Journal of Uncertain Systems
Vol.11, No.1, pp.3-17, 2017
Online at: www.jus.org.uk

A Quasi-Newton Method for Solving Fuzzy Optimization Problems

M. Ghaznavi*, N. Hoseinpoor
Department of Applied Mathematics, Faculty of Mathematical Sciences
Shahrood University of Technology, Shahrood, Iran

Received 16 April 2016; Revised 7 November 2016

Abstract

In this paper, we develop a quasi-Newton method for unconstrained optimization problems with fuzzy functions. Here, we extend the well-know BFGS method to fuzzy optimization problems. To this end, the generalized Hukuhara differentiability for fuzzy functions is employed. By using a Hessian approximation, we resolve the high computational cost of finding the Hessian in Newton method for fuzzy optimization problems. Utilizing the quasi-Newton algorithm, we find nondominated solutions of a fuzzy optimization problem. Finally, we provide some numerical examples to show the efficiency of the proposed approach over the other methods.

Keywords: fuzzy optimization problem, quasi-Newton method, generalized Hukuhara differentiability, nondominated solution

1 Introduction

In optimizing real world systems, we deal with linear and nonlinear programming problems. In the conventional optimization problems, all of the coefficients are real numbers. However, in the real world, the coefficients involved in the objective and constraint functions are imprecise in nature and have to be stated in fuzzy sense to reflect the real world situation. Fuzzy optimization problems were first considered by Bellman and Zadeh [5]. Thereafter, Tanaka et al. [26] introduced the concept of fuzzy mathematical programming in a general level. Over the last decades, many researchers have studied optimization problems with fuzzy-valued objective functions. We refer to [1, 8, 9, 14, 15, 16, 17, 18, 19, 20] that have been done in this direction.

Pirzada and Pathak [22] proposed the Newton method for unconstrained optimization problems with fuzzy-valued functions. In their proposed method, they utilized Hukuhara differentiability of fuzzy-valued functions and max-ordering relation defined on the set of fuzzy numbers. Afterwards, Chalco-Cano et al. [6] addressed some of the difficulties of the method proposed by Pirzada and Pathak [22] and by using generalized Hukuhara differentiability (gH-differentiability) of fuzzy functions they resolved these difficulties. More recently, Ghosh [10, 11, 12] proposed (quasi-)Newton methods for finding efficient solutions of optimization problems with interval-valued objective functions.

The Newton method proposed in [6, 22] is well defined only when the optimization problem is convex. Moreover, for large scale optimization problems, computing the Hessian matrix is very hard and cost effective. Therefore, in this paper, we are going to provide a quasi-Newton method to resolve the computational cost in computing the Hessian of the Newton method [6, 22] for unconstrained fuzzy optimization problems. Resolving the computational cost is done through approximating the Hessian matrix by another matrix that is available at lower cost. Quasi-Newton methods are among the most widely used methods for solving nonlinear optimization problems, see [2, 21]. They are effective in solving a wide variety of small to mid-size optimization problems, in particular when the Hessian is hard to compute. There are many different quasi-Newton methods, but they are all based on approximating the Hessian matrix by another matrix that is available at lower cost. Here, we extend the well-known BFGS method to fuzzy optimization problems. There are several advantages in this approach. At first, an approximation matrix can be found by using

*Corresponding author.
Email: Ghaznavi@shahroodut.ac.ir (M. Ghaznavi).

only first-derivatives information. Second, for problems that their Hessian is hard to compute, the proposed algorithm works efficiently. We show that the method converges superlinearly.

The outline of this paper is as follows. In Section 2, some basic properties and arithmetics of fuzzy sets are introduced. In Section 3, some results on gH-differentiability of fuzzy functions and fuzzy optimization are provided. The proposed quasi-Newton method is given in Section 4. To illustrate the efficiency of the proposed method, numerical examples are presented in Section 5. Finally, conclusions and suggestions for future research are given in Section 6.

2 Preliminaries and Basic Definitions

Let $\mathbb{R}$ be the set of real numbers. A fuzzy set u on $\mathbb{R}$ is a mapping $u : \mathbb{R} \to [0, 1]$. A fuzzy set u is characterized by its membership function $\mu_u : \mathbb{R} \to [0, 1]$, which associates with each x in $\mathbb{R}$, a real number $\mu_u(x)$ in $[0, 1]$. Let u be a fuzzy set. The α-cut or α- level of the fuzzy set u is given by $[u]^\alpha = \{x \in \mathbb{R} : u(x) \geq \alpha\}$. Also, the support of u, is denoted by $S(u) = \{x \in \mathbb{R} : u(x) > 0\}$. The closure of support u defines the 0-level of u, that is, $[u]^0 = cl(S(u))$.

Definition 2.1. [3] A fuzzy set u on $\mathbb{R}$ that satisfies the following properties, is called a fuzzy number:

(i) u is normal, i.e. there exists $x_0 \in \mathbb{R}$ such that $u(x_0) = 1$;

(ii) u if a fuzzy convex set, i.e., $u(\lambda x + (1 - \lambda)y) \geq \min\{u(x), u(y)\}$, whenever $x, y \in \mathbb{R}$ and $\lambda \in [0, 1]$;

(iii) u is upper semi-continuous on $\mathbb{R}$;

(iv) $[u]^0$ is a compact set.

Let $F(\mathbb{R})$ denote the family of all fuzzy numbers on $\mathbb{R}$. By definition, it can be seen that $[u]^\alpha$ is a compact interval in $\mathbb{R}$, for all $\alpha \in [0, 1]$, and therefore the α-level of a fuzzy number u is denoted by $[u]^\alpha = [\underline{u}_\alpha, \overline{u}_\alpha]$, where $\underline{u}_\alpha, \overline{u}_\alpha \in \mathbb{R}$ for all $\alpha \in [0, 1]$.

Let u and v be two fuzzy numbers. Using the α-level sets, their addition and scalar multiplication in $F(\mathbb{R})$ are defined as follows, respectively:

$$[u + v]^\alpha = [\underline{u}_\alpha + \underline{v}_\alpha, \overline{u}_\alpha + \overline{v}_\alpha] \tag{2.1}$$

and

$$[\lambda u]^\alpha = [\min\{\lambda\underline{u}_\alpha, \lambda\overline{u}_\alpha\}, \max\{\lambda\underline{u}_\alpha, \lambda\overline{u}_\alpha\}], \tag{2.2}$$

where $\lambda \in \mathbb{R}$ and $\alpha \in [0, 1]$.

Definition 2.2. [3] Triangular fuzzy numbers are a special type of fuzzy numbers which are defined as $u = (a, b, c)$, where a, b and c are three real numbers and their membership function is defined as:

$$\mu_u(x) = \begin{cases} \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & otherwise. \end{cases} \tag{2.3}$$

The α-level set of a triangular fuzzy number $u = (a, b, c)$ is given by:

$$[u]^\alpha = [(1 - \alpha)a + \alpha b, (1 - \alpha)c + \alpha b], \ \forall\alpha \in [0, 1].$$

Definition 2.3. [3] Let u and v be two fuzzy numbers in $F(\mathbb{R})$. Hence $[u]^\alpha = [\underline{u}^\alpha, \overline{u}^\alpha]$ and $[v]^\alpha = [\underline{v}^\alpha, \overline{v}^\alpha]$ are two intervals in $\mathbb{R}$, for all $\alpha \in [0, 1]$. We define

$$u \preceq v \leftrightarrow [u]^\alpha \preceq [v]^\alpha, \forall\alpha \in [0, 1] \leftrightarrow \underline{u}^\alpha \leq \underline{v}^\alpha \text{ and } \overline{u}^\alpha \leq \overline{v}^\alpha, \forall\alpha \in [0, 1]$$

and

$$u \prec v \leftrightarrow u \preceq v \text{ and } u \neq v \leftrightarrow [u]^\alpha \preceq [v]^\alpha \ \forall\alpha \in [0, 1], \text{ and } \exists\alpha^* \in [0, 1] \text{ s.t. } \underline{u}^{\alpha^*} < \underline{v}^{\alpha^*} \text{ or } \overline{u}^{\alpha^*} < \overline{v}^{\alpha^*}.$$

Definition 2.4. [24] Given $u, v \in F(\mathbb{R})$. The fuzzy number w is called the generalized Hukuhara difference (gH-difference for short) between u and v, if

$$u \ominus_{gh} v = w \Longleftrightarrow \begin{cases} (i)\ u = v + w \qquad or \\ (ii)\ v = u + (-1)w. \end{cases}$$

Using the α-levels we have

$$[u \ominus_{gh} v]^{\alpha} = [u]^{\alpha} \ominus_{gh} [v]^{\alpha} = [\min\{\underline{u}_{\alpha} - \underline{v}_{\alpha}, \overline{u}_{\alpha} - \overline{v}_{\alpha}\}, \max\{\underline{u}_{\alpha} - \underline{v}_{\alpha}, \overline{u}_{\alpha} - \overline{v}_{\alpha}\}], \quad \forall \alpha \in [0,1]$$

where $[u]^{\alpha} \ominus_{gh} [v]^{\alpha}$ is the gH-difference between two intervals (see [24, 25]).

