Optimality

www.opt.reapress.com

Opt. Vol. 2, No. 2 (2025) 106–117.

Paper Type: Original Article

An Inexact Line Search FQNDFP Algorithm to Investigate Uncertain Optimization Problems

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

[1]Department of Mathematics Vignan's Institute of Information Technology (A),Duvvada, Vishakhapatnam, AP 530049, India; jrpareshkp@gmail.com.

[2]Department of Mathematics, School of Advanced Sciences, VIT-AP University, Amaravati, AP, India; sukant-gacr@gmail.com.

Citation:

Received: 03 September 2024 *Revised: 20 November 2024* *Accepted: 02 March 2025*	Panigrahi1, P. K., & Nayak, S. (2025). An Inexact line search FQNDFP algorithm to investigate uncertain optimization problems. *Optimality*, 2(2), 106-117.

Abstract

This paper introduces a Fuzzy Quasi Newton Davidon Fletcher Powell (FQNDFP) optimization algorithm incorporating with Armijo line search technique to effectively handle imprecisely defined optimization problems. Unlike traditional probabilistic methods, this approach leverages fuzzy set theory to model uncertainties in optimization variables given in the objective function. The proposed algorithm integrates the Davidon-Fletcher-Powell (DFP) update formula, ensuring computational efficiency and rapid convergence by approximating the inverse Hessian matrix. The Armijo line search guarantees sufficient descent while adapting the step size dynamically. This combination enhances the algorithm's ability to navigate complex, nonlinear, and uncertain objective landscapes effectively. The performance of the algorithm is evaluated on benchmark problems and fuzzy objective functions, demonstrating accuracy, robustness, and convergence compared to existing methods.

Keywords: Fuzzy number, Optimization problem, IFQNDFP optimization technique, Inexact line search, Armijo line search.

1|Introduction

Unconstrained optimization problems are prevalent across various fields, including engineering, economics, and machine learning. Many developments of efficient optimization algorithms have emerged for this type of problem. Among these, the quasi-Newton methods, particularly those employing the Davidon-Fletcher-Powell (DFP) update have shown promising results. In this context, several classical optimization methods are available to solve the nonlinear system and unconstrained optimization problems. In gradient-based approaches,

Corresponding Author: jrpareshkp@gmail.com

https://doi.org/10.22105/opt.v2i2.81

such as the Steepest Descent method, Newton's method, Quasi-Newton method, Broyden's method, and the Levenberg–Marquardt method, can be applied to nonlinear systems as described by [1]. For unconstrained optimization problems, these methods are discussed in [2]; [3]; and [4]. But in real practice, the system possesses uncertainties due to most of the parameters are imprecise in nature. In this regards, [5] introduced the fuzzy set theory, which needs to incorporate for handling such impreciseness. To address the inherent imprecision in objective functions and constraint within decision-making problems, the fuzzy optimization problems have been extensively studied by [6]. In a survey article, [7] highlighted the key developments and future directions in the theories and applications of fuzzy optimization. In [8], authors provided an insightful overview of the evolution of fuzzy optimization problems. In light of this, [9], along with [10] developed a Newton method for solving unconstrained fuzzy optimization problems. Likewise, in [11] introduced both a Newton method and a quasi-Newton method [12] for interval optimization problems. In real-life optimization scenarios, it is often necessary to optimize a fuzzy function over a real data. This literature review explores the integration of fuzzy computation into quasi-Newton methods, specifically focusing on the DFP update to enhance optimization performance. Quasi-Newton methods are iterative optimization techniques that used to find the approximation of Hessian matrix to determine the optimal solution. Among these, the DFP method is notable for its simplicity and efficiency. In [1], authors provide a comprehensive overview of numerical optimization techniques, highlighting the importance of quasi-Newton methods in solving unconstrained problems. Despite their effectiveness, traditional quasi-Newton methods can struggle with non-convex optimization problems. Further, the DFP update mechanism is designed to iteratively refine the approximation of the Hessian matrix, which is crucial for convergence study of optimal solutions efficiently. Recent advancements in stochastic quasi-Newton methods have broadened the applicability of DFP updates to large-scale optimization scenarios [13]. However, integrating fuzzy concepts into this process could provide additional flexibility and robustness, particularly in environments characterized by uncertainty.

Recent literature has introduced various competitive swarm optimizers and bio-inspired algorithms that have been successfully applied to large-scale optimization problems [14]. These approaches often leverage elements of fuzzy computation to enhance their search capabilities. There remains a significant gap in synthesizing these contemporary methods with traditional quasi-Newton approaches, particularly regarding the DFP update.

As such, the fuzzy computation [17] has been increasingly utilized in optimization problems due to its ability to handle uncertainty and imprecision. The incorporation of fuzzy theory [2] into optimization frameworks can enhance the robustness of solutions, especially in scenarios where the objective function or constraints are not clearly defined. Previous studies have demonstrated that fuzzy logic can effectively improve the performance of various optimization algorithms [15]. A fuzzy set is the union of intervals with different α-cuts. In [16] authors provided an optimization approach to solve an interval nonlinear system with a limited parameter. Recently, the authors [17] developed a derivative free approach to solve the fuzzy nonlinear system. In [18], authors provided an IODS optimization algorithm to handle the fuzzy unconstrained optimization problem. Consequently, various direct search methods have been employed to solve nonlinear systems of equations, enabling the analysis of fuzzy nonlinear systems. The descent direction method plays a major role to develop the optimization algorithm. Therefore, in [19] proposed a FCGDO method for unconstrained optimization problem. However, the application of fuzzy principles within the quasi-Newton framework remains under explored. Despite the substantial body of work surrounding quasi-Newton methods and fuzzy optimization, several knowledge gaps persist:

(1) There is limited research on the direct application of fuzzy theory within the DFP update mechanism, which could potentially enhance convergence rates and solution quality.

(2) Comparative studies that evaluate the performance of fuzzy quasi-Newton methods with DFP updates against other optimization techniques are sparse. Such analyses are essential to establish the efficacy of this hybrid approach.

(3) The scalability of fuzzy quasi-Newton methods in large-scale optimization problems remains largely unexplored. Future research should address how these methods can be adapted for high-dimensional large scale problems.

