzy system of nonlinear equations may be

$$\begin{bmatrix} [\tilde{a}_{L_{11}}, \tilde{a}_{N_{11}}, \tilde{a}_{R_{11}}] & [\tilde{a}_{L_{12}}, \tilde{a}_{N_{12}}, \tilde{a}_{R_{12}}] & \cdots & [\tilde{a}_{L_{1n}}, \tilde{a}_{N_{1n}}, \tilde{a}_{R_{1n}}] \\ [\tilde{a}_{L_{21}}, \tilde{a}_{N_{21}}, \tilde{a}_{R_{21}}] & [\tilde{a}_{L_{22}}, \tilde{a}_{N_{22}}, \tilde{a}_{R_{22}}] & \cdots & [\tilde{a}_{L_{1n}}, \tilde{a}_{N_{1n}}, \tilde{a}_{R_{1n}}] \\ \vdots & \vdots & \ddots & \vdots \\ [\tilde{a}_{L_{n1}}, \tilde{a}_{N_{n1}}, \tilde{a}_{R_{11}}] & [\tilde{a}_{L_{n2}}, \tilde{a}_{N_{n2}}, \tilde{a}_{R_{n2}}] & \cdots & [\tilde{a}_{L_{nn}}, \tilde{a}_{N_{nn}}, \tilde{a}_{R_{nn}}] \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}. \tag{11}$$

Case 2. In Eq (10), the right-side vector is $\tilde{b}_j$ TFNs. The right-side vector of $\tilde{b}_j$ is defined as $[b_{L_j}, b_{N_j}, b_{R_j}]$. The coefficient of the left side $\tilde{a}_{ij}$ is crisp in nature.

The matrix representation of the Case 2 type of fuzzy system of nonlinear equations may be

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{bmatrix} = \begin{bmatrix} [\tilde{b}_{L_1}, \tilde{b}_{N_1}, \tilde{b}_{R_1}] \\ [\tilde{b}_{L_2}, \tilde{b}_{N_2}, \tilde{b}_{R_2}] \\ \vdots \\ [\tilde{b}_{L_n}, \tilde{b}_{N_n}, \tilde{b}_{R_n}] \end{bmatrix}. \tag{12}$$

Case 3. In Eq (10), the entries of the coefficient matrix and the right-side vector are TFNs.

The matrix representation of the Case 3 type of fuzzy system of nonlinear equations may be

$$\begin{bmatrix} [\tilde{a}_{L_{11}}, \tilde{a}_{N_{11}}, \tilde{a}_{R_{11}}] & [\tilde{a}_{L_{12}}, \tilde{a}_{N_{12}}, \tilde{a}_{R_{12}}] & \cdots & [\tilde{a}_{L_{1n}}, \tilde{a}_{N_{1n}}, \tilde{a}_{R_{1n}}] \\ [\tilde{a}_{L_{21}}, \tilde{a}_{N_{21}}, \tilde{a}_{R_{21}}] & [\tilde{a}_{L_{22}}, \tilde{a}_{N_{22}}, \tilde{a}_{R_{22}}] & \cdots & [\tilde{a}_{L_{1n}}, \tilde{a}_{N_{1n}}, \tilde{a}_{R_{1n}}] \\ \vdots & \vdots & \ddots & \vdots \\ [\tilde{a}_{L_{n1}}, \tilde{a}_{N_{n1}}, \tilde{a}_{R_{11}}] & [\tilde{a}_{L_{n2}}, \tilde{a}_{N_{n2}}, \tilde{a}_{R_{n2}}] & \cdots & [\tilde{a}_{L_{nn}}, \tilde{a}_{N_{nn}}, \tilde{a}_{R_{nn}}] \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{bmatrix} = \begin{bmatrix} [\tilde{b}_{L_1}, \tilde{b}_{N_1}, \tilde{b}_{R_1}] \\ [\tilde{b}_{L_2}, \tilde{b}_{N_2}, \tilde{b}_{R_2}] \\ \vdots \\ [\tilde{b}_{L_n}, \tilde{b}_{N_n}, \tilde{b}_{R_n}] \end{bmatrix}. \tag{13}$$

The above cases will be discussed and solved through example problems in the coming sections. The next section deals with the Inner Outer Direct Search (IODS) optimization technique and its extension in the fuzzy environment.

4. Inner Outer Direct Search (IODS) optimization technique in fuzzy environment

The basic idea of the IODS optimization technique is to convert the system of equations to an unconstrained optimization problem. Then, the same can be solved through a direct search method. In a fuzzy environment, the extensions are done with the involved fuzzy numbers. In addition to the system of nonlinear equations, here we are dealing with a fuzzy system of nonlinear equations. As a result, we get a fuzzy unconstrained optimization problem which needs to be investigated through the direct search approach. Moreover, we need to minimize the fuzzy unconstrained optimization problem. The search process in IODS algorithm executes in two steps, viz., exploratory search and pattern search. However, with the inclusion of epistemic uncertainties, inner and outer calculations need to be performed [24].

For the sake of understanding, we have considered a system of two nonlinear equations, where the right-hand side values are TFN.

$$\mathcal{F}_k(x_1, x_2) = [b_{L_k}, b_{N_k}, b_{R_k}]. \tag{14}$$

Equation (14) can be transformed into the interval parametric form

$$\mathcal{F}_k(x_1, x_2) = \left[\xi_{L_k}, \xi_{R_k}\right], \tag{15}$$

where $\xi_{L_k} = b_{L_k} + \alpha_k\left(b_{N_k} - b_{L_k}\right)$ and $\xi_{R_k} = b_{R_k} - \alpha_k\left(b_{R_k} - b_{N_k}\right)$. Here, $k = 1, 2$, and $0 \leq \alpha_k \leq 1$.

In the above two nonlinear equations, the solution derives x_1 and x_2 in terms of α and β. Then, the desired solution can be achieved by substituting the values of the parameters α and β. The maximum and minimum are taken as the upper and lower bounds of the solution. This method can be extended to more variables by taking an equal number of parameters as unknown variables. The search to get the minimum vector is controlled by seven different computations: (i) left outer computation, (ii) right outer computation, (iii) center outer computation, (iv) inner computation for $\alpha = 0$ and $\beta = 0$, (v) inner computation for $\alpha = 0$ and $\beta = 1$, (vi) inner computation for $\alpha = 1$ and $\beta = 0$ and (vii) inner computation for $\alpha = 1$ and $\beta = 1$. The procedure consists of two types of searches, viz., exploratory move and pattern search, inspired by the well-known Hooke-Jeeves pattern search optimization technique [17].

Using Eq (9), the left-hand side outer computation can be depicted through the following matrix:

$$\begin{bmatrix} a_{L_{11}} & a_{L_{12}} & \cdots & a_{L_{1n}} \\ a_{L_{21}} & a_{L_{22}} & \cdots & a_{L_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{L_{n1}} & a_{L_{n2}} & \cdots & a_{L_{nn}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_{L_1} \\ b_{L_2} \\ \vdots \\ b_{L_n} \end{bmatrix}. \tag{16}$$

Using Eq (9), the right-hand side outer computation can be depicted through the following matrix:

$$\begin{bmatrix} a_{R_{11}} & a_{R_{12}} & \cdots & a_{R_{1n}} \\ a_{R_{21}} & a_{R_{22}} & \cdots & a_{R_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{R_{n1}} & a_{R_{n2}} & \cdots & a_{R_{nn}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_{R_1} \\ b_{R_2} \\ \vdots \\ b_{R_n} \end{bmatrix}. \tag{17}$$

Similarly, using the value of $\alpha = 1$ in Eq (9), the center outer computation can be shown in the following matrix:

$$\begin{bmatrix} a_{N_{11}} & a_{N_{12}} & \cdots & a_{N_{1n}} \\ a_{N_{21}} & a_{N_{22}} & \cdots & a_{N_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N_{n1}} & a_{N_{n2}} & \cdots & a_{N_{nn}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_{N_1} \\ b_{N_2} \\ \vdots \\ b_{N_n} \end{bmatrix}. \tag{18}$$

The left outer, right outer, center outer and inner computation are performed through the Inner Outer Search algorithm given below.

4.1. Fuzzy IODS algorithm

Step 1: Convert fuzzy system of nonlinear equations to fuzzy unconstrained multivariable optimization problem.

Formulate inner and outer unconstrained multivariable optimization problem.

Formulate center (α-cut is unity) unconstrained multivariable optimization problem.

Step 2: Introduce two parameters $\alpha, \beta \in \{0,1\}$ and consider it with initial values. There will be four different unconstrained multivariable optimization problems.

There will be a total of seven unconstrained multivariable optimization problems which need to be solved using the following steps.

Step 3: Choose an initial point $x^{(0)}$, variable increments Δ_i, a step reduction factor $\lambda > 1$ and a convergence parameter ϵ.

Using Steps 1 and 2, we get the objective function $\mathcal{F}(x)$, $x = (x_1, x_2, \cdots, x_n)T$.

Step 4: **Exploratory search**

Step 4.1: Calculate $\mathcal{F} = \mathcal{F}(x_i), \mathcal{F}^+ = \mathcal{F}(x_i + \Delta_i)$, and $\mathcal{F}^- = \mathcal{F}(x_i - \Delta_i)$.

Step 4.2: Find $\mathcal{F}_{min} = \min(\mathcal{F}, \mathcal{F}^+, \mathcal{F}^-)$ and set x_i resembles to F_{min}.

Step 4.3: Is $i = N$ (Number of independent variables)? If no, set $i = i + 1$ and go to Step 4.1.

Else, x is the result and go to Step 4.4.

Step 4.4: If $x \neq x^c$, x^c is the current x value, then success. Else, failure.

Step 5: **Pattern search**

Step 5.1: Choose a starting point $x^{(0)}$, variable increment Δ_i $(i = 1,2, \cdots, N)$, a step reduction factor $\lambda > 1$, and a termination parameter ε.

Set $k = 0$.

Step 5.2: Perform an exploratory move using $x^{(k)}$ as the base point. Consider x to be the result of the exploratory move. If the exploratory move is successful, set $x^{(k+1)} = x$ and proceed to Step 5.4.

Else, go to Step 5.3.

Step 5.3: Is $\|\Delta\| < \epsilon$? If yes, Terminate.

Else, set $\Delta_i = \frac{\Delta_i}{\lambda}$ for $i = 1,2, \cdots, N$ and go to Step 5.2.

Step 5.4: Set $k = k + 1$ and perform the pattern search.

The new point becomes

$$x_p^{(k+1)} = x^{(k)} + \left(x^{(k)} - x^{(k-1)}\right).$$

Step 5.5: Perform another exploratory move with $x_p^{(k+1)}$ as the best point. Let the result be $x_p^{(k+1)}$.

Step 5.6: Is $\mathcal{F}\left(x^{(k+1)}\right) < \mathcal{F}\left(x^{(k)}\right)$?

If yes, go to Step 5.4.

Else go to Step 5.3.

Step 6: Assemble all the solutions and sort it by using regularity principle.

In this above algorithm, one needs to solve seven unconstrained optimization problems. Out of these, three are solved by outer computations, viz., left, center and right unconstrained optimization problems. The other four are investigated by inner computation when initial conditions are incorporated with the values $\alpha, \beta = \{0, 1\}$. The solution vector for the same is written as α, β_{x_k}. Here,

k is the number of variables ($k = 1, 2, 3, \ldots, n$).

To understand the practical utility of the fuzzy IODS algorithm, we discuss its convergence analysis in the next section.

5. Convergence analysis of fuzzy IODS method

We have considered TFNs to analyze the convergence of the fuzzy IODS algorithm.

Definition 5.1. For a fuzzy iterative method with TFN, the field variables converge if the obtained TFNs possess both center and width convergence.

The center convergence occurs if the membership value of the TFN is unity, and the width convergence occurs when the membership value is zero.

To analyze the convergence of the proposed algorithm, we need to discuss the convergence analysis of the IODS algorithm. Hence, using Definition 5.1, we present the center convergence and then width convergence.

Theorem 1. [34] Let x be a crisp vector, and $\nabla\mathcal{F}(x)$ is Lipschitz continuous on an open neighborhood Ω of $L(x_0)$ with Lipschitz constant T. Then, there exist $\delta, \tau > 0$ such that the following holds.

