Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Robust Optimization-Based Quasi-Newton Method for Uncertain Multiobjective Optimization Problems

Shubham Kumar
Pandit Dwarka Prasad Mishra Indian Institute of Information Technology, Design and Manufacturing (PDPM IIITDM) Jabalpur
Nihar Kumar Mahato
nihar@iiitdmj.ac.in

Pandit Dwarka Prasad Mishra Indian Institute of Information Technology, Design and Manufacturing (PDPM IIITDM) Jabalpur
Md Abu T Ansary
Indian Institute of Technology Jodhpur
Debdas Ghosh
Indian Institute of Technology Varanasi

Research Article

Keywords: Multiobjective optimization problem, Uncertainty, Robust optimization, Robust efficiency, Quasi Newton's method, Line search techniques.

Posted Date: April 22nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9344639/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: No competing interests reported.

A Robust Optimization-Based Quasi-Newton Method for Uncertain Multiobjective Optimization Problems

Shubham Kumar[*], Nihar Kumar Mahato [†], Md Abu T Ansary [‡], Debdas Ghosh [§]

Abstract

Uncertain multiobjective optimization problems arise in various real-world scenarios where objectives are affected by uncertainty. To address this, we propose a quasi-Newton method to solve the robust counterpart of an uncertain multiobjective optimization problem under an arbitrary finite uncertainty set. The robust counterpart is formulated as a nonsmooth deterministic multiobjective optimization problem, where we construct a sub-problem using Hessian approximation to determine a descent direction. An Armijo-type inexact line search technique is introduced to compute an appropriate step length, and a modified BFGS formula ensures positive definiteness of the Hessian matrix at each iteration. By incorporating these components, we develop a quasi-Newton descent algorithm for the robust counterpart and establish its convergence under standard assumptions, proving a superlinear convergence rate. Numerical experiments validate the effectiveness of our method by comparing it with the weighted sum method through a performance profile, demonstrating its efficiency and robustness in solving uncertain multiobjective problems.

Keywords: Multiobjective optimization problem, Uncertainty, Robust optimization, Robust efficiency, Quasi Newton's method, Line search techniques.

1 Introduction

Before introducing an uncertain multiobjective optimization problem, we start with a multiobjective optimization problem. In multiobjective optimization problem, we want to solve more than one objective functions simultaneously. But, there is no single point which minimize all objective function at once, so the concept of optimality is replaced by Pareto optimality for multiobjective optimization problem. Multiobjective optimization problem have many application in field of science, for that see Bhaskar et al. (2000); Stewart et al. (2008) and references therein. Mathematically, a deterministic multiobjective optimization problem is given as follows

$$P: \quad \min_{x \in D \subset \mathbb{R}^n} \zeta(x) \tag{1.1}$$

where $\zeta : \mathbb{R}^n \to \mathbb{R}^m$ such that $\zeta(x) = (\zeta_1(x), \zeta_2(x), \ldots, \zeta_m(x))$ and $\zeta_j : \mathbb{R}^n \to \mathbb{R}$, $j \in \Lambda = \{1, 2, \ldots, m\}$, $D \subset \mathbb{R}^n$ is a feasible set. In particular, if $D = \mathbb{R}^n$ then the above deterministic multiobjective optimization problem is called unconstrained multiobjective optimization problem.

Various solution strategies for multiobjective optimization problems have been proposed in the literature Ansary and Panda (2015, 2019, 2020, 2021); Drummond and Svaiter (2005); Fliege et al. (2009); Fliege and Svaiter (2000); Fukuda and Drummond (2014); Gass and Saaty (1955); Lai et al. (2020); Miettinen (1999); Povalej (2014). Several iterative methods originally designed for scalar optimization problems have been extended and analyzed to address multiobjective optimization problems. The field of iterative methods for multiobjective optimization problems was initially introduced by Fliege and

[*]IIITDM Jabalpur, India, E-mail: kumarshubham3394@gmail.com
[†]IIITDM Jabalpur, India, E-mail: nihar@iiitdmj.ac.in, Corresponding Author
[‡]IIT Jodhpur, India, E-mail: md.abutalha2009@gmail.com
[§]IIT(BHU), India, E-mail: debdas.mat@iitbhu.ac.in

Svaiter (2000). Since then, several authors have expanded upon this area, including the development of Newton's method (Fliege et al. (2009)), quasi Newton method (Ansary and Panda (2015); Lai et al. (2020); Mahdavi-Amiri and Salehi Sadaghiani (2020); Morovati et al. (2018); Povalej (2014); Qu et al. (2011)), conjugate gradient method (Gonçalves and Prudente (2020); Lucambio Pérez and Prudente (2018)), projected gradient method (Cruz et al. (2011); Drummond and Iusem (2004); Fazzio and Schuverdt (2019); Fukuda and Drummond (2011); Fukuda and Graña Drummond (2013)), and proximal gradient method (Bonnel et al. (2005); Ceng et al. (2010)). Convergence properties are a common characteristic of these methods. Notably, these approaches do not require transforming the problem into a parameterized form, distinguishing them from scalarization techniques given by Eichfelder and heuristic approaches given by Laumanns et al. (2002).