3 Differentiable Fuzzy Functions and Fuzzy Optimization

In this section, at first we introduce the concept of differentiability of fuzzy functions. Later, we consider fuzzy optimization problems, and define nondominated solutions of a fuzzy optimization problem.

3.1 Differentiable Fuzzy Functions

Henceforth, X denotes an open subset of $\mathbb{R}^n$. A mapping $F : X \longrightarrow F(\mathbb{R})$ is said to be a fuzzy function defined on X. Let X_c denote the family of all bounded closed intervals in $\mathbb{R}$. We associate with each fuzzy function $F : X \to F(\mathbb{R})$, the family of interval-valued functions $F_{\alpha} : X \to X_c$ given by $F_{\alpha}(x) = [F(x)]^{\alpha}$. For each $\alpha \in [0,1]$, we denote $F_{\alpha}(x) = [F(x)]^{\alpha} = \left[\underline{f}^{\alpha}(x), \overline{f}^{\alpha}(x)\right]$. The endpoint functions $\underline{f}^{\alpha}, \overline{f}^{\alpha} : X \longrightarrow \mathbb{R}$ are said to be upper and lower functions of $F_{\alpha}(x)$, respectively.

Definition 3.1. [4] Let $X \subset \mathbb{R}$ and $F : X \longrightarrow F(\mathbb{R})$ be a fuzzy function. Also, assume that $x_0 \in X$ and h be such that $x_0 + h \in X$. The generalized Hukuhara derivative (gH-derivative) of F at x_0 is defined as

$$F'(x_0) = \lim_{h \to 0} \frac{F(x_0 + h) \ominus_{gh} F(x_0)}{h}. \tag{3.1}$$

If $F'(x_0) \in F(\mathbb{R})$ satisfying (3.1) exists, then F is said to be generalized Hukuhara differentiable (gH-differentiable) at x_0. If F is gH-differentiable at any $x \in X$, we say that F is gH-differentiable over X.

Definition 3.2. [25] Let X be an open set in $\mathbb{R}$. An interval-valued function $F : X \to X_c$ is gH-differentiable at $x_0 \in X$, if (3.1) exists with respect to the limit in the metric space (X_c, H), where the difference is given by the gH-difference between intervals.

Theorem 3.3. [1] *Let $F : X \longrightarrow F(\mathbb{R})$ be a fuzzy function. If F is gH-differentiable, then the interval-valued function $F_{\alpha} : X \longrightarrow X_c$ is gH-differentiable for each $\alpha \in [0,1]$. Moreover*

$$F'_{\alpha}(x) = [F'(x)]^{\alpha} = \left[(\underline{f}^{\alpha})'(x), (\overline{f}^{\alpha})'(x)\right].$$

Theorem 3.4. [6] *Let $F : X \longrightarrow F(\mathbb{R})$ be a fuzzy function. If F is gH-differentiable at $x_0 \in X$ then, for each $\alpha \in [0,1]$, one of the following cases holds:*

a) $\underline{f}^{\alpha}$ and $\overline{f}^{\alpha}$ are differentiable at x_0 and

$$[F'(x_0)]^{\alpha} = \left[\min\{(\underline{f}^{\alpha})'(x_0), (\overline{f}^{\alpha})'(x_0)\}, \max\{(\underline{f}^{\alpha})'(x_0), (\overline{f}^{\alpha})'(x_0)\}\right];$$

b) $(\underline{f}^{\alpha})'_{-}(x_0), (\overline{f}^{\alpha})'_{-}(x_0), (\underline{f}^{\alpha})'_{+}(x_0)$ and $(\overline{f}^{\alpha})'_{+}(x_0)$ exist and satisfy $(\underline{f}^{\alpha})'_{-}(x_0) = (\overline{f}^{\alpha})'_{+}(x_0)$ and $(\underline{f}^{\alpha})'_{+}(x_0) = (\overline{f}^{\alpha})'_{-}(x_0)$. Moreover

$$\begin{aligned} [F'(x_0)]^{\alpha} &= \left[\min\{(\underline{f}^{\alpha})'_{-}(x_0), (\overline{f}^{\alpha})'_{-}(x_0)\}, \max\{(\underline{f}^{\alpha})'_{-}(x_0), (\overline{f}^{\alpha})'_{-}(x_0)\}\right] \\ &= \left[\min\{(\underline{f}^{\alpha})'_{+}(x_0), (\overline{f}^{\alpha})'_{+}(x_0)\}, \max\{(\underline{f}^{\alpha})'_{+}(x_0), (\overline{f}^{\alpha})'_{+}(x_0)\}\right]. \end{aligned}$$

Definition 3.5. [1] Let $F : X \longrightarrow F(\mathbb{R})$ be a fuzzy function and let $x^0 = (x_1^0, x_2^0, \cdots, x_n^0)$ be a fixed element of X. Consider the fuzzy function $h(x_i) = F(x_1^0, \cdots, x_{i-1}^0, x_i, x_{i+1}^0, \cdots, x_n^0)$. If h is gH-differentiable at x_i^0, then we say that F has the ith partial gH-derivative at x^0 (denoted by $\frac{\partial F}{\partial x_i}(x^0)$) and $\frac{\partial F}{\partial x_i}(x^0) = (h')(x_i^0)$.

Definition 3.6. [1] Let F be a fuzzy function defined on $X \subseteq \mathbb{R}^n$ and let $x^0 = (x_1^0, x_2^0, \ldots, x_n^0) \in X$ be a fixed element of X. F is said to be gH-differentiable at x^0 if all the partial gH-derivatives $\frac{\partial F}{\partial x_1}(x^0)$, $\frac{\partial F}{\partial x_2}(x^0), \ldots,$ $\frac{\partial F}{\partial x_n}(x^0)$ exist in some neighborhood of x^0 and are continuous at x^0.