If x_k gives an unsuccessful iteration and $\Delta_k < \delta$, then

$$\|\nabla\mathcal{F}(x_k)\| \leq \tau\Delta_k. \tag{19}$$

Hypothesis 1. Suppose P_k represents an exploratory search, where the objective is to minimize. Then, it can be written as $P_k = \left[c_k^1 \ \ldots \ c_k^{p_k}\right]$. Here, P_k is bounded in norm, that is, $\exists\ \mathrm{T} > 0$ such that $\forall\ \left\|c_k^i\right\| < \mathrm{T}$ for $i = 1, 2, \cdots, p_k$. Thus, $\exists\ \tau_0$ such that for any arbitrary step s_k satisfies

$$\|s_k\| \leq \tau_0\Delta_k, \tag{20}$$

where $s_k = P_k\Delta_k$.

Hypothesis 2. It is ensured that $\exists\ N$ such that Δ_k is monotonically nonincreasing $\forall\ k \geq N$. Further, it can be noted that we are not allowing increase in Δ_k after some iteration N, that is, Δ_k stays the same or decreases.

Here, our main aim is to study the behavior of pattern search in the neighborhood of an isolated local minimizer, say x_* [35]. Next, the same will be used for analyzing the behavior of $\mathcal{F}$ in the neighborhood of x_*.

Hypothesis 3. It can be assumed that $\mathcal{F}$ is twice continuously differentiable on an open ball $B(x_*, \eta)$ of x_* with radius η, $\nabla\mathcal{F}(x_*) = 0$, and that the second-order sufficiency condition $\nabla^2\mathcal{F}(x_*) > 0$ holds at x_*.

As the objective is to minimize the objective function, we assume $\nabla^2\mathcal{F}(x)$ is positive definite $\forall x \in B(x_*, \eta)$. Let ρ_{min} and ρ_{max} be the lower and upper bounds of the open ball $B(x_*, \eta)$, respectively, on the singular values of $\nabla^2\mathcal{F}(x_*)$ on $B(x_*, \eta)$.

Theorem 2. Using the Hypotheses 1–3, we get $\exists\ \eta, \tau_1 > 0$, and the following holds.
If there is an unsuccessful iteration at vector x_k, $\Delta_k < \eta$, and $\|x_k - x_*\| < \eta$, then

$$\|x_k - x_*\| \leq \tau_1\Delta_k. \tag{21}$$

Proof. Suppose $\nabla\mathcal{F}(x)$ is a continuous and differentiable function, and then using the mean value theorem [36], we may write

$$\nabla^2\mathcal{F}(\zeta) = \frac{\nabla f(x_k) - \nabla f(x_*)}{x_k - x_*}. \tag{22}$$

Equation (22) can be rewritten as

$$\nabla\mathcal{F}(x_k) - \nabla\mathcal{F}(x_*) = \nabla^2\mathcal{F}(\zeta)(x_k - x_*), \tag{23}$$

for some ζ that is a line segment placed between x_k and x_*.
Since $\nabla\mathcal{F}(x_*) = 0$ by Hypothesis 2, we get

$$\|\nabla\mathcal{F}(x_k)\| = \|\nabla^2\mathcal{F}(\zeta)(x_k - x_*)\| \geq \rho_{min}\|x_k - x_*\|, \tag{24}$$

where ρ_{min} is the lower bound of the open ball $B(x_*, \eta)$.
From Theorem 1 and $\Delta_k < \eta < \delta$,

$$\rho_{min}\|x_k - x_*\| \leq \|\nabla\mathcal{F}(x_k)\| \leq \tau\Delta_k, \tag{25}$$

where $\tau_1 = {}^{\tau}/_{\rho_{min}}$.
This proves that the set of iterations converges to x_*.
Theorem 3. If $x_k, y_k \in B(x_*, \eta)$ and $\mathcal{F}(x_k) \leq \mathcal{F}(y_k)$, then using the Lipchitz condition [37], it is observed that

$$\|x_k - x_*\| \leq \rho\|y_k - x_*\|. \tag{26}$$

Proof. Suppose, $x_k, y_k \in B(x_*, \eta)$ and $\mathcal{F}(x_k) \leq \mathcal{F}(y_k)$. With Taylor's theorem with $\nabla\mathcal{F}(x_*) = 0$, it follows that

$$\mathcal{F}(x_k) = \frac{1}{2}(x_k - x_*)^T\nabla^2\mathcal{F}(\zeta)(x_k - x_*), \tag{27}$$

$$\mathcal{F}(y_k) = \frac{1}{2}(y_k - x_*)^T\nabla^2\mathcal{F}(\xi)(y_k - x_*), \tag{28}$$

where $\zeta \in [x_*, x_k]$ and $\xi \in [x_*, y_k]$.
Using the given condition $\mathcal{F}(x_k) \leq \mathcal{F}(y_k)$ with Eqs (27) and (28), we can write

$$0 \leq \mathcal{F}(y_k) - \mathcal{F}(x_k) = \frac{1}{2}(y_k - x_*)^T\nabla^2\mathcal{F}(\xi)(y_k - x_*) - \frac{1}{2}(x_k - x_*)^T\nabla^2\mathcal{F}(\zeta)(x_k - x_*).$$

Therefore,

$$0 \leq \rho_{max}\|y_k - x_*\|^2 - \rho_{min}\|x_k - x_*\|^2.$$

Theorem 4. There exist $\delta > 0$, $\epsilon > 0$ and $\tau_2 > 0$ for which the following holds.
For $k \geq N$, if x_k is an iterate for which $\Delta_k < \delta$ and $\|x_k - x_*\| < \epsilon$, then $\forall\ n \geq k$,

$$\|x_n - x_*\| \leq \eta. \tag{29}$$

Proof. From the above theorem and hypothesis, there exists an $\epsilon > 0$, and we take $\min\{\delta, \tau, \Delta_k\} > \epsilon$. We consider the proof by induction. Take $x_{k+1} = x_k + (x_k - x_{k-1})$. Then, it can be written as $x_{k+1} = x_k + s_k$.

By Hypothesis 1, we have $\|s_k\| < \tau_0\Delta_k$, that is,

$$\|x_{k+1} - x_k\| = \|s_k\| < \tau_0\Delta_k. \tag{30}$$

Applying the triangle inequality in Eq (30), it is observed that

$$\|x_{k+1} - x_*\| \leq \|x_{k+1} - x_k\| + \|x_k - x_*\| < \tau_0\Delta_k + \epsilon < \eta. \tag{31}$$

From Theorem 3 and Eq (31), it is concluded that $x_{k+1} \in B(x_*, \eta)$. As such,

$$\|x_{k+1} - x_*\| \leq \rho\|x_k - x_*\| \leq \rho\epsilon < \eta.$$

Now, assume $k + 1 \leq m$,

$$\|x_m - x_*\| \leq \eta. \tag{32}$$

Then,

$$\|x_{m+1} - x_*\| \leq \|x_{m+1} - x_m\| + \|x_m - x_*\|. \tag{33}$$

From Hypothesis 2, one can guarantee that $\Delta_m \leq \Delta_k$ for $m \geq k$. Hence, by the algorithm and hypothesis,

$$\|x_{m+1} - x_m\| \leq \tau_0\Delta_m \leq \tau_0\Delta_k.$$

Using the induction hypothesis, $x_m \in B(x_*, \eta)$. Since, $\mathcal{F}(x_m) \leq \mathcal{F}(x_k)$, we assume that $\|x_k - x_*\| < \epsilon$, and one can get

$$\|x_m - x_*\| \leq \rho\|x_k - x_*\| \leq \rho\epsilon.$$

Thus,

$$\|x_{m+1} - x_*\| < \tau_0\Delta_k + \rho\epsilon < \eta.$$

Now suppose that $m + 1 \leq n$,

$$\|x_n - x_*\| \leq \eta. \tag{34}$$

Then,

$$\|x_{n+1} - x_*\| \leq \|x_{n+1} - x_n\| + \|x_n - x_*\|. \tag{35}$$

By Hypothesis 2, we guarantee that $\Delta_n \leq \Delta_m \leq \Delta_k$ for $n \geq m \geq k$. So, $\|x_{n+1} - x_n\| \leq \tau_0\Delta_n \leq \tau_0\Delta_m \leq \tau_0\Delta_k$.

By the induction hypothesis, $x_n \in B(x_*, \eta)$. Since $\mathcal{F}(x_n) \leq \mathcal{F}(x_m)$, we assume that $\|x_k - x_*\| < \epsilon_1$ and $\|x_m - x_*\| \leq \epsilon_2$, and then we have

$$\|x_m - x_*\| \leq \rho\|x_k - x_*\| \leq \rho\epsilon_1,$$

$$\|x_n - x_*\| \leq \rho\|x_m - x_*\| \leq \rho\epsilon_2.$$

Thus,

$$\|x_{n+1} - x_*\| \leq \|x_{n+1} - x_n\| + \|x_n - x_*\|,$$

$$\|x_{n+1} - x_*\| < \tau_0\Delta_k + \rho\epsilon_2 < \eta. \tag{36}$$

Hypothesis 4. If the set $L(x_0)$ is compact [38], then $\lim\ \inf_{k\to\infty} \Delta_k = 0$.

Proof. Using the theorems [28,39] and mentioned Hypotheses 1–3, we obtain the above local convergence of pattern search.

Theorem 5. Suppose that, given a pattern search algorithm satisfying the given hypothesis, there exists a limit point x_* of the sequence of iterates produced by an algorithm that is a local minimizer.

There exist $\tau_2 > 0$ and K such that for all $k \geq K$,

$$\|x_k - x_*\| \leq \tau_2\Delta_{n(k)}, \tag{37}$$

where $n(k)$ is the final unsuccessful iterate. Therefore, we have $\lim\limits_{k\to\infty} x_k = x_*$.

Proof. By Hypothesis 2, consider the iteration after which we stop allowing Δ_k to increase. Since x_* is the limit point of $\{x_k\}$ and $\lim_{k\to\infty} x_k = x_*$, there exists an iterate x_{k+1}, $k+1 \geq N$,

$$\Delta_{k+1} < \min\{\eta, \delta\},$$

$$\|x_{k+1} - x_*\| < \min\{\eta, \epsilon\},$$

where η, δ and ϵ are introduced in the previous hypothesis and Theorems. We have

$$\|x_k - x_*\| \leq \eta.$$

By Theorem 4, we know, $\forall\, m \geq k+1$, and using Theorem 2, we get

$$\|x_m - x_*\| \leq \tau_2 \Delta_m.$$

Similarly, from Theorem 4, we know that

$$\|x_n - x_*\| \leq \eta, \forall n \geq m+1.$$

By Theorem 2, we have

$$\|x_n - x_*\| \leq \tau_2 \Delta_n,$$

for all unsuccessful iterates x_n with $n \geq m+1$.

We can proceed with the iteration value of K being the subsequent iteration $m+1$ at which we have an unsuccessful iteration. For $k \geq K$ and Theorem 2,

$$\|x_k - x_*\| \leq c_4 \Delta_k = c_4 \Delta_{n(k)}.$$

Here, $k = n(k)$.
Simultaneously, iteration is successful for $k \geq K$, and we have $\mathcal{F}(x_k) < \mathcal{F}(x_{n(k)})$. By Theorem 4,

$$\|x_k - x_*\| \leq \rho \left\|x_{n(k)} - x_*\right\| \leq \rho c_4 \Delta_{n(k)}. \tag{38}$$

Hence, we conclude that by hypothesis $\Delta_k \to 0$, and it shows that $\lim_{k\to\infty} x_k = x_*$.

This completes the convergence proofs of IODS algorithm when the membership value is one. In other words, the above theorems ensure the center convergence of fuzzy IODS algorithm for TFNs. The second part of the convergence of fuzzy IODS is the width convergence that is discussed in the following theorem.

Theorem 6. Suppose $\{x_n\}$ and $\{y_n\}$ are two convergent sequences for left and right values of the TFN, respectively, such that $\lim_{n\to\infty} x_n \to a$ and $\lim_{n\to\infty} y_n \to b$. Suppose, η_n and ξ are the widths of the TFNs which are defined as $\eta_n = |x_n - y_n|$ and $\xi = |a - b|$. Then, $\lim_{n\to\infty} \eta_n \to \xi$.