The integration of fuzzy computation into the quasi-Newton framework, specifically through the DFP update, presents a promising avenue for enhancing the performance of unconstrained optimization algorithms. Thus, in this article, main intend is to develop an inexact quasi-Newton method as an alternative of the traditional quasi Newton method, primarily to reduce the computational overhead associated with computing the inverse of the Hessian matrix. To handle such issue in fuzzy environment While significant advancements have been made in both fuzzy computation and quasi-Newton methods. Additional research is required to fully exploit the synergies between these fields. Addressing the identified knowledge gaps and pursuing the suggested research directions will contribute to the development of more robust and efficient optimization techniques suitable for complex real-world applications.

2|Fuzzy Optimization Problem Formulation

The fuzzy system of equations can be constructed as

$$\begin{aligned}\mathcal{F}_1(x_1,x_2,\ldots,x_n)&=\xi_1\\ \mathcal{F}_2(x_1,x_2,\ldots,x_n)&=\xi_2\\ &\vdots\\ \mathcal{F}_n(x_1,x_2,\ldots,x_n)&=\xi_n\end{aligned} \tag{1}$$

Thus, Eq. (1) can be converted into an interval form using a parametric representation.

$$\begin{aligned}\left[\mathcal{F}^{\alpha}_{L_1}(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_n),\mathcal{F}^{\alpha}_{R_1}(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_n)\right]&=[\xi_{L_1},\xi_{R_1}]\\ \left[\mathcal{F}^{\alpha}_{L_1}(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_n),\mathcal{F}^{\alpha}_{R_2}(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_n)\right]&=[\xi_{L_2},\xi_{R_2}]\\ &\vdots\\ \left[\mathcal{F}^{\alpha}_{L_n}(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_n),\mathcal{F}^{\alpha}_{R_n}(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_n)\right]&=[\xi_{L_n},\xi_{R_n}]\end{aligned} \tag{2}$$

Fuzzy system in Eq. (2) is transformed into an interval system by using the different values of α.

$$\begin{aligned}\mathcal{F}^{\alpha}_1(\tilde{x})&=\left(\left[\mathcal{F}^{a}_{L_1}(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_n),\mathcal{F}^{\alpha}_{R_1}(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_n)\right]-[\xi_{L_1},\xi_{R_1}]\right)\\ \mathcal{F}^{\alpha}_2(\tilde{x})&=\left(\left[\mathcal{F}^{a}_{L_2}(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_n),\mathcal{F}^{\alpha}_{R_2}(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_n)\right]-[\xi_{L_2},\xi_{R_2}]\right)\\ &\vdots\\ \mathcal{F}^{\alpha}_n(\tilde{x})&=\left(\left[\mathcal{F}^{\alpha}_{L_n}(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_n),\mathcal{F}^{a}_{R_n}(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_n)\right]-[\xi_{L_n},\xi_{R_n}]\right)\end{aligned} \tag{3}$$

The above system Eq. (3) can be converted into nonlinear unconstrained optimization problem which is written as

$$\tilde{\mathcal{F}}(\tilde{x})=\left(\mathcal{F}^{a}_1(\tilde{x})\right)^2+\left(\mathcal{F}^{\alpha}_2(\tilde{x})\right)^2+\cdots+\left(\mathcal{F}^{\alpha}_2(\tilde{x})\right)^2 \tag{4}$$

A multi-variable function of $\mathcal{F}:\mathbb{R}^n\to\mathbb{R}$) possess a minimum that can be obtained with the help of IFQN-DFP optimization algorithm. Then, the unconstrained optimization problem is defined as

$$\min_{x^n}\mathcal{F}(\tilde{x}),\text{ where }\tilde{x}=[x_1,x_2,\ldots,x_n]^{]^n}\in R^n \tag{5}$$

In the next section, we will discuss about the Inexact IFQN-DFP optimization algorithm in details.

3|METHODOLOGY

In this section, we develop inexact fuzzy quasi-newton-Davidon-Fletcher-Powell (IFQN-DFP) optimization algorithm to solve fuzzy unconstrained multivariable optimization problem. The same is derived from the fuzzy nonlinear problem. The derivation IFQN-DFP is performed by using the combination of Daviden-Fletcher-Powell and Armijo type inexact line search. The exact line rule ideal line search rules, but due to some difficult, it is impossible to implement in real life problem. As such, several researchers have studied various type of line search technique to solve the optimization problem. Moreover, This algorithm is proposed by using modified Armijo type line search to minimize the field variables of fuzzy unconstrained optimization problem. In the next we discuss the procedure to develop IFQN-DFP. Let us consider the fuzzy unconstrained optimization problem

$$\min_{\mathbb{R}^n}\mathcal{F}(\tilde{x}),\tilde{x}=[x_1,x_2,\ldots,x_n]^T\in\mathbb{R}^n \tag{6}$$

where, $\tilde{\mathcal{F}}(\tilde{x}) : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable convex function. For instance, quasi-Newton methods to adjust the objective function using an approximation of the inverse Hessian based on a series of rank updates, which makes the computations more affordable. So, we execute IFQN-DFP with a starting point $x_0 \in \mathbb{R}^n$. Here, the IFQN-DFP algorithm can be obtained from the following expression

To find the solution of this problem, very known Newton method is existed, the issue with the Newton method for optimization is that it requires computing the inverse of the Hessian matrix, which is the second-order derivative of the objective function. This operation may be very expensive computationally, especially for large-scale problems, because computing and storing the inverse of the Hessian is prohibitively expensive. There have been several methods developed to circumvent this problem. One common approach is to use approximations of the Hessian instead of computing its inverse.