Definition 3.7. [1] Let $F : X \longrightarrow F(\mathbb{R})$ be a fuzzy function. The gradient of F at x^0, denoted by $\tilde{\nabla} F(x^0)$, is defined by

$$\tilde{\nabla} F(x^0) = \left((\frac{\partial F}{\partial x_1})(x_0), (\frac{\partial F}{\partial x_2})(x_0), \cdots, (\frac{\partial F}{\partial x_n})(x_0) \right).$$

The α-level set of $\tilde{\nabla} F(x^0)$ is defined and denoted by

$$[\tilde{\nabla} F(x^0)]^\alpha = \left([\frac{\partial F}{\partial x_1}]^\alpha(x_0), [\frac{\partial F}{\partial x_2}]^\alpha(x_0), \cdots, [\frac{\partial F}{\partial x_n}]^\alpha(x_0) \right),$$

where

$$[\frac{\partial F}{\partial x_i}]^\alpha = [\frac{\partial \underline{F}^\alpha}{\partial x_i}, \frac{\partial \overline{F}^\alpha}{\partial x_i}].$$

Theorem 3.8. [6] *Let $F : X \longrightarrow F(\mathbb{R})$ be a fuzzy function. If F is gH-differentiable at $x_0 \in X$ then, for each $\alpha \in [0,1]$, the real-valued function $\underline{f}^\alpha + \overline{f}^\alpha : X \longrightarrow \mathbb{R}$ is differentiable at x_0. Moreover,*

$$\frac{\partial \underline{F}^\alpha}{\partial x_i}(x_0) + \frac{\partial \overline{F}^\alpha}{\partial x_i}(x_0) = \frac{\partial \left(\underline{f}^\alpha + \overline{f}^\alpha \right)}{\partial x_i}(x_0). \tag{3.2}$$

Definition 3.9. [6] Let $F : X \longrightarrow F(\mathbb{R})$ be a fuzzy function. If gradient of F, $\tilde{\nabla} F$, is itself gH-differentiable at x^0, that is, for each i, the function $\frac{\partial F}{\partial x_i} : X \longrightarrow F(\mathbb{R})$ is gH-differentiable at x^0, then we say that F is twice gH-differentiable at x^0. Denote the gH-partial derivative of $\frac{\partial F}{\partial x_i}$ by

$$\tilde{D}_{ij}^2 F(x^0) = \frac{\partial^2 F}{\partial x_i x_j}(x^0), \qquad i \neq j,$$

and

$$\tilde{D}_{ii}^2 F(x^0) = \frac{\partial^2 F}{\partial x_i^2}(x^0), \qquad i = j.$$

If for each $i, j = 1, 2, \ldots, n$, the cross-partial derivative $\frac{\partial^2 F}{\partial x_i x_j}$ is continuous from X to $F(\mathbb{R})$, we say that F is twice continuously differentiable.

Theorem 3.10. [6] *Let $F : X \longrightarrow F(\mathbb{R})$ be a fuzzy function. If F is m-times gH-differentiable at $x_0 \in X$ then, for each $\alpha \in [0,1]$, the real-valued function $\underline{f}^\alpha + \overline{f}^\alpha : X \longrightarrow \mathbb{R}$ is m-times differentiable at x_0.*

3.2 Fuzzy Optimization

We consider the following fuzzy optimization problem (FOP):

$$(FOP) \quad \min_{x \in X} \quad F(x) \tag{3.3}$$

where the objective function $F : X \longrightarrow F(\mathbb{R})$ is a fuzzy-valued function and $X \subseteq \mathbb{R}^n$ is the domain of F which is assumed to be an open set. In the remainder of the paper we assume that F is gH-differentiable.

Definition 3.11. [22] Let $X \subset \mathbb{R}^n$ be an open set. We say that $x^* \in X$ is a locally nondominated solution of FOP (3.3) if there exists no $x \in N_\epsilon(x^*) \cap X$ such that $F(x) \prec F(x^*)$, where $N_\epsilon(x^*)$ is an ϵ-neighborhood of x^*.

Theorem 3.12. [6] *Let $X \subset \mathbb{R}^n$ be an open set and $F : X \longrightarrow F(\mathbb{R})$ be a fuzzy function. If x^* is a local minimizer of the real-valued function $\underline{f}^\alpha + \overline{f}^\alpha$, for all $\alpha \in [0,1]$, then x^* is a locally nondominated solution of the FOP (3.3).*

Theorem 3.13. *Let $\alpha \in [0,1]$ and the real-valued function $\underline{f}^\alpha + \overline{f}^\alpha : X \longrightarrow \mathbb{R}$ be differentiable at x_0. If there is a vector d such that $\left(\nabla(\overline{f}^\alpha + \underline{f}^\alpha)(x_0)\right)^T d < 0$, then there exists a $\delta > 0$ such that $(\underline{f}^\alpha + \overline{f}^\alpha)(x_0 + \lambda d) < (\underline{f}^\alpha + \overline{f}^\alpha)(x_0)$ for each $\lambda \in (0, \delta)$. So that, d is a descent direction of $\underline{f}^\alpha + \overline{f}^\alpha$.*

Proof. Let $\alpha \in [0,1]$ and $\left(\nabla(\overline{f}^\alpha + \underline{f}^\alpha)(x_0)\right)^T d < 0$. By the differentiability of $\overline{f}^\alpha + \underline{f}^\alpha$ at x_0, we have

$$(\overline{f}^\alpha + \underline{f}^\alpha)(x_0 + \lambda d) = (\overline{f}^\alpha + \underline{f}^\alpha)(x_0) + \lambda \left(\nabla(\overline{f}^\alpha + \underline{f}^\alpha)(x_0)\right)^T d + \lambda \|d\| o(x_0; \lambda d),$$

where $o(x_0; \lambda d) \to 0$ as $\lambda \to 0$. We get

$$\frac{(\overline{f}^\alpha + \underline{f}^\alpha)(x_0 + \lambda d) - (\overline{f}^\alpha + \underline{f}^\alpha)(x_0)}{\lambda} = \left(\nabla(\overline{f}^\alpha + \underline{f}^\alpha)(x_0)\right)^T d + \|d\| o(x_0; \lambda d), \quad \lambda \neq 0.$$

Since $\left(\nabla(\overline{f}^\alpha + \underline{f}^\alpha)(x_0)\right)^T d < 0$ and $o(x_0; \lambda d) \to 0$ as $\lambda \to 0$, there is a $\delta > 0$ such that $\left(\nabla(\overline{f}^\alpha + \underline{f}^\alpha)(x_0)\right)^T d + \|d\| o(x_0; \lambda d) < 0$ for all $\lambda \in (0, \delta)$. Therefore, the proof is completed. □

We assume that at each point x_k, we can calculate $F(x_k)$, $\tilde{\nabla} F(x_k)$. Since F is gH-differentiable, according to Theorems 3.8 and 3.10 we can calculate $\nabla(\underline{f}^\alpha + \overline{f}^\alpha)(x_k)$.

4 Quasi-Newton Method

In order to solve the fuzzy optimization problem (3.3), the Newton method proposed in [6, 22] requires computation of the Hessian matrix. When $\nabla^2(\overline{f}^\alpha + \underline{f}^\alpha)(x_k)$ is not positive definite, the Newton direction $p^\alpha(x_k) = -\left[\nabla^2(\overline{f}^\alpha + \underline{f}^\alpha)(x_k)\right]^{-1} \nabla(\overline{f}^\alpha + \underline{f}^\alpha)(x_k)$ may not even be defined, since $\left[\nabla^2(\overline{f}^\alpha + \underline{f}^\alpha)(x_k)\right]^{-1}$ may not exist. Even when the Newton direction $p^\alpha(x_k)$ is defined, it may not satisfy the descent property $\nabla(\overline{f}^\alpha + \underline{f}^\alpha)(x_k)^T p^\alpha(x_k) < 0$, in this case it is unsuitable as a search direction. The main drawback of the Newton method given in [6, 22], is the need of the Hessian $\nabla^2(\overline{f}^\alpha + \underline{f}^\alpha)(x_k)$. However, explicit computation of this matrix of second derivatives is sometimes, though not always, a cumbersome, error-prone and expensive process. Therefore, in this section, we will provide a quasi-Newton method for the FOP (3.3) to sidestep the high computational expense of the Newton method. The most popular quasi-Newton algorithm for crisp optimization problems is the BFGS method [2, 21]. We extend this algorithm for the unconstrained fuzzy optimization problem (3.3). In this algorithm, computation of $\nabla^2(\overline{f}^\alpha + \underline{f}^\alpha)$ is not required. Instead of the true Hessian $\nabla^2(\overline{f}^\alpha + \underline{f}^\alpha)(x_k)$, we will use an approximation $(\underline{B}^\alpha + \overline{B}^\alpha)(x_k)$, which is updated after each step to take account of the additional knowledge gained during the step.