Proof. Given $\lim_{n\to\infty} x_n \to a$ and $\lim_{n\to\infty} y_n \to b$,

let there exist $\epsilon > 0$ and integers N_1 and N_2 such that

$$d(x_n, a) < \frac{\epsilon}{2} \text{ for } n_1 > N_1 \text{ and } d(y_n, b) < \frac{\epsilon}{2} \text{ for } n_2 > N_2.$$

Then, we need to prove that $|d(x_n - y_n, a - b)| < \epsilon$.
Using the triangle inequality,

$$d(x_n, y_n) \leq d(x_n, a) + d(y_n, b) + d(a, b). \tag{39}$$

Equation (39) gives

$$d(x_n, y_n) - d(a, b) \leq d(x_n, a) + d(y_n, b) = \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon. \tag{40}$$

Similarly,

$$d(a, b) \leq d(x_n, a) + d(y_n, b) + d(x_n, y_n). \tag{41}$$

Equation (41) can be represented as

$$d(a, b) - d(x_n, y_n) \leq d(x_n, a) + d(y_n, b) = \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon. \tag{42}$$

From Eqs (40) and (42) we get

$$|d(x_n, y_n) - d(a, b)| \leq d(x_n - a) + d(y_n - b) < \epsilon \ \forall n > \max\{N_1, N_2\}. \tag{43}$$

This proves

$$|d(x_n, y_n) - d(a, b)| \leq \epsilon. \tag{44}$$

Scalability refers to the ability of an algorithm to handle a system of linear and/or nonlinear problem sizes while maintaining convergence rate and computational efficiency. As per the scalability, it is found that the performance of the algorithm gets affected by the size and nature of the system of equations in terms of time of the computations as well as the complexity.

6. Fuzzy IODS approach to solve fuzzy system of nonlinear equations

In this section, we have considered three example problems and investigated the same case-by-case using the fuzzy IODS algorithm.

Example 1. Take a fuzzy system of nonlinear equations

$$\begin{aligned} \tilde{a}_{11}x_1^2 + \tilde{a}_{12}x_2 &= \tilde{\chi}_1, \\ \tilde{a}_{21}x_1 + \tilde{a}_{22}x_2^2 &= \tilde{\chi}_2, \end{aligned} \tag{45}$$

where, $\tilde{a}_{11} = [0.4, 1, 1.4]$, $\tilde{a}_{12} = [0.6\,, 1\,, 1.6]$, $\tilde{a}_{21} = [0.7\,, 1\,, 1.5]$, $\tilde{a}_{22} = [0.5\,, 1\,, 1.7]$, $\tilde{\chi}_1 = [2.5, 5.3, 7.5]$ and $\tilde{\chi}_2 = [3.2, 6.7, 8.6]$. Assume the initial approximation $x^{(0)} = (1,1)^T$, step size $\Delta = (0.5, 0.5)^T$, and $\epsilon = 10^{-4}$. Here, the tolerance value is ϵ. It is noted that the iterations are terminated if the step size become less than the tolerance value.

The first step is to construct the outer system and then convert the Eq (45) into a fuzzy unconstrained minimization problem. The transformed fuzzy unconstrained minimization problem is formulated as

$$\mathcal{F}(x) = (\tilde{a}_{11}x_1^2 + \tilde{a}_{12}x_2 - \tilde{\chi}_1)^2 + (\tilde{a}_{21}x_1 + \tilde{a}_{22}x_2^2 - \tilde{\chi}_2)^2. \tag{46}$$

Applying the IODS algorithm, the desired solution is obtained after 13 iterations using MATLAB [40]. The solution vectors after 13 iterations $(\tilde{x}_1, \tilde{x}_2)^T$ are shown in Tables 1–6 and Figures 1–6, respectively. The width of the solution TFNs are computed with the help of outer solutions. The width of a TFN is defined as the interval length of the TFN at membership value zero. In the following, the solutions are presented case by case. Case 1 represents the fuzzy system of nonlinear equations where only coefficient matrix is fuzzy. Case 2 possesses only right-side vector fuzzy. In Case 3 both the coefficient matrix and right-side vector are fuzzy. The aim of discussing three cases is to get a complete idea of propagation of fuzzy uncertainties in only fuzzy and fully fuzzy systems of nonlinear equations.

Case 1. Here, the left-hand side is taken as fuzzy, that is, the coefficient matrix is considered as fuzzy. As such, the following coefficients are assumed for the investigation: $\tilde{a}_{11} = [0.4, 1, 1.4]$, $\tilde{a}_{12} = [0.6 , 1 , 1.6]$, $\tilde{a}_{21} = [0.7 , 1 , 1.5]$, $\tilde{a}_{22} = [0.5 , 1 , 1.7]$, $\tilde{\chi}_1 = 5.3$ and $\tilde{\chi}_2 = 6.7$. Applying the proposed fuzzy IODS algorithm, iteration-wise the inner and outer solutions are listed in Tables 1 and 2. In Table 1, the first component of the solution set $\tilde{x}_1$ is included. In Table 2, the second component of the solution set $\tilde{x}_2$ is depicted. There are four inner and three outer solutions. It is noticed that the algorithm converges with the given tolerance condition at 13 iterations.

Table 1. Solution for component $\tilde{x}_1$ of Example 1.

Iteration	x_1^L	x_1^C	x_1^R	$0,0_{x_1}$	$0,1_{x_1}$	$1,0_{x_1}$	$1,1_{x_1}$	MAX	MIN	Width
1	1	1	1	1	1	1	1	1	1	0
2	1.5	2.5	3	1.5	2	1.5	1.5	3	1.5	1.5
3	1.5	1.75	3	1.75	1.75	1.75	1.5	3	1.5	1.5
4	1.5	1.75	3	1.75	1.875	1.75	1.375	3	1.375	1.625
5	1.4375	1.75	3	1.75	1.875	1.75	1.3125	3	1.3125	1.6875
6	1.375	1.75	2.9688	1.7188	1.9062	1.75	1.3438	2.9688	1.3438	1.625
7	1.375	1.75	2.9531	1.7188	1.8906	1.75	1.3281	2.9531	1.3281	1.625
8	1.375	1.7578	2.9453	1.7344	1.8984	1.75	1.3281	2.9453	1.3281	1.6172
9	1.3789	1.7539	2.9453	1.7305	1.8945	1.75	1.3281	2.9453	1.3281	1.6172
10	1.377	1.7539	2.9453	1.7324	1.8945	1.75	1.3262	2.9453	1.3262	1.6191
11	1.3779	1.7539	2.9463	1.7344	1.8945	1.75	1.3262	2.9463	1.3262	1.6201
12	1.3774	1.7539	2.9463	1.7339	1.895	1.75	1.3257	2.9463	1.3257	1.6206
13	1.3782	1.7537	2.9465	1.7339	1.8948	1.75	1.3259	2.9465	1.3259	1.6206
End Point	1.3781	1.7539	2.9465	1.7338	1.8949	1.75	1.326	2.9465	1.326	1.6205

Table 2. Solution for component $\tilde{x}_2$ of Example 1

Iteration	x_2^L	x_2^C	x_2^R	$0,0_{x_2}$	$0,1_{x_2}$	$1,0_{x_2}$	$1,1_{x_2}$	MAX	MIN	Width
1	1	1	1	1	1	1	1	1	1	0
2	1.5	2	3	4	2	1.5	1.5	4	1.5	2.5
3	1.5	2.25	3	4	2	2.25	1.5	4	1.5	2.5
4	1.625	2.25	3	4	2	2.25	1.75	4	1.625	2.375
5	1.625	2.25	3	4.0625	2	2.25	1.75	4.0625	1.625	2.4375
6	1.6562	2.2188	3.0312	4.0625	2	2.2188	1.75	4.0625	1.6562	2.4063
7	1.6562	2.2188	3.0469	4.0625	2	2.2188	1.75	4.0625	1.6562	2.4063
8	1.6484	2.2188	3.0469	4.0547	2	2.2266	1.75	4.0547	1.6484	2.4063
9	1.6523	2.2227	3.0469	4.0547	2	2.2266	1.75	4.0547	1.6523	2.4024
10	1.6504	2.2246	3.0469	4.0527	2	2.2246	1.75	4.0527	1.6504	2.4023
11	1.6514	2.2236	3.0459	4.0518	2	2.2246	1.75	4.0518	1.6514	2.4004
12	1.6509	2.2241	3.0454	4.0518	2	2.2251	1.75	4.0518	1.6509	2.4009
13	1.6509	2.2241	3.0454	4.052	2	2.2249	1.75	4.052	1.6509	2.4011
End Point	1.6509	2.224	3.0454	4.052	2	2.2249	1.75	4.052	1.6509	2.4011

In Table 1, MIN is calculated as $\min\{x_1^L, x_1^C, x_1^R, 0{,}0_{x_1}, 0{,}1_{x_1}, 1{,}0_{x_1}, 1{,}1_{x_1}\}$, and MAX is evaluated as $\max\{x_1^L, x_1^C, x_1^R, 0{,}0_{x_1}, 0{,}1_{x_1}, 1{,}0_{x_1}, 1{,}1_{x_1}\}$. The width is defined as the difference between MAX and MIN, that is, MAX−MIN.

In Table 2, MIN is calculated as $\min\{x_2^L, x_2^C, x_2^R, 0{,}0_{x_2}, 0{,}1_{x_2}, 1{,}0_{x_2}, 1{,}1_{x_2}\}$, and MAX is evaluated as $\max\{x_2^L, x_2^C, x_2^R, 0{,}0_{x_2}, 0{,}1_{x_2}, 1{,}0_{x_2}, 1{,}1_{x_2}\}$. From Tables 1 and 2, numerically it is observed that the center solution and the width of the solution TFNs converge at 13 iterations subject to the given tolerance value. The obtaned TFN solution vector components of Example 1, Case 1 are shown in Figure 1. Based on the width of the solution iteration wise the solution components are depicted in Figure 2. From Figure 2, graphically it is seen that at 13 iterations the width of solution TFNs converges subject to the tolerance value.

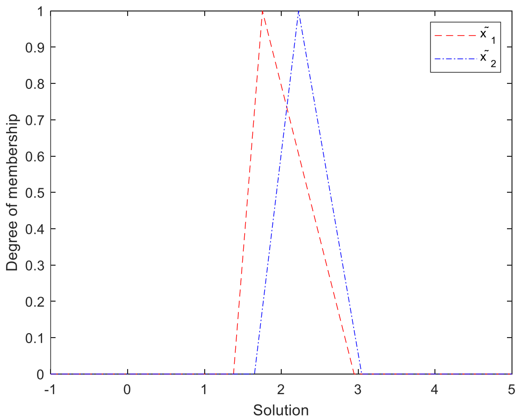

Figure 1. TFN solution of $\tilde{x}_1$ and $\tilde{x}_2$ in Case 1 of Example 1.

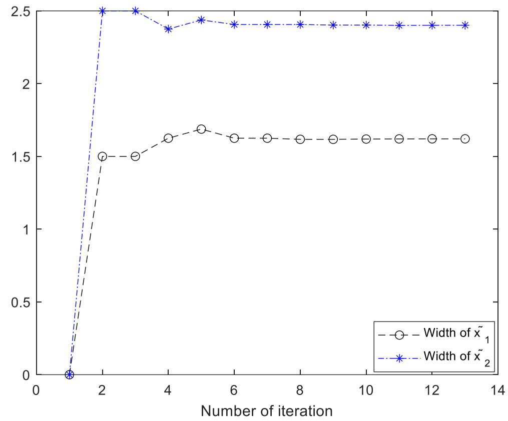

Figure 2. Uncertain width of TFN solution components $\tilde{x}_1$ and $\tilde{x}_2$ in Case 1 of Example 1.

Case 2. Here, the right-hand side is taken as fuzzy, that is, the right-side vector is considered as fuzzy. As such, the following vectors are assumed for the investigation: $\tilde{a}_{ij} = [a_{L_{ij}}, a_{N_{ij}}, a_{R_{ij}}]$ are crisp, $(i, j = 1, 2)$, $\tilde{\chi}_1 = [2.5, 5.3, 7.5]$, and $\tilde{\chi}_2 = [3.2, 6.7, 8.6]$. Applying the proposed fuzzy IODS algorithm, iteration-wise the inner and outer solutions are listed in Tables 3 and 4. In Table 3, the first component of the solution set $\tilde{x}_1$ is included. In Table 4, the second component of the solution set $\tilde{x}_2$ is depicted. There are four inner and three outer solutions. It is noticed that the algorithm converges with the given tolerance condition at 13 iterations.

In Table 3, MIN is calculated as $\min\{x_1^L, x_1^C, x_1^R, 0{,}0_{x_1}, 0{,}1_{x_1}, 1{,}0_{x_1}, 1{,}1_{x_1}\}$, and MAX is evaluated as $\max\{x_1^L, x_1^C, x_1^R, 0{,}0_{x_1}, 0{,}1_{x_1}, 1{,}0_{x_1}, 1{,}1_{x_1}\}$. The width is defined as the difference between MAX and MIN, that is, MAX−MIN.