IFQN-DFP Algorithm

Let us consider quadratic approximation for left and right fuzzy value function $\tilde{\mathcal{F}}_L^\alpha$ and $\tilde{\mathcal{F}}_R^\alpha$ at $\tilde{x}_k$. We may calculate $\tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k), \tilde{\mathcal{F}}_R^\alpha(\tilde{x}_k), \nabla\tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k), \nabla\tilde{\mathcal{F}}_R^\alpha(\tilde{x}_k)$ and $\nabla^2\tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k), \nabla^2\tilde{\mathcal{F}}_R^\alpha(\tilde{x}_k)$ for all $\alpha \in [0,1]$ and $k = 1, 2, 3, \ldots$ Hence, we can have a Taylor's quadratic approximation of $\tilde{\mathcal{F}}_L^\alpha$ and $\tilde{\mathcal{F}}_R^\alpha$ at $\tilde{x}_k$ yields a function $\mathbb{F}_L^\alpha(\tilde{x})$ and $\mathbb{F}_R^\alpha(\tilde{x})$ as follows

$$\begin{aligned}\mathbb{F}_L^\alpha(\tilde{x}) &= \tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k) + \nabla\tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k)^T(x - \tilde{x}_k)\\ &+\frac{1}{2}(x - \tilde{x}_k)^T\nabla^2\tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k)(x - \tilde{x}_k)\end{aligned} \tag{7}$$

and

$$\begin{aligned}\mathbb{F}_R^\alpha(\tilde{x}) &= \tilde{\mathcal{F}}_R^\alpha(\tilde{x}_k) + \nabla\tilde{\mathcal{F}}_R^\alpha(\tilde{x}_k)^T(x - \tilde{x}_k)\\ &+\frac{1}{2}(x - \tilde{x}_k)^T\nabla^2\tilde{\mathcal{F}}_R^\alpha(\tilde{x}_k)(x - \tilde{x}_k)\end{aligned} \tag{8}$$

By setting the partial derivatives of Eqs. (7) and (8) equal to zero for the minimum of $\mathbb{F}_L^\alpha(\tilde{x})$. we obtain, the derivative of $\mathbb{F}_L^\alpha(\tilde{x})$ and $\mathbb{F}_R^\alpha(\tilde{x})$, we get

$$\begin{aligned}\nabla\mathbb{F}_L^\alpha(\tilde{x}) &= \nabla\tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k) + \nabla^2\tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k)(\tilde{x} - \tilde{x}_k)\\ \nabla\mathbb{F}_R^\alpha(\tilde{x}) &= \nabla\tilde{\mathcal{F}}_R^\alpha(\tilde{x}_k) + \nabla^2\tilde{\mathcal{F}}_R^\alpha(\tilde{x}_k)(\tilde{x} - \tilde{x}_k)\end{aligned} \tag{9}$$

The Newton method is attempted to find the solution of the $\tilde{x}_{k+1}$ in term of $\tilde{x}_k$. The iterative updated field variables can be obtained from the following expression.

$$\begin{cases}\tilde{x}_{L_{k^2}}^\alpha = \tilde{x}_{L_k}^\alpha - \left[\nabla^2\tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k)\right]^{-1}\nabla\tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k)\\ \tilde{x}_{R_{k+1}}^\alpha = \tilde{x}_{R_k}^\alpha - \left[\nabla^2\tilde{\mathcal{F}}_R^\alpha(\tilde{x}_k)\right]^{-1}\nabla\tilde{\mathcal{F}}_R^\alpha(\tilde{x}_k)\end{cases} \tag{10}$$

where $\tilde{x}_{L_{k+1}}$ and $\tilde{x}_{R_{k+1}}$ is the new iteration point for every left and right value, and $\tilde{x}_{L_k}$ and $\tilde{x}_{R_k}$ is the previous point. However, due to the computational difficulty to find $\nabla^2\tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k)$ and $\nabla^2\tilde{\mathcal{F}}_R^\alpha(\tilde{x}_k)$ with different α value, it is often to suggested to consider an appropriate approximation $H_{L_k} = \left[\nabla^2\tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k)\right]^{-1}$ and $H_{R_k} = \left[\nabla^2\tilde{\mathcal{F}}_L^a(\tilde{x}_k)\right]^{-1}$. Note that the $[H_{L_k}, H_{R_k}]$ is composed of the second derivation of nonlinear fuzzy objective function $\left[\tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k), \tilde{\mathcal{F}}_R^\alpha(\tilde{x}_k)\right]$. Hence, the basic idea behind quasi-newton method is to update $[H_{L_k}, H_{R_k}]$. In this regard, the update of field variables can be obtained from quasi-Newton is

$$\begin{cases}\tilde{x}_{L_{k+1}}^\alpha = \tilde{x}_{L_k}^\alpha - \lambda_k H_L^\alpha\nabla\tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k)\\ \tilde{x}_{R_{k+1}}^\alpha = \tilde{x}_{R_k}^\alpha - \lambda_k H_{R_k}^\alpha\nabla\tilde{\mathcal{F}}_R^\alpha(\tilde{x}_k)\end{cases} \tag{11}$$

where, λ_k is the step size which is consider as Armijo type line search along the direction of $\left[S_{L_{k'}}^\alpha, S_{R_k}^\alpha\right]$ is search direction. Here, $S_k^{\alpha,\gamma}$ is represented in terms $\left[S_{L_k}^\alpha, S_{R_k}^\alpha\right]$ of parametric from is define follows,

$$S_k^{a,\gamma} = (1-\gamma)S_{L_k}^\alpha + \gamma S_{R_k}^\alpha, \text{ for all } \gamma \in [0,1] \tag{12}$$

where

$$\left[S_{L_k}^\alpha = -H_{L_k}^\alpha\nabla\tilde{\mathcal{F}}_L^\alpha(\tilde{x}_k), S_{R_k}^\alpha = -H_{R_k}^\alpha\nabla\tilde{\mathcal{F}}_R^\alpha(\tilde{x}_k)\right]$$

At $S_k^{\alpha,\gamma}$ occurs at $\gamma=0$, in which case $S_k^{\alpha\gamma}=-H_{L_k}\nabla\tilde{F}_L^{\alpha}\left(\tilde{x}_k\right)$ and $H_k^{\alpha\gamma}=(1-\gamma)H_{L_k}^{\alpha}+\gamma H_{R_k}^{\alpha}$ at $\gamma=1$, in which case $S_k^{a\gamma\gamma}=-H_{R_k}\nabla\tilde{\mathcal{F}}_R^{\alpha}\left(\tilde{x}_k\right)$.