Consider the following quadratic forms in x, around x_k and x_{k+1} for $\alpha \in [0,1]$,

$$m_k^\alpha(x) = (\underline{f}^\alpha + \overline{f}^\alpha)(x_k) + (x - x_k)^T \nabla(\underline{f}^\alpha + \overline{f}^\alpha)(x_k) + \frac{1}{2}(x - x_k)^T (\underline{B}^\alpha + \overline{B}^\alpha)(x_k)(x - x_k)$$

and

$$m_{k+1}^\alpha(x) = (\underline{f}^\alpha + \overline{f}^\alpha)(x_{k+1}) + (x - x_{k+1})^T \nabla(\underline{f}^\alpha + \overline{f}^\alpha)(x_{k+1}) + \frac{1}{2}(x - x_{k+1})^T (\underline{B}^\alpha + \overline{B}^\alpha)(x_{k+1})(x - x_{k+1}).$$

The matrix $(\underline{B}^\alpha + \overline{B}^\alpha)(x_{k+1})$ will be a good approximation of $\nabla^2(\underline{f}^\alpha + \overline{f}^\alpha)(x_k)$ if the following condition holds:

$$\nabla m_k^\alpha(x_k) = \nabla m_{k+1}^\alpha(x_k). \tag{4.1}$$

After simplifying (4.1) we have

$$\nabla(\underline{f}^\alpha + \overline{f}^\alpha)(x_k) = \nabla(\underline{f}^\alpha + \overline{f}^\alpha)(x_{k+1}) + (\underline{B}^\alpha + \overline{B}^\alpha)(x_{k+1})(x_k - x_{k+1}).$$

So,

$$(\underline{B}^{\alpha}+\overline{B}^{\alpha})(x_{k+1})(x_{k+1}-x_k)=\nabla(\underline{f}^{\alpha}+\overline{f}^{\alpha})(x_{k+1})-\nabla(\underline{f}^{\alpha}+\overline{f}^{\alpha})(x_k).$$

We define

$$y^{\alpha}(x_k)=\nabla(\underline{f}^{\alpha}+\overline{f}^{\alpha})(x_{k+1})-\nabla(\underline{f}^{\alpha}+\overline{f}^{\alpha})(x_k),\qquad s_k=x_{k+1}-x_k$$

for $\alpha\in[0,1]$. So we have

$$(\underline{B}^{\alpha}+\overline{B}^{\alpha})(x_{k+1})s_k=y^{\alpha}(x_k), \tag{4.2}$$

which is the fuzzy secant condition.

Theorem 4.1. *Let $\alpha\in[0,1]$. If $d^{\alpha}(x_k)$ is a descent direction of $\underline{f}^{\alpha}+\overline{f}^{\alpha}$, and*

$$\left(\nabla(\underline{f}^{\alpha}+\overline{f}^{\alpha})(x_{k+1})\right)^T d^{\alpha}(x_k)\geq c\left(\nabla(\underline{f}^{\alpha}+\overline{f}^{\alpha})(x_k)\right)^T d^{\alpha}(x_k)$$

for some $c<1$, then $y^{\alpha}(x_k)^T s_k>0$.

Proof. Let $x_{k+1}=x_k+t_k d^{\alpha}(x_k)$ be chosen in such way that

$$\left(\nabla(\underline{f}^{\alpha}+\overline{f}^{\alpha})(x_{k+1})\right)^T d^{\alpha}(x_k)\geq c\left(\nabla(\underline{f}^{\alpha}+\overline{f}^{\alpha})(x_k)\right)^T d^{\alpha}(x_k),$$

for some $c<1$. Since $d^{\alpha}(x_k)$ is a descent direction so

$$\left(\nabla(\underline{f}^{\alpha}+\overline{f}^{\alpha})(x_k)\right)^T d^{\alpha}(x_k)<0.$$

Hence

$$\begin{aligned}y^{\alpha}(x_k)^T s_k&=\left(\nabla(\underline{f}^{\alpha}+\overline{f}^{\alpha})(x_{k+1})-\nabla(\underline{f}^{\alpha}+\overline{f}^{\alpha})(x_k)\right)^T t_k d^{\alpha}(x_k)\\&\geq t_k c\left(\nabla(\underline{f}^{\alpha}+\overline{f}^{\alpha})(x_k)\right)^T d^{\alpha}(x_k)-t_k\left(\nabla(\underline{f}^{\alpha}+\overline{f}^{\alpha})(x_k)\right)^T d^{\alpha}(x_k)\\&=t_k(c-1)\left(\nabla(\underline{f}^{\alpha}+\overline{f}^{\alpha})(x_k)\right)^T d^{\alpha}(x_k)>0.\end{aligned}$$

□

Relation $y^{\alpha}(x_k)^T s_k>0$ is the fuzzy curvature condition. The fuzzy curvature condition for all $\alpha\in[0,1]$ implies that

$$\int_0^1 y^{\alpha}(x_k)^T s_k d\alpha=y(x_k)^T s_k>0. \tag{4.3}$$

This relation is called the curvature condition.

Since in the Newton method [6, 22], the Hessian matrix $\nabla^2(\overline{f}^{\alpha}+\underline{f}^{\alpha})$ for all $\alpha\in[0,1]$ is a symmetric positive definite matrix, it is reasonable that the approximation matrix $(\underline{B}^{\alpha}+\overline{B}^{\alpha})(x_{k+1})$ be positive definite for all $\alpha\in[0,1]$, as well. This will be possible only if the fuzzy curvature condition holds, as is easily seen by premultiplying (4.2) by s_k^T. Note that if Theorem 4.1 holds, then $(\underline{B}^{\alpha}+\overline{B}^{\alpha})(x_{k+1})$ is positive definite. Due to the above relations, the BFGS method updating formula for the FOP (3.3) follows as in the classical BFGS method updating formula. Therefor, we have

$$(\underline{B}^{\alpha}+\overline{B}^{\alpha})(x_{k+1})=(\underline{B}^{\alpha}+\overline{B}^{\alpha})(x_k)-\frac{(\underline{B}^{\alpha}+\overline{B}^{\alpha})(x_k)s_k s_k^T(\underline{B}^{\alpha}+\overline{B}^{\alpha})^T(x_k)}{s_k^T(\underline{B}^{\alpha}+\overline{B}^{\alpha})(x_k)s_k}+\frac{y^{\alpha}(x_k)y^{\alpha}(x_k)^T}{y^{\alpha}(x_k)^T s_k} \tag{4.4}$$

for all $\alpha\in[0,1]$. The new matrix $(\underline{B}^{\alpha}+\overline{B}^{\alpha})(x_{k+1})$ will be positive definite only if $y^{\alpha}(x_k)^T s_k>0$.

Theorem 4.2. *Let $\alpha\in[0,1]$ and $(\underline{B}^{\alpha}+\overline{B}^{\alpha})(x_k)$ be a symmetric positive-definite matrix. Assume that $(\underline{B}^{\alpha}+\overline{B}^{\alpha})(x_{k+1})$ is obtained from $(\underline{B}^{\alpha}+\overline{B}^{\alpha})(x_k)$ using the BFGS update formula (4.4). Then $(\underline{B}^{\alpha}+\overline{B}^{\alpha})(x_{k+1})$ is positive definite if and only if $y^{\alpha}(x_k)^T s_k>0$, for all $\alpha\in[0,1]$.*

Proof. The proof is similar to that of Lemma 12.10 in [13], and hence it is omitted here. □

Consider the following quadratic model of $(\overline{f}^{\alpha} + \underline{f}^{\alpha})(x)$ at x_k:

$$m_k^{\alpha}(d^{\alpha}) = (\overline{f}^{\alpha} + \underline{f}^{\alpha})(x_k) + \nabla(\overline{f}^{\alpha} + \underline{f}^{\alpha})(x_k)^T d^{\alpha} + \frac{1}{2} d^{\alpha T} (\underline{B}^{\alpha} + \overline{B}^{\alpha})(x_k) d^{\alpha}. \tag{4.5}$$