In Table 4, MIN is calculated as $\min\{x_2^L, x_2^C, x_2^R, 0{,}0_{x_2}, 0{,}1_{x_2}, 1{,}0_{x_2}, 1{,}1_{x_2}\}$, and MAX is evaluated as $\max\{x_2^L, x_2^C, x_2^R, 0{,}0_{x_2}, 0{,}1_{x_2}, 1{,}0_{x_2}, 1{,}1_{x_2}\}$. From Tables 3 and 4, numerically it is observed that the center solution and the width of the solution TFNs converge at 13 iterations subject to the given tolerance value. The obtaned TFN solution vector components of Example 1, Case 2 are shown in Figure 3.

Based on the width of the solution iteration wise the solution components are depicted in Figure 4. From Figure 4, graphically it is seen that at 13 iterations the width of solution TFNs converges subject to the tolerance value.

Table 3. Solution for component $\tilde{x}_1$ of Example 1.

Iteration	x_1^L	x_1^C	x_1^R	$0,0_{x_1}$	$0,1_{x_1}$	$1,0_{x_1}$	$1,1_{x_1}$	MAX	MIN	Width
1	1	1	1	1	1	1	1	1	1	0
2	1.5	2.5	2.5	1.5	0.5	1.5	2.5	2.5	0.5	2
3	0.75	1.75	2.25	0.75	0.25	1.25	2.5	2.5	0.25	2.25
4	0.875	1.75	2.25	0.875	0.75	1	2.5	2.5	0.75	1.75
5	0.9375	1.75	2.25	0.9375	0.75	1	2.5	2.5	0.75	1.75
6	1.0312	1.75	2.25	1.0312	0.75	1	2.5312	2.5312	0.75	1.7812
7	1	1.75	2.2344	1	0.75	0.9844	2.5469	2.5469	0.75	1.7969
8	1.0156	1.7578	2.2344	1.0156	0.7422	0.9844	2.5508	2.5508	0.7422	1.8086
9	1.0078	1.7539	2.2305	1.0078	0.7422	0.9844	2.5508	2.5508	0.7422	1.8086
10	1.0098	1.7539	2.2305	1.0098	0.7402	0.9922	2.5498	2.5498	0.7402	1.8096
11	1.0098	1.7539	2.2305	1.0098	0.7412	0.9922	2.5498	2.5498	0.7412	1.8086
12	1.0093	1.7539	2.231	1.0093	0.7407	0.9917	2.5498	2.5498	0.7407	1.8091
13	1.01	1.7537	2.231	1.01	0.741	0.9924	2.5498	2.5498	0.741	1.8088
End Point	1.0099	1.7539	2.2307	1.0099	0.7408	0.9924	2.5498	2.5498	0.7408	1.809

Table 4. Solution for component $\tilde{x}_2$ of Example 1.

Iteration	x_2^L	x_2^C	x_2^R	$0,0_{x_2}$	$0,1_{x_2}$	$1,0_{x_2}$	$1,1_{x_2}$	MAX	MIN	Width
1	1	1	1	1	1	1	1	1	1	0
2	1	2	2.5	1	0.5	1.5	2.5	2.5	0.5	2
3	1	2.25	2.5	1.5	0.5	1.5	2.75	2.75	0.5	2.25
4	1.5	2.25	2.5	1.5	0.375	1.625	2.75	2.75	0.375	2.375
5	1.5	2.25	2.5	1.5	0.375	1.625	2.75	2.75	0.375	2.375
6	1.5	2.2188	2.5312	1.4688	0.4062	1.625	2.75	2.75	0.4062	2.3438
7	1.4688	2.2188	2.5156	1.4844	0.4062	1.5938	2.75	2.75	0.4062	2.3438
8	1.4844	2.2188	2.5234	1.4766	0.3984	1.5938	2.75	2.75	0.3984	2.3516
9	1.4766	2.2227	2.5234	1.4805	0.3984	1.5938	2.75	2.75	0.3984	2.3516
10	1.4805	2.2246	2.5234	1.4805	0.3984	1.5879	2.75	2.75	0.3984	2.3516
11	1.4805	2.2236	2.5234	1.4805	0.3994	1.5898	2.75	2.75	0.3994	2.3506
12	1.4805	2.2241	2.5234	1.4805	0.3989	1.5898	2.75	2.75	0.3989	2.3511
13	1.4805	2.2241	2.5237	1.48	0.3989	1.5894	2.75	2.75	0.3989	2.3511
End Point	1.48	2.224	2.5238	1.48	0.3989	1.5894	2.7501	2.7501	0.3989	2.3512

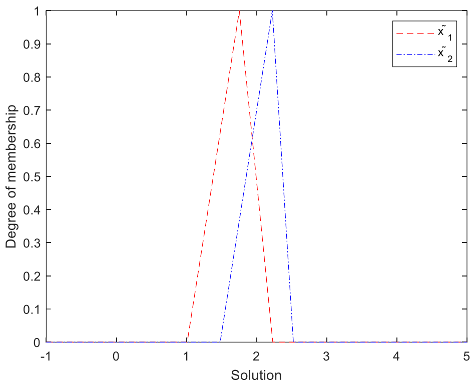

Figure 3. TFN solution of $\tilde{x}_1$ and $\tilde{x}_2$ in Case 2 of Example 1.

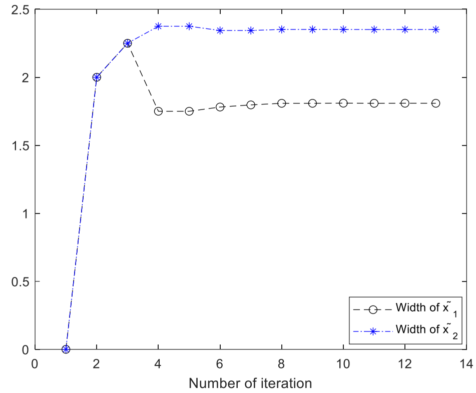

Figure 4. Uncertain width of TFN solution components $\tilde{x}_1$ and $\tilde{x}_2$ in Case 2 of Example 1.

Case 3. Here, both the left-hand and right-hand sides are taken as fuzzy, that is, the coefficient matrix and right-side vector are considered fuzzy. As such, the following coefficients matrix and right-side vector are assumed for the investigation: $\tilde{a}_{11} = [0.4, 1, 1.4]$, $\tilde{a}_{12} = [0.6\,, 1\,, 1.6]$, $\tilde{a}_{21} = [0.7\,, 1\,, 1.5]$, $\tilde{a}_{22} = [0.5\,, 1\,, 1.7]$, $\tilde{\chi}_1 = [2.5, 5.3, 7.5]$ and $\tilde{\chi}_2 = [3.2, 6.7, 8.6]$. Applying the proposed fuzzy IODS algorithm, iteration wise the inner and outer solutions are listed in Tables 5 and 6. In Table 5, the first component of the solution set $\tilde{x}_1$ is included. In Table 6, the second component of the solution set $\tilde{x}_2$ is depicted. There are four inner and three outer solutions. It is noticed that the algorithm converges with the given tolerance condition at 13 iterations.

Table 5. Solution for component $\tilde{x}_1$ of Example 1.

Iteration	x_1^L	x_1^C	x_1^R	$0,0_{x_1}$	$0,1_{x_1}$	$1,0_{x_1}$	$1,1_{x_1}$	MAX	MIN	Width
1	1	1	1	1	1	1	1	1	1	0
2	1.5	1.5	2.5	1.5	0.5	1.5	2.5	2.5	0.5	2
3	1.75	1.75	1.75	2.5	0.5	1.25	1.75	2.5	0.5	2
4	1.75	1.75	1.75	2.625	0.75	1.125	1.75	2.625	0.75	1.875
5	1.75	1.8125	1.8125	2.625	0.8125	1	1.75	2.625	0.8125	1.8125
6	1.75	1.7812	1.8438	2.6562	0.8125	1.0625	1.75	2.6562	0.8125	1.8437
7	1.75	1.7969	1.8125	2.6719	0.7969	1.0625	1.75	2.6719	0.7969	1.875
8	1.7578	1.7969	1.8125	2.5859	0.7969	1.0391	1.7578	2.5859	0.7969	1.789
9	1.7539	1.793	1.8164	2.582	0.7969	1.0391	1.7539	2.582	0.7969	1.7851
10	1.7539	1.7949	1.8164	2.582	0.7969	1.0391	1.7539	2.582	0.7969	1.7851
11	1.7539	1.7959	1.8184	2.582	0.7959	1.0361	1.7539	2.582	0.7959	1.7861
12	1.7539	1.7959	1.8184	2.5815	0.7959	1.0356	1.7539	2.5815	0.7959	1.7856
13	1.7537	1.7966	1.8181	2.5815	0.7959	1.0359	1.7537	2.5815	0.7959	1.7856
End Point	1.7539	1.7965	1.818	2.5806	0.7957	1.0359	1.7539	2.5806	0.7957	1.7849

Table 6. Solution for component $\tilde{x}_2$ of Example 1.

Iteration	x_2^L	x_2^C	x_2^R	$0,0_{x_2}$	$0,1_{x_2}$	$1,0_{x_2}$	$1,1_{x_2}$	**MAX**	**MIN**	**Width**
1	1	1	1	1	1	1	1	1	1	0
2	2	2	2	1.5	0.5	1.5	2	2	0.5	1.5
3	2	2	2.25	0.75	0.25	1.5	2.25	2.25	0.25	2
4	1.875	2	2.25	0.75	0.25	1.5	2.25	2.25	0.25	2
5	1.875	1.9375	2.25	0.75	0.3125	1.5625	2.25	2.25	0.3125	1.9375
6	1.875	1.9375	2.2188	0.7188	0.2812	1.5312	2.2188	2.2188	0.2812	1.9376
7	1.8594	1.9688	2.2188	0.7031	0.2812	1.5312	2.2188	2.2188	0.2812	1.9376
8	1.8672	1.9688	2.2188	0.8359	0.2812	1.5469	2.2188	2.2188	0.2812	1.9376
9	1.8633	1.9648	2.2227	0.8438	0.2812	1.5469	2.2227	2.2227	0.2812	1.9415
10	1.8652	1.9648	2.2246	0.8438	0.2812	1.5469	2.2246	2.2246	0.2812	1.9434
11	1.8643	1.9629	2.2236	0.8438	0.2803	1.543	2.2236	2.2236	0.2803	1.9433
12	1.8638	1.9634	2.2241	0.8442	0.2798	1.543	2.2241	2.2241	0.2798	1.9443
13	1.8638	1.9634	2.2241	0.8442	0.2798	1.543	2.2241	2.2241	0.2798	1.9443
End Point	1.8638	1.9634	2.224	0.8455	0.2798	1.5427	2.224	2.224	0.2798	1.9442

In Table 5, MIN is calculated as $\min\{x_1^L, x_1^C, x_1^R, 0{,}0_{x_1}, 0{,}1_{x_1}, 1{,}0_{x_1}, 1{,}1_{x_1}\}$, and MAX is evaluated as $\max\{x_1^L, x_1^C, x_1^R, 0{,}0_{x_1}, 0{,}1_{x_1}, 1{,}0_{x_1}, 1{,}1_{x_1}\}$. The width is defined as the difference between MAX and MIN, that is, MAX−MIN.

In Table 4, MIN is calculated as $\min\{x_2^L, x_2^C, x_2^R, 0{,}0_{x_2}, 0{,}1_{x_2}, 1{,}0_{x_2}, 1{,}1_{x_2}\}$, and MAX is evaluated as $\max\{x_2^L, x_2^C, x_2^R, 0{,}0_{x_2}, 0{,}1_{x_2}, 1{,}0_{x_2}, 1{,}1_{x_2}\}$. From Tables 5 and 6, numerically it is observed that the center solution and the width of the solution TFNs converge at 13 iterations subject to the given tolerance value. The obtaned TFN solution vector components of Example 1, Case 3 are shown in Figure 5.

Based on the width of the solution iteration-wise, the solution components are depicted in Figure 6. From Figure 6, graphically it is seen that at 13 iterations the width of solution TFNs converges subject to the tolerance value.

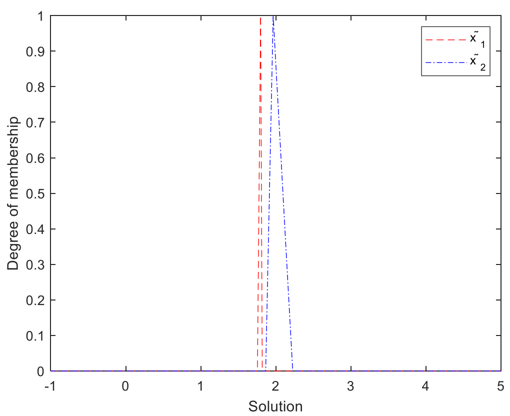

Figure 5. TFN solution of $\tilde{x}_1$ and $\tilde{x}_2$ in Case 3 of Example 1.