Computation of $[H_{L_k},H_{R_k}]$ using DPF Algorithm

To implement the Eq. (11) we need to find the approximate value of inverse of the Hessian matrix in the interval term $[H_{L_k},H_{R_k}]$. As per the Eq. (9) we can expand the function at $\tilde{x}_0$, and using the Taylor's series is written as

$$\begin{aligned}\nabla\mathbb{F}_L^{\alpha}(\hat{x})=\nabla\tilde{\mathcal{F}}_L^{\alpha}\left(\tilde{x}_0\right)+\nabla^2\tilde{F}_L^{\alpha}\left(\tilde{x}_0\right)\left(\tilde{x}-\tilde{x}_0\right)\\ \nabla\mathbb{F}_R^{\alpha}(\tilde{x})=\nabla\tilde{\mathcal{F}}_R^{\alpha}\left(\tilde{x}_0\right)+\nabla^2\tilde{\mathcal{F}}_R^{\alpha}\left(\tilde{x}_0\right)\left(\tilde{x}-\tilde{x}_0\right)\end{aligned} \tag{13}$$

If the Eq. (13) is working on two points at $\tilde{x}_k$ and $\tilde{x}_{k+1}$, which can rewrite as

$$\left\{\begin{array}{l}\nabla\mathbb{F}_{L_{k+1}}^{\alpha}=\nabla\tilde{\mathcal{F}}_{\tilde{\alpha}}^{\alpha}\left(\tilde{x}_0\right)+\nabla^2\tilde{\mathcal{F}}_L^{\alpha}\left(\tilde{x}_0\right)\left(\tilde{x}_{k+1}-\tilde{x}_0\right)\\ \nabla\mathbb{F}_{R_{k+1}}^{\alpha}=\nabla\tilde{\mathcal{F}}_R^{\alpha}\left(\tilde{x}_0\right)+\nabla^2\tilde{\mathcal{F}}_R^{\alpha}\left(\tilde{x}_0\right)\left(\tilde{x}_{k+1}-\tilde{x}_0\right)\end{array}\right. \tag{14}$$

$$\left\{\begin{array}{l}\nabla\mathbb{F}_{L_k}^{\alpha}=\nabla\tilde{\mathcal{F}}_L^{\alpha}\left(\tilde{x}_0\right)+\nabla^2\tilde{\mathcal{F}}_{\alpha}^{\alpha}\left(\tilde{x}_0\right)\left(\tilde{x}_k-\tilde{x}_0\right)\\ \nabla\mathbb{F}_{R_k}^{\alpha}=\nabla\tilde{\mathcal{F}}_R^{\alpha}\left(\tilde{x}_0\right)+\nabla^2\tilde{\mathcal{F}}_R^{\alpha}\left(\tilde{x}_0\right)\left(\tilde{x}_k-\tilde{x}_0\right)\end{array}\right\} \tag{15}$$

Subtracting Eq. (15) from Eq. (14) yields

$$\begin{aligned}\left[\nabla^2\tilde{\mathcal{F}}_L^{\alpha}\left(\tilde{x}_k\right)\right]\tilde{d}_k=\tilde{g}_{R_k}^{\alpha}\\ \left[\nabla^2\tilde{\mathcal{F}}_R^{\alpha}\left(\tilde{x}_k\right)\right]\tilde{d}_k=\tilde{g}_{R_k}^{\alpha}\end{aligned} \tag{16}$$

where, $d_k=\tilde{x}_{k+1}-\tilde{x}_k$ and $\tilde{g}_{L_k}=\nabla\mathbb{F}_{L_{k+1}}^{a}-\nabla\mathbb{F}_{L_k}^{a}$ and $\tilde{g}_{R_k}=\nabla\mathbb{F}_{R_{k+1}}^{a}-\nabla\mathbb{F}_{R_k}^{a}$ and the $\left[\tilde{g}_{L_k}^{a},\tilde{g}_{R_k}^{\alpha}\right]$ can be written as

$$\begin{aligned}\tilde{g}_k^{\alpha,\gamma}&=\left[\tilde{g}_{L_k}^{\alpha},\tilde{g}_{R_k}^{\alpha}\right]=(1-\gamma)\tilde{g}_{L_k}^{\alpha}+\gamma\tilde{g}_{R_k}^{\alpha}\\ &=(1-\gamma)\left[\nabla^2\tilde{F}_L^{\alpha}\left(\tilde{x}_k\right)\right]d_k+\gamma\left[\nabla^2\tilde{\mathcal{F}}_R^{\alpha}\left(\tilde{x}_k\right)\right]d_k,\\ &\text{for all }\gamma\in[0,1]\end{aligned}$$

At $\tilde{g}_k^{\alpha,\gamma}$ occurs at $\gamma=0$, in which case $\tilde{g}_k^{\alpha,\gamma}=\left[\nabla^2\tilde{\mathcal{F}}_L^{\alpha}\left(\tilde{x}_k\right)\right]\tilde{d}_k$ and at $\gamma=1$, in which case $S_k^{\alpha,\gamma}=\left[\nabla^2\tilde{\mathcal{F}}_R^{\alpha}\left(\tilde{x}_k\right)\right]\tilde{d}_k$. Similarly, we can saw that $H_k^{\alpha\gamma}=(1-\gamma)H_{L_k}^{\alpha}+\gamma H_{R_k}^{\alpha}$, for all $\gamma\in[0,1]$.
We may say

$$\tilde{d}_k=[H_k^{\alpha,\gamma}]\,\tilde{g}_k^{\alpha,\gamma} \tag{17}$$

The general formula for updating the $[H_k^{\alpha\gamma}]$ can be written as

$$\left[H_{k+1}^{a\gamma}\right]=[H_k^{a\gamma}]+[\Delta H_k^{a\gamma}] \tag{18}$$

where, $[\Delta H_k^{a\gamma}]$is considered as updated rank in n. Here, we tried to introduce rank update $\left[H_{k+1}^{\alpha\gamma}\right]$ is presented as follow To derive the rank update, we simple choose u_k and v_k are two vectors, $u_k,v_k\in\mathbb{R}^n$ and r_k and s_k are the two scalars. Now, we consider as

$$[\Delta H_k^{\alpha\gamma}]=r_ku_ku_k^T+s_kv_kv_k^T \tag{19}$$

Therefore, now we have

$$\left[H_{k+1}^{a\gamma}\right]=[H_k^{\alpha\gamma}]+r_ku_ku_k^T+s_kv_kv_k^T \tag{20}$$

As per the Eq. (20) is satisfying the quasi-Newton method and update $\left[H_k^{\alpha-}\right]$ to $\left[H_{k+1}^{a\gamma}\right]$ matrix. Now, we can update the Eq. (17) which is

$$\begin{aligned}\tilde{d}_k&=\left[H_{k+1}^{\alpha\gamma}\right]\tilde{g}_k^{a\gamma}\\ &=\left([H_k^{\alpha\gamma}]+r_ku_ku_k^{\tau}+s_kv_kv_k^T\right)\tilde{g}_k^{a\gamma}\\ \tilde{d}_k&=[H_k^{a\gamma}]\,\tilde{g}_k^{\alpha\gamma+}+r_ku_k\left(u_k^{\tau}\tilde{g}_k^{\alpha\gamma}\right)+s_kv_k\left(v_k^T\tilde{g}_k^{\alpha\gamma}\right)\end{aligned} \tag{21}$$