Here, for $\alpha \in [0,1]$, $(\underline{B}^{\alpha} + \overline{B}^{\alpha})(x_k)$ is an $n \times n$ symmetric positive definite matrix that will be updated at every iteration. The minimizer $d^{\alpha}(x_k)$ of this convex model, can be written explicitly as

$$d^{\alpha}(x_k) = -\left((\underline{B}^{\alpha} + \overline{B}^{\alpha})(x_k)\right)^{-1} \nabla(\overline{f}^{\alpha} + \underline{f}^{\alpha})(x_k). \tag{4.6}$$

We define,

$$\varphi(x) = \int_0^1 (\underline{f}^{\alpha} + \overline{f}^{\alpha})(x) \mathrm{d}\alpha, \quad B(x) = \int_0^1 (\underline{B}^{\alpha} + \overline{B}^{\alpha})(x) \mathrm{d}\alpha.$$

As the fuzzy function F is continuously gH-differentiable, $\underline{f}^{\alpha} + \overline{f}^{\alpha}$ is continuously gH-differentiable for all $\alpha \in [0,1]$. Thus φ is also continuously gH-differentiable. Therefore from (4.6), we have

$$d(x_k) = -B(x_k)^{-1} \nabla \varphi(x_k),$$

which is used as the search direction. The new iterate is

$$x_{k+1} = x_k + t_k d(x_k), \tag{4.7}$$

where the step length t_k is chosen to satisfy the following relationships:

$$\varphi(x_k + t_k d(x_k)) \leq \varphi(x_k) + c_1 t_k \left(\nabla \varphi(x_k)\right)^T d(x_k)$$

and

$$\left(\nabla \varphi(x_k + t_k d(x_k))\right)^T d(x_k) \geq c_2 \left(\nabla \varphi(x_k)\right)^T d(x_k),$$

with $0 < c_1 < c_2 < 1$. Therefore, by Theorems 4.1 and 4.2 we have $y(x_k)^T s_k > 0$ and $B(x_{k+1})$ is positive definite.

Note that, the BFGS method updating formula (4.4) for the FOP (3.3) changes to:

$$B(x_{k+1}) = B(x_k) - \frac{B(x_k) s_k s_k^T B(x_k)}{s_k^T B(x_k) s_k} + \frac{y(x_k) y(x_k)^T}{y(x_k)^T s_k}. \tag{4.8}$$

In Algorithm 1, we describe the quasi-Newton algorithm for fuzzy optimization problems.

Algorithm 1: Quasi-Newton algorithm for fuzzy optimization problems

Step 0. Let $x^{(0)}$ be the initial decision vector chosen from X. Given c_1, c_2, tolerance $\epsilon > 0$ and Hessian approximation $B(x^{(0)})$. Set $k := 0$ and define $J = \{1/2^n \mid n = 0, 1, 2, \dots\}$.

Step 1. If $\|x^{(k+1)} - x^{(k)}\| \leq \epsilon$, then stop. Else, go to Step 2.

Step 2. Compute search direction $d(x^{(k)}) = -(B(x^{(k)}))^{-1} \nabla \varphi(x^{(k)})$.

Step 3. Define $x^{(k+1)} := x^{(k)} + t_k d(x^{(k)})$, where t_k, as the largest $t \in J$, is chosen such that satisfies in the following relationships:

$$\varphi(x^{(k)} + t d(x^{(k)})) \leq \varphi(x^{(k)}) + c_1 t \left(\nabla \varphi(x^{(k)})\right)^T d(x^{(k)}),$$

and

$$\left(\nabla \varphi\left(x^{(k)} + t_k d(x^{(k)})\right)\right)^T d(x^{(k)}) \geq c_2 \left(\nabla \varphi(x^{(k)})\right)^T d(x^{(k)}),$$

with $0 < c_1 < c_2 < 1$.

Step 4. Generate $B(x^{(k+1)})$ by means of (4.8) and set $k := k + 1$. Go to Step 1.

In the following theorem, we consider convergence analysis of the proposed quasi-Newton method.

Theorem 4.3. *Suppose that F is a two times continuously gH-differentiable fuzzy function defined on $\mathbb{R}^n$. Let x_0 be some given initial point and let $\{x_k\}$ be defined by $x_{k+1} := x_k + t_k d(x_k)$. Assume that*

(i) the set $S = \{x : \varphi(x) \leq \varphi(x_0)\}$ is bounded;

(ii) φ, $\nabla\varphi$ and $\nabla^2\varphi$ are continuous for all $x \in S$;

(iii) $\nabla^2\varphi$ is positive definite for all x;

(iv) the search directions $\{d(x_k)\}$ are computed using $B(x_k)d(x_k) = -\nabla\varphi(x_k)$, where $B(x_0) = I$, and the matrices $B(x_k)$ are updated using the formula (4.8);

(v) the step lengths t_k satisfy

$$\varphi(x_k + t_k d(x_k)) \leq \varphi(x_k) + c_1 t_k \big(\nabla\varphi(x_k)\big)^T d(x_k),$$

$$\big(\nabla\varphi(x_k + t_k d(x_k))\big)^T d(x_k) \geq c_2 \nabla\varphi(x_k)^T d(x_k),$$

with $0 < c_1 < c_2 < 1$, and the line search algorithm uses the step length $t_k = 1$ whenever possible.

Then, the quasi-Newton method is well defined for all k and

$$\lim_{k\to\infty} x_k = x^*,$$

that is, the method converges to x^* and the rate of convergence is superlinear.

Proof. Since F is two times continuously gH-differentiable fuzzy function, then φ is tow times continuously differentiable function. So, the proof follows as in the classical proof of the quasi-Newton method [2, 13, 21]. □

5 Numerical Examples

In this section we state illustrative examples to justify the proposed algorithm. All examples are executed within MATLAB (R2013a). We use the stopping criterion $\|x^{(k+1)} - x^{(k)}\| \leq \epsilon$ for some specified parameter $\epsilon > 0$ in order to stop at the point $x^{(k)}$ and we set $B(x^{(0)}) = I$.

Example 5.1. *Consider the following nonlinear programming problem with fuzzy parameters:*

$$(FOP) \quad \min_{x\in\mathbb{R}^2} \quad F(x),$$

$$\begin{aligned} F(x) =& (-\frac{1}{2}, \frac{1}{2}, \frac{3}{2})x_1^4 + (0, \frac{25}{2}, 25)x_1^2 + (-\frac{99}{4}, -\frac{99}{10}, \frac{99}{20})x_1x_2 + (-2, 2, 6)x_2^2 \\ &+ (-40, -16, 8)x_1 + (-32, 16, 32). \end{aligned}$$

We have

$$\big(\overline{f}^{\alpha} + \underline{f}^{\alpha}\big)(x) = x_1^4 + 25x_1^2 - 19.8x_1x_2 + 4x_2^2 - 32x_1 + 32\alpha,$$

and

$$\int_0^1 \big(\overline{f}^{\alpha} + \underline{f}^{\alpha}\big)(x)\mathrm{d}\alpha = \big(\overline{f} + \underline{f}\big)(x) = x_1^4 + 25x_1^2 - 19.8x_1x_2 + 4x_2^2 - 32x_1 + 16.$$

Also

$$\nabla\big(\overline{f} + \underline{f}\big)(x) = \begin{pmatrix} 4x_1^3 + 50x_1 - 19.8x_2 - 32 \\ -19.8x_1 + 8x_2 \end{pmatrix}.$$

Then

$$\nabla^2\big(\overline{f} + \underline{f}\big)(x) = \begin{bmatrix} 12x_1^2 + 50 & -19.8 \\ -19.8 & 8 \end{bmatrix}.$$

A program in MATLAB (R2013a) is written following the Algorithm 1 and a summary of the results is provided. Consider an initial point $x^{(0)} = (0, 3)$ and stopping condition $\|x^{(k+1)} - x^{(k)}\| \leq 10^{-3}$. Nondominated solution of this problem is found at 7th iteration as $x^{(7)} = (1.9585, 4.8474)$. Table 1 exhibits the performance of Algorithm 1 in this example.