An unconstrained uncertain multiobjective optimization problem with a parameter uncertainty (perturbation) set consists of a set of deterministic multiobjective optimization problems, which can be expressed as follows:

$$P(U) = \{P(\xi) : \xi \in U\}, \tag{1.2}$$

where for any fix $\xi \in U$,

$$P(\xi) : \min_{x \in \mathbb{R}^n} \zeta(x, \xi) = (\zeta_1(x, \xi), \zeta_2(x, \xi), ..., \zeta_m(x, \xi)), \quad \xi \in U \subset \mathbb{R}^k$$

$\zeta : \mathbb{R}^n \times U \to \mathbb{R}^m$ and U is uncertainity set.

The concept of robust optimization, originally developed for uncertain single objective optimization problems by Ben-Tal et al. (2009); Soyster (1973), has been extended and analyzed for uncertain multiobjective optimization problems Ehrgott et al. (2014); Kuroiwa and Lee (2012). Kuroiwa and Lee (2012), introduced the concept of minimax robustness for multiobjective optimization problems, replacing the objective vector of the uncertain problem with a vector that incorporates worst-case scenarios for each objective component. Further, Ehrgott et al. (2014) generalized the concept of minimax robustness for multiobjective optimization in a comprehensive manner. This robustness concept, also known as robust optimization, enables the transformation of an uncertain multiobjective optimization problem into a deterministic multiobjective optimization problem, referred to as the robust counterpart. There are several type of robust counterpart used to change uncertain multiobjective optimization problem into deterministic multiobjective optimization problem. By Ehrgott et al. (2014), the minimax approach and the objective-wise worst-case cost type robust counterpart are employed to transform the uncertain multiobjective optimization problem into a deterministic one, which can then be solved using scalarization techniques such as the weighted sum method and the ϵ-constraint method. For further details of the weighted sum method and the ϵ-constraint method see Ehrgott (2005).

Scalarization techniques have drawbacks for uncertain multiobjective optimization problems (see Ehrgott et al. (2014)). These approaches require the pre-specification of weights, constraints, or function importance, which is not known in advance. To address these difficulties, Kumar et al. (2023) proposed solving uncertain multiobjective optimization problems using Newton's method by using objective-wise worst-case cost type robust counterpart (OWC). Generally, OWC is a non-smooth deterministic multiobjective optimization problem and cannot be handled using smooth multiobjective optimization techniques which is given in the literature Ansary and Panda (2015, 2019, 2020, 2021); Drummond and Svaiter (2005); Fliege et al. (2009); Fliege and Svaiter (2000); Fukuda and Drummond (2014); Gass and Saaty (1955); Lai et al. (2020); Povalej (2014). Kumar et al. (2023) extended Fliege et al. (2009) Newton's method for deterministic multiobjective optimization problems to uncertain multiobjective optimization problems using OWC. However, Newton's method is computationally expensive. To overcome this issue, our paper aims to solve uncertain multiobjective optimization problems using OWC and the quasi Newton method.

Quasi Newton algorithms, initially introduced in 1959 by Davidon and later popularized by Fletcher and Powell (1963), are a class of methods for solving single-objective optimization problems. These algorithms are known for avoiding second-order derivative computations and performing well in practice. In the quasi Newton method, the search direction is determined based on a quadratic model of the objective function, where the true Hessian is replaced by an approximation that is updated after each

iteration. Quasi Newton algorithms have also been extended for multiobjective optimization problems Qu et al. (2011). BFGS (Broyden-Fletcher-Goldfarb-Shanno) is the most effective quasi Newton update formula for single-objective optimization problems and has been extended for smooth multiobjective optimization problems. Under suitable assumptions, it has been proven that the quasi Newton algorithm with BFGS update and appropriate line search is globally convergent for strongly convex functions Qu et al. (2011).