Although, u_k and v_k are not uniquely defined, so we consider as

$$u_k=\tilde{d}_k,v_k=[H_k^{a\gamma}]\,\tilde{g}_k^{a\gamma}$$

and putting this value in Eq. (21), which are define as

$$r_k = \frac{1}{u_k^T \hat{g}_k^{a\gamma}} = \frac{1}{\tilde{d}\tilde{g}_k^{a\gamma}}$$
$$s_k = -\frac{1}{v_k^T \tilde{g}_k^{a\gamma}} = -\frac{1}{\tilde{g}_k^{a\gamma T}\left[H_k^{a\gamma}\right]\tilde{g}_k^{a\gamma}}$$

Thus, the rank update formula can be expressed as

$$\begin{aligned}\left[H_{k+1}^{a\gamma}\right] &= \left[H_k^{a\gamma}\right] + \left[\Delta H_k^{a\gamma}\right] \\ &= \left[H_k^{a\gamma}\right] + \frac{\tilde{d}_k \tilde{d}_k^T}{\tilde{d}_k^T \tilde{g}_k^{\alpha\gamma}} - \frac{\left(\left[H_k^{a\gamma}\right]\tilde{g}_k^{a\gamma}\right)\left(\left[H_k^{a\gamma}\right]\tilde{g}_k^{a\gamma}\right)^T}{\tilde{g}_k^{a\gamma T}\left[H_k^{a\gamma}\right]\tilde{g}_k^{a\gamma}}\end{aligned} \tag{22}$$

We consider this Eq. (22) to update the sequence of inverse Hessian matrix approximation of IFQN-DFP As per the Eq. (11), we may write as

$$\tilde{x}_{k+1}^{a,\gamma} = \tilde{x}_k^{a,\gamma} + \tilde{\lambda}_k S_k^{a,\gamma} \tag{23}$$

where, $s_k^{\alpha,\gamma}$ is the search process and as we know $\tilde{d}_k^{\alpha,\gamma} = \tilde{x}_{k+1}^{\alpha,\gamma} - \tilde{x}_k^{\alpha,\gamma}$ can be rewritten as

$$\tilde{d}_k^{\alpha,\gamma} = \tilde{\lambda}_k S_k^{a,\gamma} \tag{24}$$

After that $S_k^{\alpha,\gamma}$ is given and $\tilde{d}_k^{\alpha,\gamma}$ is given, thus the next task is to find a step size $\tilde{\lambda}_k$ along the search direction. Therefore, we can use some inexact line search rules. The advantage of Armijo's type line search is that it enables to estimate an initial step size. For good estimation for initial steps, it makes to decrease the objective function evaluation at each iteration. For that, here we modified a new Armijo type inexact line search rule in fuzzy environment.

Modified Armijo Line Search Rule

Generally, choosing the parameter (such as $\tilde{\theta}, \tilde{\sigma}, \tilde{\beta}$) is very important for real practice. In this regard, to choose an appropriate parameter that is satisfy the proposed FIQN-DFP algorithm. Here, we developed a modified version of Armijo line search, which is easier to find step size $\tilde{\lambda}_k$. It is obvious that the modified inexact line search rule is well define for fuzzy optimization as well as proposed FIQN-DFP. Then, the modified Armijo type line search is representing as follows Choose $\tilde{\theta} > 0, \tilde{\beta} \in (0,1)$ and $\tilde{\sigma} \in \left(0, \frac{1}{2}\right)$, and the step size $\tilde{\lambda}_k$ to be found the max $\left\{\tilde{\theta}, \tilde{\theta}\tilde{\beta}, \tilde{\theta}\tilde{\beta}^2 \ldots\right\}$ such that

$$\begin{aligned}\tilde{\mathcal{F}}^{\alpha,\gamma}\left(\tilde{x}_k + \tilde{\lambda}\tilde{S}_k^{\alpha,\gamma}\right) - \tilde{\mathcal{F}}^{\alpha\gamma}\left(\tilde{x}_k\right) \leq \tilde{\sigma}\tilde{\lambda}\big[&\nabla\tilde{\mathcal{F}}^{\alpha,\gamma}\left(\tilde{x}_k\right)^T s_k^{\alpha,\gamma} \\ &+\frac{1}{2}\tilde{S}_k^{\alpha,\gamma^T}\left[H_k^{\alpha\gamma}\right]\tilde{S}_k^{\alpha,\gamma}\big]\end{aligned}$$

Also, the Eq. (33) and modified Armijo type line search are satisfied the following Hypothesis conditions. We assume that

Assumption 1: The fuzzy valued function $\tilde{\mathcal{F}}^{\alpha,\gamma}\left(\tilde{x}_k\right)$ has a lower and upper bound on the level set

$$L_0 = \left\{\tilde{x}_k \in \mathbb{R}^\eta \mid \tilde{\mathcal{F}}^{\alpha_k\gamma}\left(\tilde{x}_k\right) \leq \tilde{\mathcal{F}}^{\alpha_\gamma\gamma}\left(\tilde{x}_0\right)\right\} \tag{25}$$

Assumption 2 The gradient of fuzzy value function $\nabla\tilde{\mathcal{F}}^{a_\gamma}\left(\tilde{x}_k\right)$ of $\tilde{\mathcal{F}}^{a,\gamma}\left(\tilde{x}_k\right)$ is Lipschitz continuous in an open convex set $\tilde{B}$ that contains L_0, i.e. there exist a constant $L \geq 0$ such that

$$\begin{aligned}\left\|\nabla\mathcal{F}\alpha_k\left(\tilde{x}_k\right) - \nabla\mathcal{F}\alpha^\alpha\left(\tilde{y}_k\right)\right\| \leq \boldsymbol{L}\left\|\tilde{x}_k - \tilde{y}_k\right\|, \\ \text{for all } \tilde{x}_k \text{ and } \tilde{y}_k \in \boldsymbol{B}\end{aligned} \tag{26}$$

For the fuzzy computational aspect, modified Armijo's line search technique is employed to in the proposed Inexact FQNDFP to efficiently determine the step size in the search direction, helping to converge toward the optimal solution during the iterative optimization process. Next, we describe the proposed Inexact FQNDFP algorithm.The proposed algorithm is used to solve the unconstrained optimization problem in a fuzzy environment.