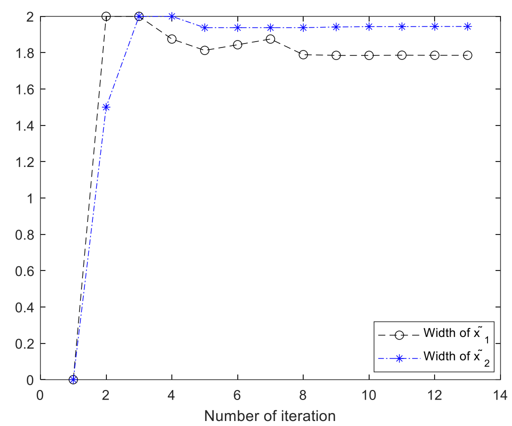

Figure 6. Uncertain width-wise solution of $\tilde{x}_1$ and $\tilde{x}_2$ in Case 3 of Example 1.

Example 2. Here, we have considered the following example to apply the fuzzy IODS algorithm. Consider a fuzzy system of nonlinear equations,

$$\tilde{a}_{11}x_1^2 + \tilde{a}_{12}x_2^2 + \tilde{a}_{13}x_3 = \tilde{\chi}_1,$$
$$\tilde{a}_{21}x_1^2 + \tilde{a}_{22}x_2 + \tilde{a}_{23}x_3^2 = \tilde{\chi}_2,$$
$$\tilde{a}_{31}x_1 + \tilde{a}_{32}x_2^2 + \tilde{a}_{33}x_3^2 = \tilde{\chi}_3, \quad (47)$$

where $\tilde{a}_{11} = [0.5, 1, 1.5\ ]$, $\tilde{a}_{12} = [0.2, 1, 1.2]$, $\tilde{a}_{13} = [0.3, 1, 1.3]$, $\tilde{a}_{21} = [0.4, 1, 1.6]$, $\tilde{a}_{22} = [0.6, 1, 1.4]$, $\tilde{a}_{23} = [0.7, 1, 1.7]$, $\tilde{a}_{31} = [0.8, 1, 1.4]$, $\tilde{a}_{32} = [0.7, 1,1.7]$, $\tilde{a}_{33} = [0.5, 1, 1.8]$, $\tilde{\chi}_1 = [4, 7, 11]$, $\tilde{\chi}_2 = [6, 9, 14]$ and $\tilde{\chi}_3 = [8, 12, 17]$.Assume the initial approximation $x^0 = (1,1,1)^T$, step size $\Delta = (0.5,0.5, 0.5)^T$, and $\epsilon = 10^{-4}$. Here, the tolerance value is ϵ. It is noted that the iterations are terminated if the step size become less than the tolerance value.

The first step is to construct the outer system and then convert the Eq (47) into a fuzzy unconstrained minimalization problem. The transformed fuzzy unconstrained minimization problem is formulated as

$$\mathcal{F}(x) = (\tilde{a}_{11}x_1^2 + \tilde{a}_{12}x_2^2 + \tilde{a}_{13}x_3 - \tilde{\chi}_1)^2 + (\tilde{a}_{21}x_1^2 + \tilde{a}_{22}x_2 + \tilde{a}_{23}x_3^2 - \tilde{\chi}_2)^2 + (\tilde{a}_{31}x_1 + \tilde{a}_{32}x_2^2 + \tilde{a}_{33}x_3^2 - \tilde{\chi}_3)^2. \quad (48)$$

Applying the IODS algorithm, the desired solution is obtained after 13 iterations using MATLAB [40]. The solution vectors after 13 iterations $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)^T$ are shown in Tables 7–15 and Figures 7–12, respectively. The width of the solution TFNs is computed with the help of outer solutions. The width of a TFN is defined as the interval length of the TFN at membership value zero. In the following, the solutions are presented case by case. Case 1 represents the fuzzy system of nonlinear equations where only the coefficient matrix is fuzzy. Case 2 has only the right-side vector fuzzy. In Case 3 both the coefficient matrix and right-side vector are fuzzy. The aim of discussing three cases is to get a complete idea of propagation of fuzzy uncertainties in only fuzzy and fully fuzzy systems of nonlinear equations.

Case 1. Here, the left-hand side is taken as fuzzy, that is, the coefficient matrix is considered as fuzzy. As such, the following coefficients are assumed for the investigation: $\tilde{a}_{11} = [0.5, 1, 1.5\,]$, $\tilde{a}_{12} = [0.2, 1, 1.2]$, $\tilde{a}_{13} = [0.3, 1, 1.3]$, $\tilde{a}_{21} = [0.4, 1, 1.6]$, $\tilde{a}_{22} = [0.6, 1, 1.4]$, $\tilde{a}_{23} = [0.7, 1, 1.7]$, $\tilde{a}_{31} = [0.8, 1, 1.4]$, $\tilde{a}_{32} = [0.7, 1,1.7]$, $\tilde{a}_{33} = [0.5, 1, 1.8]$, $\tilde{\chi}_1 = 7$, $\tilde{\chi}_2 = 9$ and $\tilde{\chi}_3 = 12$. Applying the proposed fuzzy IODS algorithm, iteration-wise the inner and outer solutions are listed in Tables 7–9. In Table 7, the first component of the solution set $\tilde{x}_1$ is included. In Table 8, the second component of the solution set $\tilde{x}_2$ is depicted. In Table 9, the third component of the solution set $\tilde{x}_3$ is represented. There are four inner and three outer solutions. It is noticed that the algorithm converges with the given tolerance condition at 13 iterations.

In Table 7, MIN is calculated as $\min\{x_1^L, x_1^C, x_1^R, 0{,}0_{x_1}, 0{,}1_{x_1}, 1{,}0_{x_1}, 1{,}1_{x_1}\}$, and MAX is evaluated as $\max\{x_1^L, x_1^C, x_1^R, 0{,}0_{x_1}, 0{,}1_{x_1}, 1{,}0_{x_1}, 1{,}1_{x_1}\}$. Width is defined as the difference between MAX and MIN, that is, MAX−MIN.

In Table 8, MIN is calculated as $\min\{x_2^L, x_2^C, x_2^R, 0{,}0_{x_2}, 0{,}1_{x_2}, 1{,}0_{x_2}, 1{,}1_{x_2}\}$, and MAX is evaluated as $\max\{x_2^L, x_2^C, x_2^R, 0{,}0_{x_2}, 0{,}1_{x_2}, 1{,}0_{x_2}, 1{,}1_{x_2}\}$. Width is defined as the difference between MAX and MIN, that is, MAX−MIN.

In Table 9, MIN is calculated as $\min\{x_3^L, x_3^C, x_3^R, 0{,}0_{x_3}, 0{,}1_{x_3}, 1{,}0_{x_3}, 1{,}1_{x_3}\}$, and MAX is evaluated as $\max\{x_3^L, x_3^C, x_3^R, 0{,}0_{x_3}, 0{,}1_{x_3}, 1{,}0_{x_3}, 1{,}1_{x_3}\}$. From Tables 7–9, numerically it is observed that the center solution and the width of the solution TFNs converge at 13 iterations subject to the given tolerance value.

The obtaned TFN solution vector components of Example 2, Case 1 are shown in Figure 7.

Based on the width of the solution iteration-wise the solution components are depicted in Figure 8. From Figure 8, graphically it is seen that at 13 iterations the width of solution TFNs converges subject to the tolerance value.

Table 7. Solution for component $\tilde{x}_1$ of Example 2.

Iteration	x_1^L	x_1^C	x_1^R	$0,0_{x_1}$	$0,1_{x_1}$	$1,0_{x_1}$	$1,1_{x_1}$	**MAX**	**MIN**	**Width**
1	1	1	1	1	1	1	1	1	1	0
2	1.5	2.5	2.5	2.5	2	1.5	1.5	2.5	1.5	1
3	0.25	1	2.75	1.75	1.5	1.25	1.25	2.75	0.25	2.5
4	0.25	0.875	2.875	1.625	1.625	1.25	1.25	2.875	0.25	2.625
5	0.25	0.875	2.875	1.5625	1.5625	1.1875	1.25	2.875	0.25	2.625
6	0.2812	0.875	2.9375	1.5625	1.5625	1.1875	1.25	2.9375	0.2812	2.6563
7	0.2656	0.8906	2.9375	1.5781	1.5781	1.1875	1.2344	2.9375	0.2656	2.6719
8	0.2969	0.8438	2.9375	1.5703	1.5703	1.1875	1.2344	2.9375	0.2969	2.6406
9	0.2969	0.8555	2.9375	1.5664	1.5703	1.1836	1.2344	2.9375	0.2969	2.6406
10	0.3047	0.8594	2.9355	1.5703	1.5703	1.1914	1.2344	2.9355	0.3047	2.6308
11	0.3057	0.8594	2.9326	1.5703	1.5713	1.1895	1.2354	2.9326	0.3057	2.6269
12	0.3042	0.8564	2.9331	1.5698	1.5713	1.1899	1.2354	2.9331	0.3042	2.6289
13	0.3042	0.8564	2.9333	1.5698	1.5713	1.1892	1.2354	2.9333	0.3042	2.6291
End Point	0.3032	0.8561	2.9333	1.5697	1.5712	1.1903	1.2352	2.9333	0.3032	2.6301

Table 8. Solution for component $\tilde{x}_2$ of Example 2.

Iteration	x_2^L	x_2^C	x_2^R	$0,0_{x_2}$	$0,1_{x_2}$	$1,0_{x_2}$	$1,1_{x_2}$	**MAX**	**MIN**	**Width**
1	1	1	1	1	1	1	1	1	1	0
2	1.5	1.5	2.5	2	2	1.5	1.5	2.5	1.5	1
3	1.75	2	2.75	2.5	2.5	2.25	2	2.75	1.75	1
4	1.75	2	3.125	2.5	2.5	2.25	2	3.125	1.75	1.375
5	1.75	2.125	3.125	2.5625	2.5625	2.25	2	3.125	1.75	1.375
6	1.75	2.125	3.1562	2.5312	2.5312	2.25	2	3.1562	1.75	1.4062
7	1.75	2.1094	3.1719	2.5312	2.5312	2.25	2	3.1719	1.75	1.4219
8	1.7266	2.1094	3.1719	2.5391	2.5391	2.25	1.9844	3.1719	1.7266	1.4453
9	1.7305	2.1094	3.1719	2.5391	2.5352	2.25	1.9844	3.1719	1.7305	1.4414
10	1.7285	2.1094	3.1738	2.5352	2.5352	2.2559	1.9844	3.1738	1.7285	1.4453
11	1.7266	2.1094	3.1729	2.5352	2.5352	2.2588	1.9844	3.1729	1.7266	1.4463
12	1.728	2.1104	3.1719	2.5356	2.5352	2.2588	1.9849	3.1719	1.728	1.4439
13	1.7275	2.1104	3.1719	2.5359	2.5352	2.259	1.9851	3.1719	1.7275	1.4444
End Point	1.7278	2.1105	3.1719	2.5358	2.5355	2.2583	1.9852	3.1719	1.7278	1.4441

Table 9. Solution for component $\tilde{x}_3$ of Example 2.

Iteration	x_3^L	x_3^C	x_3^R	$0,0_{x_3}$	$0,1_{x_3}$	$1,0_{x_3}$	$1,1_{x_3}$	MAX	MIN	Width
1	1	1	1	1	1	1	1	1	1	0
2	1.5	2	2.5	1.5	2	1.5	1.5	2.5	1.5	1
3	1.75	2.75	2.75	2	2	2	1.5	2.75	1.5	1.25
4	1.75	2.375	2.75	2	2	2	1.5	2.75	1.5	1.25
5	1.75	2.375	2.625	2	2	2	1.5	2.625	1.5	1.125
6	1.75	2.2812	2.6562	2	2	2.0625	1.5	2.6562	1.5	1.1562
7	1.75	2.2812	2.6562	2	2	2.0625	1.5781	2.6562	1.5781	1.0781
8	1.7891	2.2812	2.6484	2	2	2.0625	1.5859	2.6484	1.5859	1.0625
9	1.7812	2.2812	2.6484	2	2	2.0625	1.5859	2.6484	1.5859	1.0625
10	1.7812	2.2812	2.6465	2	2	2.0488	1.5859	2.6465	1.5859	1.0606
11	1.7832	2.2842	2.6465	2	1.998	2.0469	1.5752	2.6465	1.5752	1.0713
12	1.7832	2.2852	2.646	2	1.998	2.0469	1.5752	2.646	1.5752	1.0708
13	1.7837	2.2852	2.646	2	1.998	2.0476	1.5752	2.646	1.5752	1.0708
End Point	1.7837	2.285	2.6461	2.0001	1.998	2.0485	1.5756	2.6461	1.5756	1.0705

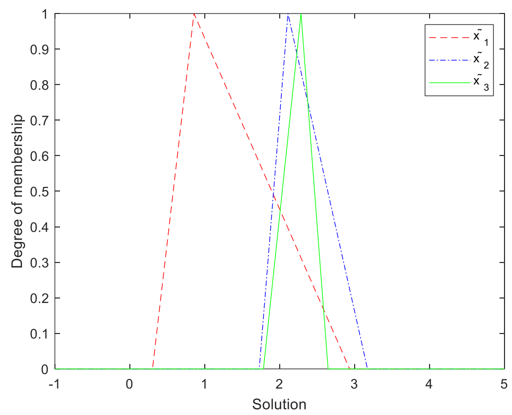

Figure7. TFN solution of $\tilde{x}_1, \tilde{x}_2$ and $\tilde{x}_3$ in Case 1 of Example 2.