Table 1: Convergence of the quasi-Newton Algorithm 1 for Example 5.1

k	$x^{(k)}$	t_k	$\varphi(x^{(k)})$	$\nabla\varphi(x^{(k)})$	$d(x^{(k)}) = -(B(x^{(k)}))^{-1}\nabla\varphi(x^{(k)})$	$\|x^{(k+1)} - x^{(k)}\|$
0	$(0, 3)$	0.0156	52	$\begin{pmatrix}-91.4000\\24.0000\end{pmatrix}$	$(91.4000, -24.0000)$	1.4765
1	$(1.4281, 2.6250)$	0.2500	-21.2160	$\begin{pmatrix}-0.9179\\-7.2769\end{pmatrix}$	$(2.3441, 6.8267)$	1.8045
2	$(2.0142, 4.3317)$	1	-28.2694	$\begin{pmatrix}15.6247\\-5.2268\end{pmatrix}$	$(-0.0804, 0.5337)$	0.5398
3	$(1.9337, 4.8654)$	1	-30.0114	$\begin{pmatrix}-2.7243\\0.6352\end{pmatrix}$	$(0.0281, -0.0073)$	0.0290
4	$(1.9618, 4.8581)$	1	-30.0507	$\begin{pmatrix}0.1029\\0.0207\end{pmatrix}$	$(-0.0038, -0.0130)$	0.0136
5	$(1.9580, 4.8451)$	1	-30.0510	$\begin{pmatrix}-0.0043\\-0.0082\end{pmatrix}$	$(0.0005, 0.0023)$	0.0024
6	$(1.9585, 4.8474)$	1	-30.0510	$\begin{pmatrix}-10^{-3}\times 0.3526\\10^{-3}\times 0.0912\end{pmatrix}$	$(10^{-5}\times 0.2776, -10^{-5}\times 0.4741)$	$10^{-6}\times 5.4943$
7	$(1.9585, 4.8474)$		-30.0510	$\begin{pmatrix}10^{-5}\times 0.7850\\-10^{-5}\times 0.1750\end{pmatrix}$	$(-10^{-7}\times 0.7486, 10^{-7}\times 0.3342)$	

Table 2: Convergence of the Newton method [6] for Example 5.1

k	$x^{(k)}$	$\varphi(x^{(k)})$	$\nabla\varphi(x^{(k)})$	$d(x^{(k)}) = -\nabla\varphi(x^{(k)})(\nabla^2\varphi(x^{(k)}))^{-1}$	$\|x^{(k+1)} - x^{(k)}\|$
0	$(0, 3)$	52	$\begin{pmatrix}-91.4000\\24.0000\end{pmatrix}$	$(32.1608, 76.5980)$	83.0757
1	$(32.1608, 79.5980)$	$10^6\times 1.0693$	$\begin{pmatrix}10^5\times 1.3306\\10^5\times 0\end{pmatrix}$	$(-10.7194, -26.5305)$	28.6142
2	$(21.4414, 53.0675)$	$10^5\times 2.1091$	$\begin{pmatrix}10^4\times 3.9419\\10^4\times 0.0000\end{pmatrix}$	$(-7.1439, -17.6812)$	19.0699
3	$(14.2975, 35.3863)$	$10^4\times 4.1447$	$\begin{pmatrix}10^4\times 1.1673\\10^4\times 0.0000\end{pmatrix}$	$(-4.7567, -11.7727)$	12.6973
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
11	$(1.9585, 4.8474)$	-30.0510	$\begin{pmatrix}10^{-4}\times 0.1725\\10^{-4}\times 0\end{pmatrix}$	$(-10^{-6}\times 0.3668, -10^{-6}\times 0.9079)$	$10^{-7}\times 9.7918$
12	$(1.9585, 4.8474)$	-30.0510	$\begin{pmatrix}10^{-11}\times 0.3141\\10^{-11}\times 0\end{pmatrix}$	$(-10^{-12}\times 0.0668, -10^{-12}\times 0.1653)$	

The Newton method proposed in [6, 22] could be applied for this example. Employing the Newton method [6] with the same starting point and stopping criterion results in a slower convergence after 12 iterations, with $x^{(12)} = (1.9585, 4.8474)$. *The iterations of* $x^{(k)}$ *by Newton method [6] are given in Table 2. As it can be seen in this example, for fuzzy optimization problems that explicit computation of the Hessian matrix is hard, the proposed quasi-Newton method converges faster than the Newton method.*

Example 5.2. *Consider the following nonlinear programming problem with fuzzy parameters:*

$$(FOP) \quad \min_{x\in\mathbb{R}^2} \quad F(x),$$

where

$$F(x) = \quad \mathrm{e}^{(-\frac{1}{2},\frac{1}{2},\frac{3}{2})x_1+(-\frac{5}{4},-\frac{1}{2},\frac{1}{4})} + \mathrm{e}^{(-\frac{3}{2},-\frac{1}{2},\frac{1}{2})x_2+(-\frac{1}{4},\frac{1}{2},\frac{5}{4})} + (-2,1,2)x_1^2 + (-4,1,4)x_2^2 + (-3,-1,1)x_1x_2.$$

Using fuzzy arithmetic, we have:

$$\int_0^1 \left(\overline{f}^{\alpha} + \underline{f}^{\alpha}\right)(x)\mathrm{d}\alpha = \mathrm{e}^{x_1-1} + \mathrm{e}^{-x_2+1} + x_1^2 + x_2^2 - 2x_1x_2.$$

Also,

$$\nabla\left(\overline{f} + \underline{f}\right)(x) = \begin{pmatrix} \mathrm{e}^{x_1-1} + 2x_1 - 2x_2 \\ -\mathrm{e}^{-x_2+1} + 2x_2 - 2x_1 \end{pmatrix},$$

$$\nabla^2\left(\overline{f} + \underline{f}\right)(x) = \begin{bmatrix} \mathrm{e}^{x_1-1} + 2 & -2 \\ -2 & \mathrm{e}^{-x_2+1} + 2 \end{bmatrix}.$$

Consider an initial point $x^{(0)} = (-20, 15)$ *and stopping condition* $\|x^{(k+1)} - x^{(k)}\| \leq 10^{-3}$. *Nondominated solution of this problem is found at 11th iteration as* $x^{(11)} = (0.7961, 1.2039)$. *The iterations of* $x^{(k)}$ *are given in Table 3.*

Employing the Newton method [6] with the same starting point and stopping criterion results in a slower convergence after 19 iterations, with $x^{(19)} = (0.7961, 1.2039)$. *Table 4 exhibits the performance of the Newton algorithm [6] in this example. As it can be seen, the proposed quasi-Newton method converges faster than the Newton method.*

Example 5.3. *Consider the following nonlinear programming problem with fuzzy parameters:*

$$(FOP) \quad \min_{x\in\mathbb{R}^2} \quad F(x),$$

$$\begin{aligned} F(x) =& (-2,1,2)x_1^4 + (-12,-4,4)x_1^3 + (-\frac{25}{2}, \frac{25}{2}, \frac{75}{2})x_1^2 + (0,2,4)x_2^2 \\ &+ (-6,-2,2)x_1x_2 + (-48,-16,16)x_1 + (-32,16,32). \end{aligned}$$

We have,

$$\big(\overline{f}^{\alpha} + \underline{f}^{\alpha}\big)(x) = 2\alpha x_1^4 - 8x_1^3 + 25x_1^2 + 4x_2^2 - 4x_1x_2 - 32x_1 + 32\alpha,$$

and

$$\int_0^1 \big(\overline{f}^{\alpha} + \underline{f}^{\alpha}\big)(x)\mathrm{d}\alpha = \big(\overline{f} + \underline{f}\big)(x) = x_1^4 - 8x_1^3 + 25x_1^2 + 4x_2^2 - 4x_1x_2 - 32x_1 + 16.$$

Then

$$\nabla\big(\overline{f} + \underline{f}\big)(x) = \begin{pmatrix} 4x_1^3 - 24x_1^2 + 50x_1 - 4x_2 - 32 \\ 8x_2 - 4x_1 \end{pmatrix}.$$

Nondominated solution of this problem with the initial point $x^{(0)} = (1, 1)$ *and termination condition* $\|x^{(k+1)} - x^{(k)}\| \leq 10^{-3}$ *is found at 5th iteration as* $x^{(5)} = (2.0451, 1.0225)$. *The performance of Algorithm 1 is given in Table 5.*

If we apply the Newton algorithm [6] for this problem with the same initial point and the same termination condition, we will find the nondominated solution at the 16th iteration as $x^{(16)} = (1.9985, 0.9992)$.