For a better understanding and validity of the proposed method, we have discussed the numerical analysis of the same in the next section.

Algorithm 1 IFQN-DFP Algorithm

Step 1: Formulate the fuzzy unconstrained multi-variable optimization problem $\mathcal{F}^{\alpha,\beta}(\tilde{x}) : R^n \to R$.
Step 2: Choose an initial vector $\tilde{x}_0$
Step 3: Take convergence parameters ϵ and ϵ'.
Step 4: Compute $\nabla\mathcal{F}^{\alpha,\gamma} = \nabla\mathcal{F}^{\alpha,\gamma}(\tilde{x}_k)$, at the point $\tilde{x}_k$, where $k = 1, 2, \ldots, n$
if $\left\|\nabla\mathcal{F}^{\alpha,\beta}(\tilde{x}_k)\right\| < \epsilon$ **then**,
    Terminate.
else if
    thengo to the step 5.
end if
Step 5: Compute the search direction

$$S_k^{\alpha,\gamma} = -H_k^{\alpha}\nabla\mathcal{F}_L^{\alpha}(\tilde{x}_k)$$

Step 6: Set $\tilde{x}_{k+1}$ and update the $\tilde{x}_k$ values with the following relation.

$$\tilde{x}_{k+1}^{\alpha,\gamma} = \tilde{x}_k^{\alpha,\gamma} + \tilde{\lambda}_k S_k^{\alpha,\gamma}$$

where $S_k^{\alpha,\beta} = -H_k^{\alpha}\nabla\mathcal{F}_L^{\alpha}(\tilde{x}_k)$ and $\tilde{\lambda}_k$ is determined by the following step 7
Step 7: Choose $\tilde{\theta} > 0, \tilde{\beta} \in (0,1)$ and $\tilde{\sigma} \in \left(0, \frac{1}{2}\right)$, and the step size $\tilde{\lambda}_k$ to be found the max $\left\{\tilde{\theta}, \tilde{\theta}\tilde{\beta}, \tilde{\theta}\tilde{\beta}^2 \ldots\right\}$ such that

$$\begin{aligned} &\tilde{F}^{\alpha,\gamma}\left(\tilde{x}_k + \tilde{\lambda}S_k^{\alpha,\gamma}\right) - \tilde{F}^{\alpha,\gamma}(\tilde{x}_k) \\ &\leq \tilde{\sigma}\tilde{\lambda}\left[\nabla\tilde{F}^{\alpha,\gamma}(\tilde{x}_k)^T S_k^{\alpha,\gamma} + \frac{1}{2}\tilde{\lambda}S_k^{\alpha,\gamma T}\left[H_k^{\alpha\gamma}\right]S_k^{\alpha,\gamma}\right] \end{aligned}$$

Step 8: Test the point $\tilde{x}_{k+1}^{\alpha,\gamma}$ for optimality.
if $\left\|\nabla\mathcal{F}^{\alpha,\beta}(\tilde{x}_{k+1})\right\| < \epsilon'$ **then**,
    Terminate.
else if
    thengo to the step 9 .
end if
Step 9: Update the Hessian matrix by using DFP as follows

$$\begin{aligned} \left[H_{k+1}^{a\gamma}\right] &= \left[H_k^{a\gamma}\right] + \left[\Delta H_k^{a\gamma}\right] \\ &= \left[H_k^{a\gamma}\right] + \frac{\tilde{d}_k\tilde{d}_k^T}{\tilde{d}_k^T\tilde{g}_k^{\alpha\gamma}} - \frac{\left(\left[H_k^{a\gamma}\right]\tilde{g}_k^{a\gamma}\right)\left(\left[H_k^{a\gamma}\right]\tilde{g}_k^{a\gamma}\right)^T}{\tilde{g}_k^{a\gamma T}\left[H_k^{a\gamma}\right]\tilde{g}_k^{a\gamma}} \end{aligned}$$

Step 10: Set $k = k + 1$ and go to step 4.

4|Numerical Example

Example 1

Here, we have considered an example [18] to apply the IFQN-DFP algorithm. Consider a fuzzy system of nonlinear equations

$$\begin{aligned} \tilde{a}_{11}x_1^2 + \tilde{a}_{12}x_2 &= \tilde{\chi}_1 \\ \tilde{a}_{21}x_1 + \tilde{a}_{22}x_2^2 &= \tilde{\chi}_2 \end{aligned} \tag{27}$$

Here, the coefficients are taken as triangular fuzzy number (TFN)[20]. 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 $\tilde{\chi}_2 = [3.2, 6.7, 8.6]$. Let the initial approximation be $x^{(0)} = (1, 1)^T$ with a tolerance value of $\epsilon = 10^{-4}$. The process begins by formulating the system and transforming Eq.(27) into a fuzzy unconstrained minimization problem, which is solved using the proposed IFQN-DFP algorithm. By applying the IFQN-DFP algorithm, the solution components of x_1 and x_2 with their objective function and corresponding gradient value are represented in Table 1 and 3 for $\alpha = 0$ and Table 2 for $\alpha = 1$. Whereas, the uncertain TFN solution with different α-cut solutions of x_1 and x_2 are represented in Fig. 1 and Fig. 2 respectively. The IFQN-DFP algorithm demonstrates superior performance across various nonlinear benchmark optimization problems. As shown in Table 5 and Fig. 4, IFQN-DFP consistently achieves faster convergence, requiring fewer iterations and function evaluations, with a convergence rate. It significantly reduces computational cost, as indicated by the lower CPU time, making it highly efficient for solving fuzzy optimization problems. The algorithm also produces more accurate solutions, with lower absolute error compared to other methods.