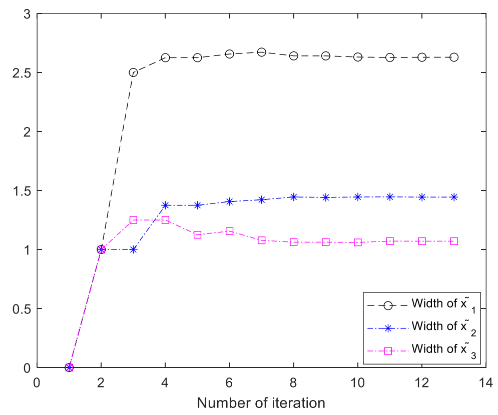

Figure 8. Uncertain width-wise solution of $\tilde{x}_1$, $\tilde{x}_2$ and $\tilde{x}_3$ in Case 1 of Example 2.

Case 2. Here, the right-hand side is taken as fuzzy, that is, the right-side vector is considered as fuzzy. As such, the following vectors are assumed for the investigation: $\tilde{a}_{ij} = [a_{L_{ij}}, a_{N_{ij}}, a_{R_{ij}}]$ are crisp, $(i, j = 1, 2, 3)$, $\tilde{\chi}_1 = [4, 7, 11]$, $\tilde{\chi}_2 = [6, 9, 14]$ and $\tilde{\chi}_3 = [8, 12, 17]$. Applying proposed fuzzy IODS algorithm, iteration-wise the inner and outer solutions are listed in Tables 10–12. In Table 10, the first component of the solution set $\tilde{x}_1$ is included. In Table 11, the second component of the solution set $\tilde{x}_2$ is depicted. In Table 12, the third component of the solution set $\tilde{x}_3$ is represented. There are four inner and three outer solutions. It is noticed that the algorithm converges with the given tolerance condition at 13 iterations.

In Table 10, MIN is calculated as $\min\{x_1^L, x_1^C, x_1^R, 0{,}0_{x_1}, 0{,}1_{x_1}, 1{,}0_{x_1}, 1{,}1_{x_1}\}$, and MAX is evaluated as $\max\{x_1^L, x_1^C, x_1^R, 0{,}0_{x_1}, 0{,}1_{x_1}, 1{,}0_{x_1}, 1{,}1_{x_1}\}$. The width is defined as the difference between MAX and MIN, that is, MAX−MIN.

In Table 11, MIN is calculated as $\min\{x_2^L, x_2^C, x_2^R, 0{,}0_{x_2}, 0{,}1_{x_2}, 1{,}0_{x_2}, 1{,}1_{x_2}\}$, and MAX is evaluated as $\max\{x_2^L, x_2^C, x_2^R, 0{,}0_{x_2}, 0{,}1_{x_2}, 1{,}0_{x_2}, 1{,}1_{x_2}\}$. The width is defined as the difference between MAX and MIN, that is, MAX$-$MIN.

Table 10. Solution for component $\tilde{x}_1$ of Example 2.

Iteration	x_1^L	x_1^C	x_1^R	$0,0_{x_1}$	$0,1_{x_1}$	$1,0_{x_1}$	$1,1_{x_1}$	MAX	MIN	Width
1	1	1	1	1	1	1	1	1	1	0
2	1.5	2.5	2.5	1.5	2.5	1.5	1.5	2.5	1.5	1
3	0	0.25	1.25	0.75	1.25	0	0.25	1.25	0	1.25
4	0.125	0.25	1.375	0.5	0.75	0.375	0.5	1.375	0.125	1.25
5	0.25	0.25	1.4375	0.375	0.8125	0.4375	0.5625	1.4375	0.25	1.1875
6	0.2812	0.2812	1.4375	0.3125	0.8438	0.4375	0.5312	1.4375	0.2812	1.1563
7	0.2656	0.3125	1.3594	0.3125	0.7656	0.4531	0.2031	1.3594	0.2031	1.1563
8	0.2969	0.3047	1.3828	0.1016	0.7656	0.4453	0.2266	1.3828	0.1016	1.2812
9	0.2969	0.3281	1.3867	0.1055	0.7656	0.4492	0.2305	1.3867	0.1055	1.2812
10	0.3047	0.3223	1.3848	0.1035	0.7695	0.4492	0.4883	1.3848	0.1035	1.2813
11	0.3057	0.3232	1.3848	0.1025	0.7793	0.4531	0.4854	1.3848	0.1025	1.2823
12	0.3042	0.3223	1.3843	0.1064	0.7798	0.4531	0.4634	1.3843	0.1064	1.2779
13	0.3042	0.3225	1.3843	0.1064	0.78	0.4529	0.4636	1.3843	0.1064	1.2779
End Point	0.3032	0.3226	1.3846	0.1064	0.78	0.4526	0.4637	1.3846	0.1064	1.2782

Table 11. Solution for component $\tilde{x}_2$ of Example 2.

Iteration	x_2^L	x_2^C	x_2^R	$0,0_{x_2}$	$0,1_{x_2}$	$1,0_{x_2}$	$1,1_{x_2}$	MAX	MIN	Width
1	1	1	1	1	1	1	1	1	1	0
2	1.5	1.5	2.5	1.5	1.5	1.5	1.5	2.5	1.5	1
3	1.5	2	2.25	2	2	1.5	1.25	2.25	1.25	1
4	1.5	2	2.375	1.875	2.375	1.25	2.25	2.375	1.25	1.125
5	1.4375	2.125	2.375	1.9375	2.4375	1.4375	2.25	2.4375	1.4375	1
6	1.4375	2.125	2.4375	1.9375	2.4375	2.5	2.3125	2.5	1.4375	1.0625
7	1.4531	2.1094	2.4531	1.9375	2.5156	2.4844	2.4062	2.5156	1.4531	1.0625
8	1.4453	2.1094	2.4453	2.1406	2.5156	2.4766	2.3984	2.5156	1.4453	1.0703
9	1.4492	2.1094	2.4453	2.1445	2.5156	2.4766	2.3945	2.5156	1.4492	1.0664
10	1.4512	2.1094	2.4434	2.1406	2.5098	2.4766	2.4473	2.5098	1.4512	1.0586
11	1.4512	2.1094	2.4453	2.1406	2.5068	2.4766	2.4502	2.5068	1.4512	1.0556
12	1.4507	2.1104	2.4453	2.1396	2.5068	2.4766	2.457	2.5068	1.4507	1.0561
13	1.4507	2.1104	2.4451	2.1396	2.5068	2.4768	2.457	2.5068	1.4507	1.0561
End Point	1.4507	2.1105	2.4449	2.1396	2.5068	2.4767	2.4572	2.5068	1.4507	1.0561

In Table 12, MIN is calculated as $\min\{x_3^L, x_3^C, x_3^R, 0{,}0_{x_3}, 0{,}1_{x_3}, 1{,}0_{x_3}, 1{,}1_{x_3}\}$, and MAX is evaluated as $\max\{x_3^L, x_3^C, x_3^R, 0{,}0_{x_3}, 0{,}1_{x_3}, 1{,}0_{x_3}, 1{,}1_{x_3}\}$. From Tables 10–12, numerically it is observed that the center solution and the width of the solution TFNs converge at 13 iterations subject to the given tolerance value.

The obtaned TFN solution vector components of Example 2, Case 2 are shown in Figure 9.

Based on the width of the solution iteration-wise, the solution components are depicted in Figure 10. From Figure 10, graphically it is seen that at 13 iterations the width of solution TFNs converges subject to the tolerance value.

Table 12. Solution for component $\tilde{x}_3$ of Example 2.

Iteration	x_3^L	x_3^C	x_3^R	$\mathbf{0,0}_{x_3}$	$\mathbf{0,1}_{x_3}$	$\mathbf{1,0}_{x_3}$	$\mathbf{1,1}_{x_3}$	**MAX**	**MIN**	**Width**
1	1	1	1	1	1	1	1	1	1	0
2	1.5	2	2.5	1.5	2	1.5	1.5	2.5	1.5	1
3	2.25	2.75	3.25	2.25	3	2.25	2	3.25	2	1.25
4	2.25	2.75	3.125	2.625	2.875	2.25	2.75	3.125	2.25	0.875
5	2.25	2.625	3.125	2.5625	2.875	2.125	2.75	3.125	2.125	1
6	2.25	2.6562	3.0938	2.5625	2.9062	3.0312	2.75	3.0938	2.25	0.8438
7	2.2344	2.6562	3.1094	2.5625	2.9688	3.0469	2.8438	3.1094	2.2344	0.875
8	2.2344	2.6484	3.1016	2.7578	2.9688	3.0469	2.8438	3.1016	2.2344	0.8672
9	2.2266	2.6484	3.1055	2.7617	2.957	3.0469	2.8477	3.1055	2.2266	0.8789
10	2.2266	2.6465	3.1055	2.7637	2.959	3.0469	2.8926	3.1055	2.2266	0.8789
11	2.2266	2.6465	3.1045	2.7637	2.9551	3.0508	2.8906	3.1045	2.2266	0.8779
12	2.2271	2.646	3.1045	2.7642	2.9551	3.0508	2.897	3.1045	2.2271	0.8774
13	2.2271	2.646	3.1045	2.7642	2.9551	3.0505	2.897	3.1045	2.2271	0.8774
End Point	2.2272	2.6461	3.1045	2.7642	2.9551	3.0505	2.8969	3.1045	2.2272	0.8773

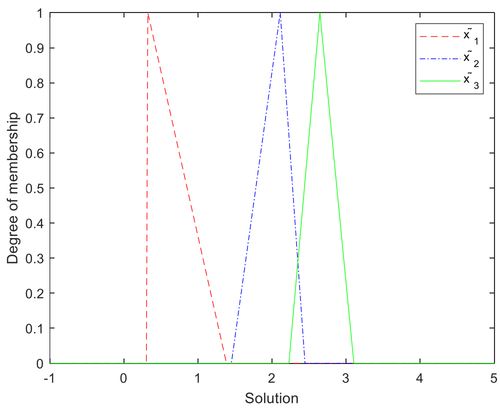

Figure 9. TFN solution of $\tilde{x}_1, \tilde{x}_2$ and $\tilde{x}_3$ in Case 2 of Example 2.

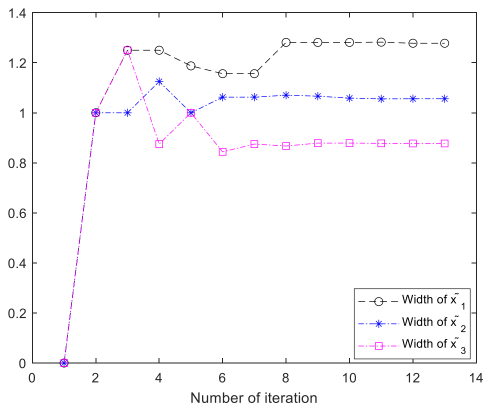

Figure 10. Uncertain width-wise solution of $\tilde{x}_1$, $\tilde{x}_2$ and $\tilde{x}_3$ Case 2 of Example 2.