Example 5.4. *Consider the following nonlinear programming problem with fuzzy parameters:*

$$(FOP) \quad \min_{x\in\mathbb{R}^3} \quad F(x),$$

where

$$F(x) = \quad (2.9, 3.0, 3.15)x_1^2 + (4.7, 5.0, 5.35)x_2^2 + (-2, 1, 2)x_3^2.$$

Using fuzzy arithmetic, it follows:

$$\int_0^1 \big(\overline{f}^{\alpha} + \underline{f}^{\alpha}\big)(x)\mathrm{d}\alpha = 6.025x_1^2 + 10.025x_2^2 + x_3^2.$$

Also,

$$\nabla\big(\overline{f} + \underline{f}\big)(x) = \begin{pmatrix} 12.05x_1 \\ 20.05x_2 \\ 2x_3 \end{pmatrix}.$$

Consider an initial point $x^{(0)} = (1, 1, 2)$ *and stopping condition* $\|x^{(k+1)} - x^{(k)}\| \leq 10^{-3}$. *Nondominated solution of this problem is found at 6th iteration as* $x^{(6)} = (10^{-7} \times 0.0274, 10^{-7} \times 0.0132, -10^{-7} \times 0.2396) \simeq (0, 0, 0)$. *Table 6 shows the performance of the proposed algorithm for this example.*

6 Conclusions

In this article, we extended the quasi-Newton method for solving unconstrained fuzzy optimization problems. In the proposed method, we found an approximation of the Hessian matrix to find a nondominated solution of a fuzzy optimization problem. An algorithmic procedure for the proposed approach accompanied by numerical examples were given to illustrate the efficiency of the proposed approach over the existing Newton method.

Table 3: Convergence of the quasi-Newton Algorithm 1 for Example 5.2

k	$x^{(k)}$	t_k	$\varphi(x^{(k)})$	$\nabla\varphi(x^{(k)})$	$d(x^{(k)}) = -(B(x^{(k)}))^{-1}\nabla\varphi(x^{(k)})$	$\|x^{(k+1)} - x^{(k)}\|$
0	$(-20, 15)$	0.2500	$10^3 \times 1.2250$	$\begin{pmatrix} -70.0000 \\ 70.0000 \end{pmatrix}$	$(70.0000, -70.0000)$	24.7487
1	$(-2.5000, -2.5000)$	0.5000	33.1456	$\begin{pmatrix} 0.0302 \\ -33.1154 \end{pmatrix}$	$(12.5817, 14.1700)$	9.4748
2	$(3.7908, 4.5850)$	0.5000	16.9531	$\begin{pmatrix} 14.7063 \\ 1.5606 \end{pmatrix}$	$(-5.4304, -0.6885)$	2.7369
3	$(1.0757, 4.2407)$	0.5000	11.1355	$\begin{pmatrix} -5.2516 \\ 6.2910 \end{pmatrix}$	$(0.3801, -0.9528)$	0.5129
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
10	$(0.7961, 1.2040)$	1	1.7974	$\begin{pmatrix} -10^{-3} \times 0.4133 \\ 10^{-3} \times 0.4744 \end{pmatrix}$	$(10^{-3} \times 0.0549, -10^{-3} \times 0.1295)$	$10^{-4} \times 1.4062$
11	$(0.7961, 1.2039)$		1.7974	$\begin{pmatrix} 10^{-6} \times 0.1934 \\ 10^{-6} \times 0.0850 \end{pmatrix}$	$(-10^{-6} \times 0.1824, -10^{-6} \times 0.1598)$	

Table 4: Convergence of the Newton method [6] for Example 5.2

k	$x^{(k)}$	$\varphi(x^{(k)})$	$\nabla\varphi(x^{(k)})$	$d(x^{(k)}) = -\nabla\varphi(x^{(k)})(\nabla^2\varphi(x^{(k)}))^{-1}$	$\|x^{(k+1)} - x^{(k)}\|$
0	$(-20, 15)$	$10^3 \times 1.2250$	$\begin{pmatrix} -70.0000 \\ 70.0000 \end{pmatrix}$	$(35.9663, 0.9633)$	35.9793
1	$(15.9663, 15.9663)$	$10^6 \times 3.1607$	$\begin{pmatrix} 10^6 \times 3.1607 \\ 10^6 \times -0.0000 \end{pmatrix}$	$(-1.0000, -10000)$	1.4142
2	$(14.9663, 14.9663)$	$10^6 \times 1.1627$	$\begin{pmatrix} 10^6 \times 1.1627 \\ -10^6 \times 0.0000 \end{pmatrix}$	$(-1.0000, -1.0000)$	1.4142
3	$(13.9663, 13.9663)$	$10^5 \times 4.2775$	$\begin{pmatrix} 10^5 \times 4.2775 \\ -10^5 \times 0.0000 \end{pmatrix}$	$(-1.0000, -1.0000)$	1.4142
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
18	$(0.7965, 1.2041)$	1.7974	$\begin{pmatrix} 10^{-3} \times 0.7018 \\ -10^{-3} \times 0.1945 \end{pmatrix}$	$(-10^{-3} \times 0.4040, -10^{-3} \times 0.2179)$	$10^{-4} \times 4.5901$
19	$(0.7961, 1.2039)$	1.7974	$\begin{pmatrix} 10^{-7} \times 0.6657 \\ -10^{-7} \times 0.1936 \end{pmatrix}$	$(-10^{-7} \times 0.3787, -10^{-7} \times 0.2002)$	

However, as it can be seen in Algorithm 1, the proposed procedure generates only one nondominated solution for a fuzzy optimization problem. Extending the method for finding the set of complete nondominated solutions can be a worthwhile direction for the future research. Moreover, we employed an exact line-search technique along the search direction. Employing an inexact line search technique [21] can be worth studying. Furthermore, future research can also be performed on extending the proposed quasi-Newton approach for unconstrained fuzzy multi-objective optimization problems. To this end, studying the references [7, 23] is recommended.

References

[1] Arana-Jiménez, M., Rufián-Lizana, A., Chalco-Cano, Y., and H. Román-Flores, Generalized convexity in fuzzy vector optimization through a linear ordering, *Information Sciences*, vol.312, pp.13–24, 2015.

[2] Bazaraa, M.S., Sherali, H.D., and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms*, John Wiley and Sons, 2013.

[3] Bector, C.R., and S. Chandra, *Fuzzy Mathematical Programming and Fuzzy Matrix Games*, Springer-Verlag, Berlin, 2005.

[4] Bede, B., and L. Stefani, Generalized differentiability of fuzzy-valued functions, *Fuzzy Sets and Systems*, vol.230, pp.119–141, 2013.

[5] Bellman, R.E., and L.A. Zadeh, Decision making in a fuzzy environment, *Management Science*, vol.17, pp.141–164, 1970.

[6] Chalco-Cano, Y., Silva, G.N., and A. Rufian-Lizana, On the Newton method for solving fuzzy optimization problems, *Fuzzy Sets and Systems*, vol.272, pp.60–69, 2015.

[7] Fliege, J., Grana Drummond, L.M., and B.F. Svaiter, Newton's method for multiobjective optimization, *SIAM Journal on Optimization*, vol.20, no.2, pp.602–626, 2009.

[8] Ganesan, K., and P. Veeramani, Fuzzy linear programming with trapezoidal fuzzy numbers, *Annals of Operations Research*, vol.143, pp.305–315, 2006.

[9] Ghaznavi, M., Soleimani, F., and N. Hoseinpoor, Parametric analysis in fuzzy number linear programming problems, *International Journal of Fuzzy Systems*, vol.18, no.3, pp.463–477, 2016.