TABLE 1. Iteration wise x_1 and x_2 solution for $\alpha = 0$

Iteration	x_1	x_2	$\mathcal{F}(\tilde{x})$	$\|\nabla\mathcal{F}(\tilde{x})\|$
0	1	1	6.25	7.7897
1	2.3	2.45	3.171	10.1592
2	1.9191	1.3497	0.9417	3.4466
3	1.9897	1.648	0.207	1.4494
4	1.836	1.9768	0.0027	0.2498
5	1.831	1.9531	0.0003	0.0354
6	1.8211	1.9614	0	0.0081
7	1.818	1.9634	0	0.0004
8	1.8179	1.9634	0	0
9	1.8179	1.9634	0	0
10	1.8179	1.9634	0	0

TABLE 2. Iteration wise x_1 and x_2 solution for $\alpha = 0$

Iteration	x_1	x_2	$\mathcal{F}(\tilde{x})$	$\|\nabla\mathcal{F}(\tilde{x})\|$
0	1	1	32.98	33.9988
1	2.4125	2.5875	15.4543	46.6979
2	1.2194	2.927	10.3135	34.4186
3	1.2853	2.1354	3.0193	14.0236
4	1.514	2.2463	0.5995	5.6276
5	1.7815	2.2726	0.0821	2.9604
6	1.7294	2.2458	0.0094	0.6048
7	1.7435	2.2334	0.0017	0.261
8	1.754	2.2241	0	0.0072
9	1.7539	2.224	0	0.0001
10	1.7539	2.224	0	0

Test Function

In this section, we have tested 3 benchmark problem [14] compared with proposed IFQN-DFP algorithm with two different method

TABLE 3. Iteration wise x_1 and x_2 solution for $\alpha = 0$

Iteration	x_1	x_2	$\mathcal{F}(\tilde{x})$	$\|\nabla\mathcal{F}(\tilde{x})\|$
0	1	1	49.41	65.7816
1	1.6469	1.7987	1.0766	14.0369
2	1.9091	1.6819	0.9467	9.6756
3	1.8335	1.8805	0.0727	3.8615
4	1.7901	1.861	0.0021	0.6408
5	1.7962	1.8638	0	0.012
6	1.7964	1.8638	0	0.0007
7	1.7964	1.8638	0	0.0001
8	1.7964	1.8638	0	0
9	1.7964	1.8638	0	0
10	1.7964	1.8638	0	0

TABLE 4. The solution of x_1 and x_2 of proposed algorithm compared with FIODS method

Decision Variable	FIODS Algorithm	Iteration	Present Algorithm	Iteration
x_1	$[1.7539, 1.7965, 1.818]$	14	$[1.7539, 1.7964, 1.8179]$	9
x_2	$[1.8638, 1.9634, 2.224]$	14	$[1.8638, 1.9634, 2.224]$	9

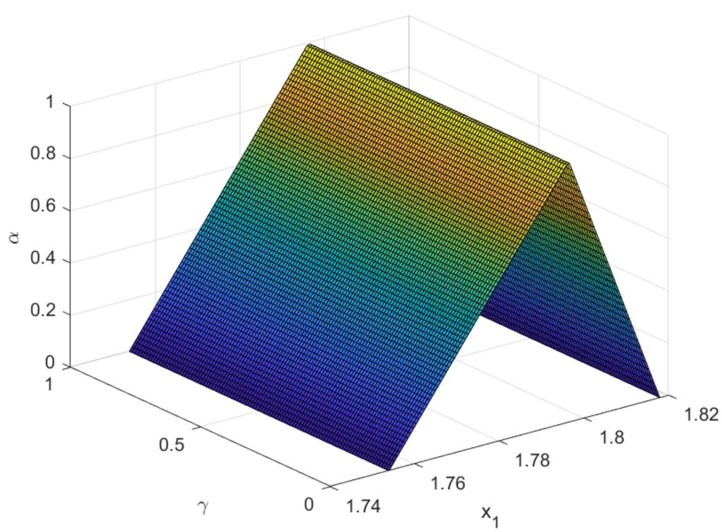

FIGURE 1. Optimal TFN solution of x_1 with membership value

(1) Rosenbrock: A classic optimization benchmark.

$$F = 100 * (x_2 - (x_1)^2)^2 + (1 - x_1))^2 \tag{28}$$

(2) Quadratic Bowl: Simple quadratic for comparison

$$F = (x_1 - 2)^2 + (x_2 - 3)^2 \tag{29}$$

(3) Nonlinear Sinusoidal: A challenging oscillatory function.

$$F = (x_1)^2 + (x_2)^2 - 10 * sin(x_1) - 10 * sin(x_2) \tag{30}$$

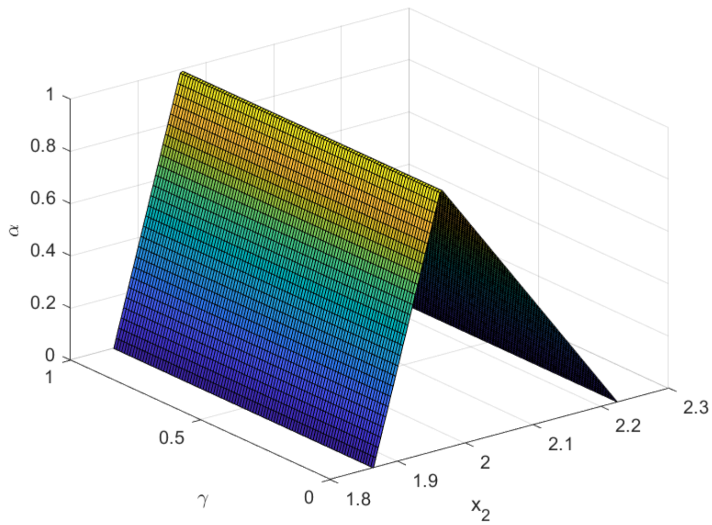

FIGURE 2. Optimal TFN solution of x_2 with membership value

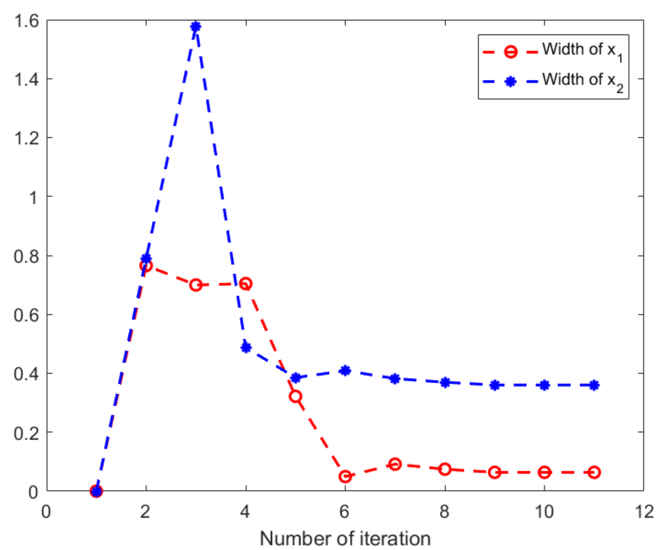

FIGURE 3. Width of TFN solution of x_1 and x_2

TABLE 5. Solution components of IFQN-DFP algorithm with three different function.