Case 3. Here, the both left-hand and right-hand sides are taken as fuzzy, that is, the coefficient matrix and right-side vector are considered as fuzzy. As such, the following coefficients matrix and right-side vector are assumed for the investigation: $\tilde{a}_{11} = [0.5, 1, 1.5]$, $\tilde{a}_{12} = [0.2, 1, 1.2]$, $\tilde{a}_{13} = [0.3, 1, 1.3]$, $\tilde{a}_{21} = [0.4, 1, 1.6]$, $\tilde{a}_{22} = [0.6, 1, 1.4]$, $\tilde{a}_{23} = [0.7, 1, 1.7]$, $\tilde{a}_{31} = [0.8, 1, 1.4]$, $\tilde{a}_{32} = [0.7, 1,1.7]$, $\tilde{a}_{33} = [0.5, 1, 1.8]$, $\tilde{\chi}_1 = [4, 7, 11]$, $\tilde{\chi}_2 = [6, 9, 14]$ and $\tilde{\chi}_3 = [8, 12, 17]$. Applying the proposed fuzzy IODS algorithm, iteration-wise, the inner and outer solutions are listed in Tables 13–15. In Table 13, the first component of the solution set $\tilde{x}_1$ is included. In Table 14, the second component of the solution set $\tilde{x}_2$ is depicted. In Table 15, the third component of the solution set $\tilde{x}_3$ is represented. There are four inner and three outer solutions. It is noticed that the algorithm converges with the given tolerance condition at 13 iterations.

In Table 13, MIN is calculated as $\min\{x_1^L, x_1^C, x_1^R, 0{,}0_{x_1}, 0{,}1_{x_1}, 1{,}0_{x_1}, 1{,}1_{x_1}\}$, and MAX is evaluated as $\max\{x_1^L, x_1^C, x_1^R, 0{,}0_{x_1}, 0{,}1_{x_1}, 1{,}0_{x_1}, 1{,}1_{x_1}\}$. The width is defined as the difference between MAX and MIN, that is, MAX−MIN.

Table 13. Solution for component $\tilde{x}_1$ of Example 2.

Iteration	x_1^L	x_1^C	x_1^R	$0,0_{x_1}$	$0,1_{x_1}$	$1,0_{x_1}$	$1,1_{x_1}$	MAX	MIN	Width
1	1	1	1	1	1	1	1	1	1	0
2	2.5	2.5	2.5	1.5	3.5	1.5	2	3.5	1.5	2
3	0.25	1.25	2	1.25	2.75	1.5	2	2.75	0.25	2.5
4	0.25	1.5	2.125	1.125	2.375	1.5	2.125	2.375	0.25	2.125
5	0.25	1.5	2.0625	1.125	2.3125	1.5	2.125	2.3125	0.25	2.0625
6	0.2812	1.5	2.0625	1.1562	2.2812	1.5	2.0625	2.2812	0.2812	2
7	0.2656	1.5	2.0625	1.1406	2.3281	1.5	2.0625	2.3281	0.2656	2.0625
8	0.2969	1.5	2.0625	1.1406	2.3203	1.5078	2.0625	2.3203	0.2969	2.0234
9	0.2969	1.5	2.0742	1.1406	2.3242	1.5078	2.0625	2.3242	0.2969	2.0273
10	0.3047	1.5059	2.0723	1.1406	2.3242	1.5098	2.0625	2.3242	0.3047	2.0195
11	0.3057	1.5039	2.0781	1.1416	2.3252	1.5078	2.0625	2.3252	0.3057	2.0195
12	0.3042	1.5049	2.0781	1.1411	2.3237	1.5073	2.0625	2.3237	0.3042	2.0195
13	0.3042	1.5049	2.0779	1.1411	2.324	1.5081	2.0625	2.324	0.3042	2.0198
End Point	0.3032	1.5045	2.0776	1.141	2.324	1.5071	2.0625	2.324	0.3032	2.0208

Table 14. Solution for component $\tilde{x}_2$ of Example 2.

Iteration	x_2^L	x_2^C	x_2^R	$0,0_{x_2}$	$0,1_{x_2}$	$1,0_{x_2}$	$1,1_{x_2}$	MAX	MIN	Width
1	1	1	1	1	1	1	1	1	1	0
2	1.5	1.5	2.5	1.5	2.5	1.5	1.5	2.5	1.5	1
3	2	2	2.5	1.25	3.25	2	1.5	3.25	1.25	2
4	2	2.125	2.5	1.375	3.25	1.625	1.125	3.25	1.125	2.125
5	2	2.125	2.5	1.375	3.25	1.625	1.0625	3.25	1.0625	2.1875
6	2	2.125	2.5	1.4062	3.2188	1.625	2.3438	3.2188	1.4062	1.8126
7	2	2.1094	2.5	1.4219	3.2188	1.625	2.3438	3.2188	1.4219	1.7969
8	2.0156	2.1094	2.4922	1.4141	3.2188	1.7344	2.3438	3.2188	1.4141	1.8047
9	2.0156	2.1094	2.4883	1.4219	3.2188	1.7305	2.3438	3.2188	1.4219	1.7969
10	2.0117	2.1094	2.4941	1.4219	3.2188	1.7363	2.3438	3.2188	1.4219	1.7969
11	2.0137	2.1094	2.4932	1.4219	3.2188	1.7383	2.3516	3.2188	1.4219	1.7969
12	2.0132	2.1104	2.4941	1.4248	3.2197	1.7427	2.3516	3.2197	1.4248	1.7949
13	2.0132	2.1104	2.4941	1.4248	3.2195	1.7424	2.3516	3.2195	1.4248	1.7947
End Point	2.0129	2.1105	2.494	1.4249	3.2195	1.7455	2.3523	3.2195	1.4249	1.7946

In Table 14, MIN is calculated as $\min\{x_2^L, x_2^C, x_2^R, 0{,}0_{x_2}, 0{,}1_{x_2}, 1{,}0_{x_2}, 1{,}1_{x_2}\}$, and MAX is evaluated as $\max\{x_2^L, x_2^C, x_2^R, 0{,}0_{x_2}, 0{,}1_{x_2}, 1{,}0_{x_2}, 1{,}1_{x_2}\}$. The width is defined as the difference between

MAX and MIN, that is, MAX−MIN.

In Table 15, MIN is calculated as $\min\{x_3^L, x_3^C, x_3^R, 0{,}0_{x_3}, 0{,}1_{x_3}, 1{,}0_{x_3}, 1{,}1_{x_3}\}$, and MAX is evaluated as $\max\{x_3^L, x_3^C, x_3^R, 0{,}0_{x_3}, 0{,}1_{x_3}, 1{,}0_{x_3}, 1{,}1_{x_3}\}$. From Tables 13–15, numerically it is observed that the center solution and the width of the solution TFNs converge at 13 iterations subject to the given tolerance value.

The obtaned TFN solution vector components of Example 2, Case 3 are shown in Figure 11.

Based on the width of the solution iteration-wise, the solution components are depicted in Figure 12. From Figure 12, graphically it is seen that at 13 iterations the width of solution TFNs converges subject to the tolerance value.

Table 15. Solution for component $\tilde{x}_3$ of Example 2.

Iteration	x_3^L	x_3^C	x_3^R	$0,0_{x_3}$	$0,1_{x_3}$	$1,0_{x_3}$	$1,1_{x_3}$	MAX	MIN	Width
1	1	1	1	1	1	1	1	1	1	0
2	1.5	2	2	1.5	1.5	1.5	1.5	2	1.5	0.5
3	2	2.25	2.75	1	2	1.5	1.5	2.75	1	1.75
4	2	2	2.75	1	2.125	1.5	0.375	2.75	0.375	2.375
5	2	2.125	2.625	0.9375	2.1875	1.5	0.3125	2.625	0.3125	2.3125
6	2	2.125	2.6562	0.9375	2.2188	1.5	0.8438	2.6562	0.8438	1.8124
7	2	2.125	2.6562	0.9375	2.1875	1.5	0.8438	2.6562	0.8438	1.8124
8	2	2.1094	2.6484	0.9375	2.1875	1.3984	0.8438	2.6484	0.8438	1.8046
9	2	2.1094	2.6484	0.9336	2.1875	1.3984	0.8438	2.6484	0.8438	1.8046
10	1.9941	2.1094	2.6465	0.9336	2.1855	1.3867	0.8438	2.6465	0.8438	1.8027
11	1.9922	2.1084	2.6465	0.9336	2.1855	1.3877	0.8389	2.6465	0.8389	1.8076
12	1.9917	2.1084	2.646	0.9316	2.1855	1.3828	0.8389	2.646	0.8389	1.8071
13	1.9915	2.1086	2.646	0.9316	2.1858	1.3821	0.8389	2.646	0.8389	1.8071
End Point	1.9918	2.1089	2.6461	0.9316	2.1858	1.3771	0.8384	2.6461	0.8384	1.8077

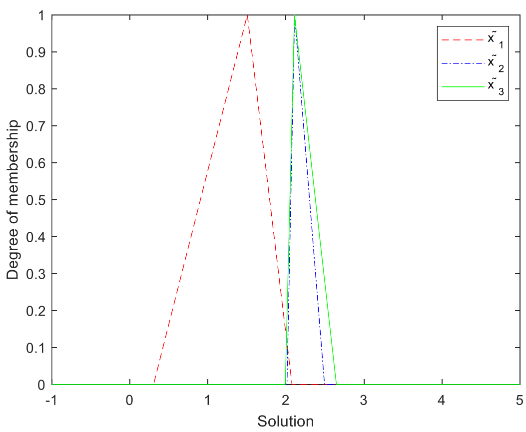

Figure 11. TFN solution of $\tilde{x}_1, \tilde{x}_2$ and $\tilde{x}_3$ in Case 3 of Example 2.

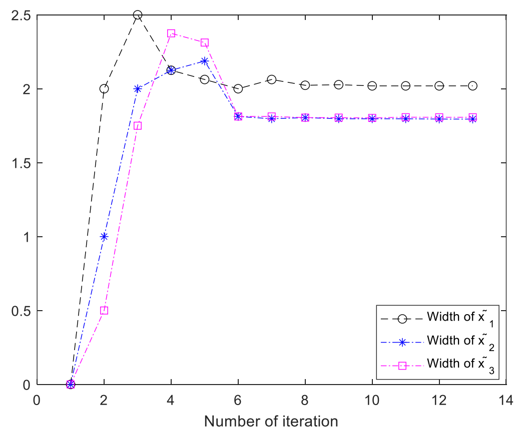

Figure 12. Uncertain solution of $\tilde{x}_1$, $\tilde{x}_2$ and $\tilde{x}_3$ Case 3 of Example 2.

Example 3. Take a fuzzy system of nonlinear equations

$$\tilde{a}_{11}x_1^2 + \tilde{a}_{12}x_2 = 10.9 + 0.1\alpha, 11.1 - 0.1\alpha,$$

$$\tilde{a}_{21}x_1 + \tilde{a}_{22}x_2^2 = 6.9 + 0.1\alpha, 7.1 - 0.1\alpha. \quad (49)$$

Assume the initial approximation $x^{(0)} = (1,1)^T$, step size $\Delta = (0.5, 0.5)^T$ and $\epsilon = 10^{-3}$. Here, the tolerance value is ϵ. It is noted that the iterations are terminated if the step size become less than the tolerance value. The first step is to construct the outer system and then convert the Eq (49) into a fuzzy unconstrained minimization problem. The transformed fuzzy unconstrained minimization problem is formulated as

$$\mathcal{F}(x) = (x_1^2 + x_2 - [10.9 + 0.1\alpha, 11.1 - 0.1\alpha])^2 + (x_1 + x_2^2 - [6.9 + 0.1\alpha, 7.1 - 0.1\alpha])^2. \quad (50)$$

Applying the IODS algorithm, the desired solution is obtained after 9 iterations using MATLAB [40]. The solution vector after 9 iterations (using tolerance value ϵ) $(\tilde{x}_1, \tilde{x}_2)^T$ with different α-cuts are shown in Figures 13 and 14. The width of a TFN is defined as the interval length of the TFN at membership value zero. For the comparison of the present method and other existing methods, in Case 2, only the right-side vector is fuzzy. The aim of discussing Case 2 with other methods is to provide a better solution about suitability of different types of fuzzy inputs.

Here, we have taken the right-side vector as fuzzy. By applying the current algorithm, the TFN solutions of $\tilde{x}_1 = [2.9869, 3, 3.0136]$ and $\tilde{x}_2 = [1.9765, 2, 2.0214]$ are graphically represented in Figures 13 and 14. Further, to the effectiveness of the proposed algorithm, the obtained TFN solutions are compared with the well-known Vertex Method, Krawczyk Method, Nayak and Chakraverty [41] and Nayak and Pooja [24], and the same is presented in Table 16.

Table 16. Comparison of solutions of $\tilde{x}_1$ and $\tilde{x}_2$ of Example 3 with the existing method [24].