[10] Ghosh, D., A quasi-Newton method with rank-two update to solve interval optimization problems, *International Journal of Applied and Computational Mathematics*, 2016, doi:10.1007/s40819-016-0202-7.

[11] Ghosh, D., A Newton method for capturing efficient solutions of interval optimization problems, *OPSEARCH*, vol.53, no.3, pp.648–665, 2016.

[12] Ghosh, D., Newton method to obtain efficient solutions of the optimization problems with interval-valued objective functions, *Journal of Applied Mathematics and Computing*, 2016, doi:10.1007/s12190-016-0990-2.

[13] Griva, I., Nash, S.G., and A. Sofer, *Linear and Nonlinear Optimization*, 2nd Edition, SIAM, 2009.

[14] Hosseinzadeh Lotfi, F., Allahviranloo, T., Alimardani Jondabeh, M., and L. Alizadeh, Solving a full fuzzy linear programming using lexicography method and fuzzy approximate solution, *Applied Mathematical Modelling*, vol.33, pp.3151–3156, 2009.

Table 5: Convergence of the quasi-Newton Algorithm 1 for Example 5.3

k	$x^{(k)}$	t_k	$\varphi(x^{(k)})$	$\nabla\varphi(x^{(k)})$	$d(x^{(k)}) = -(B(x^{(k)}))^{-1}\nabla\varphi(x^{(k)})$	$\|x^{(k+1)} - x^{(k)}\|$
0	$(1, 1)$	0.1250	2	$\begin{pmatrix} -6 \\ 4 \end{pmatrix}$	$(6, -4)$	0.9014
1	$(1.7500, 0.5000)$	1	0.5664	$\begin{pmatrix} 1.4375 \\ -3.0000 \end{pmatrix}$	$(0.6324, 1.0402)$	1.2174
2	$(2.3824, 1.5402)$	1	0.5087	$\begin{pmatrix} -1.1724 \\ 2.7923 \end{pmatrix}$	$(-0.3424, -0.5278)$	0.6291
3	$(2.0400, 1.0125)$	1	$10^{-4} \times 2.2772$	$\begin{pmatrix} 0.0303 \\ -0.0600 \end{pmatrix}$	$(0.0050, 0.0096)$	0.0108
4	$(2.0450, 1.0221)$	1	$10^{-6} \times 4.8241$	$\begin{pmatrix} 0.0021 \\ -0.0034 \end{pmatrix}$	$(10^{-3} \times 0.1405, 10^{-3} \times 0.4629)$	$10^{-4} \times 4.8376$
5	$(2.0451, 1.0225)$		$10^{-6} \times 4.1562$	$\begin{pmatrix} 10^{-3} \times 0.4987 \\ -10^{-3} \times 0.2616 \end{pmatrix}$	$(-10^{-3} \times 0.1378, -10^{-3} \times 0.0525)$	

Table 6: Convergence of the quasi-Newton Algorithm 1 for Example 5.4

k	$(x^{(k)})^T$	t_k	$\varphi(x^{(k)})$	$\nabla\varphi(x^{(k)})$	$d(x^{(k)}) = -(B(x^{(k)}))^{-1}\nabla\varphi(x^{(k)})$	$\|x^{(k+1)} - x^{(k)}\|$
0	$\begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$	0.0625	20.0500	$\begin{pmatrix} 12.0500 \\ 20.0500 \\ 4.0000 \end{pmatrix}$	$(-12.0500, -20.0500, -4.0000)$	1.4832
1	$\begin{pmatrix} 0.2469 \\ -0.2531 \\ 1.7500 \end{pmatrix}$	0.1250	4.0720	$\begin{pmatrix} 2.9748 \\ -5.0752 \\ 3.5000 \end{pmatrix}$	$(-4.4318, 1.8049, -3.7716)$	0.7616
2	$\begin{pmatrix} -0.3071 \\ -0.0275 \\ 1.2786 \end{pmatrix}$	0.5000	2.2105	$\begin{pmatrix} -3.7005 \\ -0.5515 \\ 2.5571 \end{pmatrix}$	$(0.6660, -0.0032, -3.9670)$	2.0112
3	$\begin{pmatrix} 0.0259 \\ -0.0291 \\ -0.7049 \end{pmatrix}$	1	0.5095	$\begin{pmatrix} 0.3123 \\ -0.5836 \\ -1.4099 \end{pmatrix}$	$(-0.0247, 0.0298, 0.7061)$	0.7072
4	$\begin{pmatrix} 0.0012 \\ 0.0006 \\ 0.0012 \end{pmatrix}$	1	$10^{-5} \times 1.3921$	$\begin{pmatrix} 0.0142 \\ 0.0129 \\ 0.0024 \end{pmatrix}$	$(-0.0012, -0.0007, -0.0012)$	0.0018
5	$\begin{pmatrix} -10^{-4} \times 0.1795 \\ -10^{-4} \times 0.0919 \\ 10^{-4} \times 0.0009 \end{pmatrix}$	1	$10^{-9} \times 2.7872$	$\begin{pmatrix} -10^{-3} \times 0.2163 \\ -10^{-3} \times 0.1842 \\ 10^{-3} \times 0.0002 \end{pmatrix}$	$(10^{-4} \times 0.1795, 10^{-4} \times 0.0919, -10^{-4} \times 0.0012)$	$10^{-5} \times 2.0167$
6	$\begin{pmatrix} 10^{-7} \times 0.0274 \\ 10^{-7} \times 0.0132 \\ -10^{-7} \times 0.2396 \end{pmatrix}$		$10^{-16} \times 6.3686$	$\begin{pmatrix} 10^{-7} \times 0.3302 \\ 10^{-7} \times 0.2652 \\ -10^{-7} \times 0.4792 \end{pmatrix}$	$(-10^{-7} \times 0.0274, -10^{-7} \times 0.0132, 10^{-7} \times 0.2399)$	

[15] Karun Kumar, M., and V.N. Sastry, A new algorithm to compute pareto-optimal paths in a multi objective fuzzy weighted network, *OPSEARCH*, vol.50, no.3, pp.297–318, 2013.

[16] Liu, B., *Uncertainty Theory*, Springer-Verlag, Berlin, 2015.

[17] Lodwick, W.A., and J. Kacprzyk (Eds.), *Fuzzy Optimization: Recent Advances and Applications*, Springer-Verlag, Berlin, 2010.

[18] Mahdavi-Amiri, N., and S.H. Nasseri, Duality in fuzzy number linear programming by use of a certain linear ranking function, *Applied Mathematics and Computation*, vol.180, pp.206–216, 2006.

[19] Maleki, H.R., Ranking functions and their applications to fuzzy linear programming, *Far East Journal Mathematics Sciences*, vol.4, pp.283–301, 2002.

[20] Mottaghi, A., Ezzati, R., and E. Khorram, A new method for solving fuzzy linear programming problems based on the fuzzy linear complementary problem (FLCP), *International Journal of Fuzzy Systems*, vol.17, no.2, pp.236–245, 2015.

[21] Nocedal, J., and S. Wright, *Numerical Optimization*, Springer Science and Business Media, 2006.

[22] Pirzada, U.M., and V.D. Pathak, Newton method for solving the multi-variable fuzzy optimization problem, *Journal of Optimization Theory and Applications*, vol.156, no.3, pp.867–881, 2013.

[23] Qu, S., Goh, M., and F.T.S. Chan, Quasi-Newton methods for solving multiobjective optimization, *Operations Research Letters*, vol.39, pp.397–399, 2011.

[24] Stefanini, L., A generalization of Hukuhara difference and division for interval and fuzzy arithmetic, *Fuzzy Sets and Systems*, vol.161, no.11, pp.1564–1584, 2010.

[25] Stefanini, L., and B. Barnabas, Generalized Hukuhara differentiability of interval-valued functions and interval differential equations, *Nonlinear Analysis: Theory, Methods & Applications*, vol.71, no.3, pp.1311–1328, 2009.

[26] Tanaka, H., Okuda, T., and K. Asai, On fuzzy mathematical programming, *Journal of Cybernetics*, vol.3, pp.37–46, 1974.