S.I	Function Name	Initial value	Optimal solution	Iteration
1	Rosenbrock Function	$[0, 0]$	$[1.0000, 1.0000]$	28
2	Quadratic Bowl	$[0, 0]$	$[2.0000, 3.0000]$	1
3	Nonlinear Sinusoidal	$[0, 0]$	$[1.3064, 1.3064]$	5

The width convergence plot are shown in Fig. 3, further highlights its effectiveness, in Table 1, 2 and 3 showing as rapid decrease in objective function and norm of gradient value within the first few iterations. Further, to the effectiveness of the proposed algorithm, the obtained TFN solutions are compared with the existing FIODS

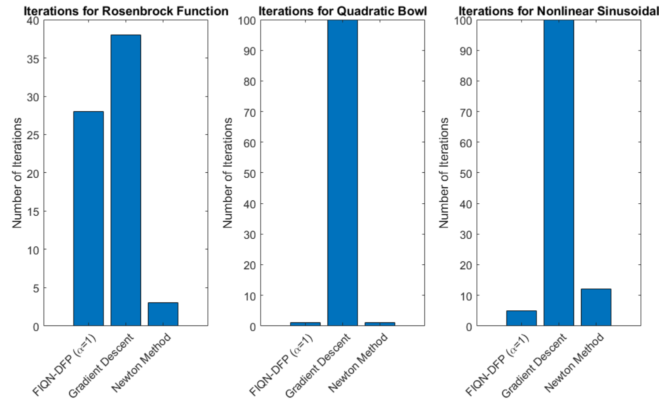

Figure 4. Performance profile different functions

[17] and the same is presented in Table 4 . Moreover, IFQN-DFP maintains robust performance and scalability, demonstrating its applicability to large-scale optimization problems under fuzzy uncertainty environment.

5|Conclusion

This study presents a IFQN-DFP optimization algorithm with Armijo line search to effectively handle imprecisely defined optimization problems. By leveraging fuzzy set theory, the algorithm models uncertainties without relying on probabilistic assumptions, making it well-suited for problems where precise data is unavailable. The integration of the DFP update ensures efficient Hessian approximation, while the Armijo line search adaptively controls step sizes for stable and rapid convergence. Experimental results on benchmark and real-world optimization problems demonstrate the algorithm's robustness, accuracy, and superior performance compared to existing approaches. This work provides a significant contribution to non-probabilistic optimization, offering a reliable method for solving complex, uncertain, and nonlinear optimization problems in engineering, economics, and applied sciences.

References

[1] Nocedal, Jorge, and Stephen J. Wright, eds. Numerical optimization. New York, NY: Springer New York, 1999.
[2] Rao, Singiresu S. Engineering optimization: theory and practice. John Wiley and Sons, 2019.
[3] Deb, Kalyanmoy. Optimization for engineering design: Algorithms and examples. PHI Learning Pvt. Ltd., 2012.
[4] Nayak, Sukanta. Fundamentals of optimization techniques with algorithms. Academic Press, 2020.
[5] Zadeh, Lotfi A. "Fuzzy sets." Information and Control (1965).
[6] Bellman, Richard E., and Lotfi Asker Zadeh. "Decision-making in a fuzzy environment." Management science 17.4 (1970): B-141.
[7] Luhandjula, M. K. "Fuzzy optimization: Milestones and perspectives." Fuzzy Sets and Systems 274 (2015): 4-11.
[8] Lodwick, Weldon A., and Elizabeth Untiedt. "Introduction to fuzzy and possibilistic optimization." Fuzzy Optimization: Recent Advances and Applications. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. 33-62.
[9] Pirzada, U. M., and V. D. Pathak. "Newton method for solving the multi-variable fuzzy optimization problem." Journal of Optimization Theory and Applications 156 (2013): 867-881.
[10] Chalco-Cano, Yurilev, Geraldo Nunes Silva, and Antonio Rufián-Lizana. "On the Newton method for solving fuzzy optimization problems." Fuzzy Sets and Systems 272 (2015): 60-69.
[11] Ghosh, Debdas. "A Newton method for capturing efficient solutions of interval optimization problems." Opsearch 53.3 (2016): 648-665.
[12] Ghosh, Debdas. "A quasi-Newton method with rank-two update to solve interval optimization problems." International Journal of Applied and Computational Mathematics 3.3 (2017): 1719-1738.
[13] Byrd, Richard H., et al. "A stochastic quasi-Newton method for large-scale optimization." SIAM Journal on Optimization 26.2 (2016): 1008-1031.

[14] Cheng, Ran, and Yaochu Jin. "A social learning particle swarm optimization algorithm for scalable optimization." Information Sciences 291 (2015): 43-60.
[15] Castillo, Oscar. Type-2 fuzzy logic in intelligent control applications. Vol. 272. Heidelberg: Springer, 2012.
[16] Nayak, Sukanta, and J. Pooja. "Numerical optimisation technique to solve imprecisely defined nonlinear system of equations with bounded parameters." International Journal of Mathematics in Operational Research 23.3 (2022): 394-411.
[17] Panigrahi, Paresh Kumar, and Sukanta Nayak. "Numerical investigation of non-probabilistic systems using Inner Outer Direct Search optimization technique." AIMS Mathematics 8.9 (2023): 21329-21358.
[18] Panigrahi, Paresh Kumar, and Sukanta Nayak. "Numerical approach to solve imprecisely defined systems using Inner Outer Direct Search optimization technique." Mathematics and Computers in Simulation 215 (2024): 578-606.
[19] Panigrahi, Paresh Kumar, and Sukanta Nayak. "Conjugate gradient with Armijo line search approach to investigate imprecisely defined unconstrained optimisation problem." International journal of computational science and engineering 27.4 (2024): 458-471.
[20] Zimmermann, H-J. "Fuzzy set theory." Wiley interdisciplinary reviews: computational statistics 2.3 (2010): 317-332.