Solution	Vertex Method	Krawczyk method	Nayak and Chakraverty [24]	Nayak and Pooja [41]	Fuzzy IODS
$\tilde{x}_1$	[2.9782,3.0217]	[2.744592,3.255408]	[2.9782, 3.0217]	[2.9160, 3.0137]	[2.9882, 3.01367]
$\tilde{x}_2$	[1.9693, 2.0302]	[1.603804, 2.396196]	[1.9693,2.0302]	[1.8555,2.0215]	[1.9765, 2.0214]

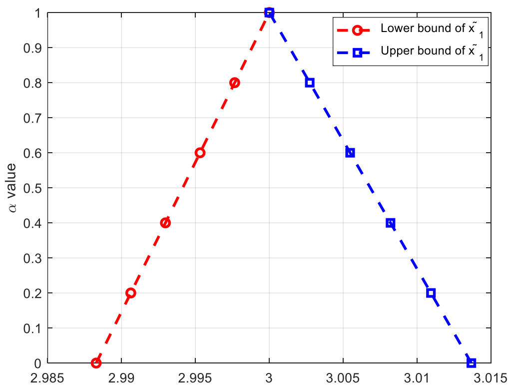

Figure 13. TFN solution of $\tilde{x}_1$ for Example 3.

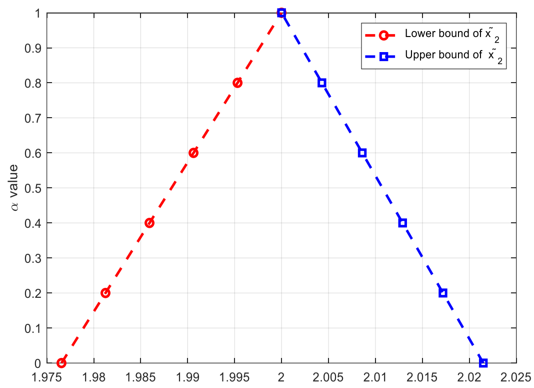

Figure 14. TFN solution of $\tilde{x}_2$ for Example 3.

Finally, the fuzzy IODS algorithm stands out as an effective and useful method for solving a fuzzy system of nonlinear equations. Its computational capabilities make it an effective tool for dealing with real-world problems where uncertainty and ambiguity exist.

Next, considering the width of the solutions with respect to the different combinations of input uncertainties, the sensitiveness of the systems can be analyzed. In this case, the input parameters which give high width solution are more sensitive, whereas the input parameters which give less width solution are less sensitive.

7. Conclusions

In this work, the IODS optimization approach was extended in fuzzy environment to solve the fuzzy system of nonlinear equations. As such, the fuzzy IODS optimization algorithm was developed. To study the efficacy of the algorithm, convergence analysis was performed. The developed algorithm was demonstrated with three example problems of system of nonlinear equations in fuzzy environment. The system of nonlinear equations was divided into three cases, viz., only fuzzy (only coefficient matrix is fuzzy and only right-side vector is fuzzy) and fully fuzzy system of nonlinear equations. Then, by using MATLAB code, the same was solved through the proposed algorithm, and the fuzzy solutions were reported. For each case, both numerical and graphical convergences were shown. Further, iteration-wise solutions were included. From the current work, the following observations are drawn.

1) The proposed algorithm is easy to use, and only seven sub-problems are needed to investigate, irrespective of the size of the system.
2) For a linear system of equations, the proposed method guarantees an optimal solution.
3) Fuzzy IODS is a derivative free method which reduces the computational complexities and time cost.
4) The fuzzy IODS as compared to many other methods has effectively handled an uncertain system, and it provides an efficient solution.
5) Using the proposed fuzzy IODS method, we can solve both only and fully fuzzy systems of equations, which shall be beneficial to handle real-life engineering and science problems.

There are some shortcomings of the present algorithm: As the number of variables and the complexity of the fuzzy system increase, it may be challenging to handle the algorithm in terms of extensible and computational efficiency. Furthermore, for highly nonlinear system, the present algorithm prematurely converges.

An overall study of the proposed method suggests the fuzzy IODS algorithm may be extended to handle more complex uncertain systems, higher dimensional environment, eigenvalue problems, systems with constraints, etc. by updating the search approach. Finally, it can be applicable to quantify the uncertainties in many science and engineering problems, such as risk assessment problems, structural engineering problems, fluid mechanics, etc.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

Authors would like to thank VIT-AP University, Amaravati for providing financial support to this research work.

Conflict of interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

1. D. Dubois, H. Prade, Operations on fuzzy numbers, *Int. J. Syst. Sci.*, **9** (1978), 613–626. https://doi.org/10.1080/00207727808941724
2. S. Nahmias, Fuzzy variables, *Fuzzy Set. Syst.*, **1** (1978), 97–110. https://doi.org/10.1016/0165-0114(78)90011-8
3. L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, *Inform. Sciences*, **8** (1975), 199–249. https://doi.org/10.1016/0020-0255(75)90036-5
4. R. Jafari, W. Yu, Fuzzy modeling for uncertainty nonlinear systems with fuzzy equations, *Math. Probl. Eng.*, **2017** (2017). https://doi.org/10.1155/2017/8594738
5. J. J. Buckley, Y. Qu, Solving linear and quadratic fuzzy equations, *Fuzzy Set. Syst.*, **38** (1990), 43–59. https://doi.org/10.1016/0165-0114(90)90099-R
6. J. J. Buckleym Y. Qu, Solving fuzzy equations: A new solution concept, *Fuzzy Set. Syst.*, **39** (1991), 291–301. https://doi.org/10.1016/0165-0114(91)90099-C
7. S. Abbasbandy, B. Asady, Newton's method for solving fuzzy nonlinear equations, *Appl. Math. Comput.*, **159** (2004), 349–356. https://doi.org/10.1016/j.amc.2003.10.048
8. S. Abbasbandy, R. Ezzati, Newton's method for solving a system of fuzzy nonlinear equations, *Appl. Math. Comput.*, **175** (2006), 1189–1199. https://doi.org/10.1016/j.amc.2005.08.021
9. J. Shokri, On systems of fuzzy nonlinear equations, *Appl. Math. Sci.*, **2** (2008), 1205–1217.
10. R. Jafari, S. Razvarz, A. Gegov, *A new computational method for solving fully fuzzy nonlinear systems*, In: Computational Collective Intelligence, Cham, 2018. https://doi.org/10.1007/978-3-319-98443-8_46
11. M. T. Kajani, B. Asady, A. H. Vencheh, An iterative method for solving dual fuzzy nonlinear equations, *Appl. Math. Comput.*, **167** (2005), 316–323. https://doi.org/10.1016/j.amc.2004.06.113
12. A. N. Gani, S. N. M. Assarudeen, A new operation on triangular fuzzy number for solving fuzzy linear programming problem, *Appl. Math. Sci.*, **6** (2012), 525–532. https://doi.org/10.13140/2.1.3405.8881
13. C. E. Sherbrooke, M. N. Patrikalakis, Computation of the solutions of nonlinear polynomial systems, *Comput. Aided Geom. D.*, **10** (1993), 379–405. https://doi.org/10.1016/0167-8396(93)90019-Y
14. J. Garlof, A. Smith, *Solution of systems of polynomial equations by using bernstein expansion*, In: Symbolic Algebraic Methods and Verification Methods, Vienna, Springer, 2001. https://doi.org/10.1007/978-3-7091-6280-4_9
15. M. Yazdi, E. Zarei, S. Adumene, R. Abbassi, P. Rahnamayiezekavat, *Uncertainty modeling in risk assessment of digitalized process systems*, In: Methods in Chemical Process Safety, Elsevier, 2022, 389–416. https://doi.org/10.1016/bs.mcps.2022.04.005
16. M. Yazdi, *Linguistic methods underfuzzy information in system safety and reliability analysis*, Switzerland: Springer Cham, 2022. https://doi.org/10.1007/978-3-030-93352-4
17 K. Deb, *Optimization for engineering design algorithms and examples*, New Delhi: Prentice Hall of India, 2004.
18. S. Nayak, *Fundamentals of optimization techniques with algorithms*, Academic Press, 2020. https://doi.org/10.1016/C2019-1-02539-9

19. Y. Luo, G. Tang, L. Zhou, Hybrid approach for solving systems of nonlinear equations using chaos optimization and quasi-Newton method, *Appl. Soft Comput.*, **8** (2008), 1068–1073. https://doi.org/10.1016/j.asoc.2007.05.013
20. I. L. Ansorena, Work planning optimisation in ports: a simplex application, *Int. J. Math. Opera. Res.*, **14** (2019), 146–155. https://doi.org/10.1504/IJMOR.2019.096985
21. A. Jafarian, R. Jafari, A new computational method for solving fully fuzzy nonlinear matrix equations, *Inter. J. Fuzzy Comput. Model.*, **2** (2019), 275–285. https://doi.org/10.1504/IJFCM.2019.100317
22. M. O. Bibi, N. Ikheneche, M. Bentobache, A hybrid direction algorithm for solving a convex quadratic problem, *Inter. J. Math. Opera. Res.*, **16** (2020), 159–178. https://doi.org/10.1504/IJMOR.2020.105862
23. S. Nayak, S. Chakraverty, A new approach to solve fuzzy system of linear equations, *J. Math. Comput. Sci.*, **7** (2013), 205–212. http://dx.doi.org/10.22436/jmcs.07.03.06
24. S. Nayak, J. Pooja, Numerical optimisation technique to solve imprecisely defined nonlinear system of equations with bounded parameters, *Inter. J. Math. Opera. Res.*, **23** (2021), 394–411. https://doi.org/10.1504/IJMOR.2022.127381
25. H. Li, M. Yazdi, *Stochastic game theory approach to solve system safety and reliability decision-making problem under uncertainty*, In: Advanced Decision-Making Methods and Applications in System Safety and Reliability Problems, Switzerland AG, Springer, Cham, 2022. https://doi.org/10.1007/978-3-031-07430-1_8
26. P. D. Hough, T. G. Kolda, V. J. Torczon, Asynchronous parallel pattern search for nonlinear optimization, *SIAM J. Sci. Comput.*, **23** (2001), 134–156. https://doi.org/10.1137/S1064827599365823
27. R. M. Lewis, V. Torczon, *Rank ordering and positive bases in pattern search algorithms*, Institute for Computer Application in Science and Engineering, 1996.
28. V. Torczon, On the convergence of the multidirectional search algorithm, *SIAM J. Optimiz.*, **1** (1991), 123–145. https://doi.org/10.1137/0801010
29. R. M. Lewis, V. Torczon, A globally convergent sugmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds, *SIAM J. Optimiz.*, **12** (2002), 1075–1089. https://doi.org/10.1137/S1052623498339727
30. G. S. Kirgat, A. N. Surde, Review of Hooke and Jeeves direct search solution method analysis applicable to mechanical design engineering, *Inter. J. Innov. Eng. Res. Technol.*, **1** (2014). Available from: https://repo.ijiert.org/index.php/ijiert/article/view/242
31. H. J. Zimmermann, *Fuzzy set theory—and its application*s, New York: Springer Dordrecht, 2001. https://doi.org/10.1007/978-94-010-0646-0
32. S. Nayak, Uncertain quantification of field variables involved in transient convection diffusion problems for imprecisely defined parameters, *Int. Commun. Heat Mass*, **119** (2020), 104894. https://doi.org/10.1016/j.icheatmasstransfer.2020.104894
33. D. Chakraborty, D. Guha, Addition of two generalized fuzzy numbers, *Int. J. Indust. Math.*, **2** (2010), 9–20.
34. E. D. Dolan, R. M. Lewis, V. Torczon, On the local convergence of pattern search, *SIAM J. Optimiz.*, **14** (2003), 567–583. https://doi.org/10.1137/S1052623400374495
35. R. M. Lewis, V. Torczon, M. W Trosset, *Why pattern search works*, Institute for Computer Application in Science and Engineering, 1998.
36. H. L. Royden, P. Fitzpatrick, *Real analysis*, New York: Macmillan, 1988.

37. A. Larry, Minimization of functions having Lipschitz continuous first partial derivatives, *Pac. J. Math.*, **16** (1996), 1–3. https://doi.org/10.2140/pjm.1966.16.1
38. V. Torczon, On the convergence of pattern search algorithms, *SIAM J. Optimiz.*, **7** (1997), 1–25. https://doi.org/10.1137/S1052623493250780
39. R. M. Lewis, V. Torczon, M. W. Trossetca, Direct search methods: Then and now, *J. Comput. Appl. Math.*, **124** (2000), 191–207. https://doi.org/10.1016/S0377-0427(00)00423-4
40. S. Nayak, S. Chakraverty, *Interval finite element method with MATLAB*, San Diego: Academic Press, 2018. https://doi.org/10.1016/C2016-0-04446-7
41. S. Nayak, S. Chakraverty, *Numerical solution of interval nonlinear system of equations*, In: 2015 International Conference on Computational Intelligence and Networks, 2015. https://doi.org/10.1109/CINE.2015.43