Steepest descent method Kumar et al. (2024) and Newton's method Kumar et al. (2023) have also been proposed for uncertain multiobjective optimization problems via a robust optimization approach. The steepest descent method Kumar et al. (2024), which uses only the first derivatives (gradients) to determine a search direction, is not always the most efficient. Utilizing higher derivatives can yield algorithms that outperform the steepest descent method. Newton's method Kumar et al. (2023), for instance, employs both first and second derivatives and typically performs better than steepest descent, especially when the starting point is close to the local efficient solution. In Newton's method, the search direction is computed based on a quadratic model of the objective function, where the Hessian of each objective function over the element of uncertainty set is positive definite at each iteration. It is also presented that Newton's method has a quadratic rate of convergence. However, for a general nonlinear objective function, convergence to a solution cannot be guaranteed from an arbitrary initial point. If the initial point is not sufficiently close to the solution, the algorithm may not maintain the descent property. A computational drawback of Newton's method Kumar et al. (2023) is the requirement to evaluate the inverse of the Hessian at each iteration. However, it is not an easy task to compute the Hessian matrix at every step of the method given in Kumar et al. (2023). Therefore, we present a quasi-Newton method for uncertain multiobjective optimization problems, in which a search direction is found by using the approximation of the Hessian matrix of each objective function over the element of uncertainty set at every iteration. The approximation of the Hessian matrix does not make the calculation easier only, but it makes the algorithm faster as well.

To our knowledge, there is currently no existing quasi Newton method for the robust counterpart of uncertain multiobjective optimization problems. Therefore, in this study, we have extended the idea proposed by Povalej (2014) to handle the robust counterpart of uncertain multiobjective optimization problems, which overcomes the difficulties of Newton's method for the robust counterpart of uncertain multiobjective optimization problems given by Kumar et al. (2023).

In this paper, we employ the quasi Newton method to solve the $P(U)$ under consideration uncertainty set U is finite by using objective-wise worst-case (OWC). The OWC problem is generally a non-smooth multiobjective optimization problem, which means that the BFGS update formula designed for smooth multiobjective optimization problems is not applicable. To address this issue, we modify the BFGS update formula with Armijo type inexact line search technique and incorporate it into the quasi Newton algorithm for the OWC problem. Furthermore, we demonstrate the convergence of the quasi Newton algorithm to the Pareto optimal point of the OWC problem with a superlinear rate. Importantly, the Pareto optimal solution for the OWC problem also serves as the robust Pareto optimal solution for $P(U)$.

The paper is structured as follows: Section 2 provides important results, basic definitions, and theorems relevant to our problem. In subsection 3.1, we present the solution to the quasi Newton descent direction finding subproblem. Subsequently, in subsection 3.2, we introduce an Armijo-type line search rule to determine a suitable step size, ensuring a decrease in the function value along the quasi Newton descent direction. Subsection 3.3 presents a modified BFGS-type formula and the quasi Newton's algorithm for RC_1^*. By utilizing this algorithm, we generate a sequence, and its convergence to a critical point is proven in subsection 3.4. In subsection 3.5, we establish that under certain assumptions, the sequence generated by the quasi Newton algorithm converges to a Pareto optimum with a superlinear rate. Finally, in subsection 3.6, we numerically verify the quasi Newton's algorithm for RC_1^* using appropriate examples. Section 4 concludes the paper with remarks on the algorithm.

2 Preliminaries

Let's introduce some notations. $\mathbb{R}$: the set of real numbers, $\mathbb{R}_{\geq}$: the set of non-negative real numbers, $\mathbb{R}_{>}$: the set of positive real numbers, $\mathbb{R}_{>} = \{x \in \mathbb{R} : x > 0\}$, $\mathbb{R}_{\geq} = \{x \in \mathbb{R} : x \geq 0\}$, $\mathbb{R}^n = \mathbb{R} \times ... \times \mathbb{R}$ (n times), $\mathbb{R}^n_{\geq} = \mathbb{R}_{\geq} \times ... \times \mathbb{R}_{\geq}$ (n times), $\mathbb{R}^n_{>} = \mathbb{R}_{>} \times ... \times \mathbb{R}_{>}$ (n times). In the context of $p, q \in \mathbb{R}^n$, the notation $p \leq q$ means that each component of p is less than or equal to the corresponding component of q ($p_i \leq q_i$ for all $i = 1, 2, ..., n$). Similarly, for any $p, q \in \mathbb{R}^n$: $p \geq q$ if and only if $p - q \in \mathbb{R}^n_{\geq}$, which is equivalent to $p_i - q_i \geq 0$ for each i, $p > q$ if and only if $p - q \in \mathbb{R}^n_{>}$, which is equivalent to $p_i - q_i > 0$ for each i. Lastly, we denote the indexed sets as $\bar{\Lambda} = \{1, 2, ..., p\}$ and $\Lambda = \{1, 2, ..., m\}$, which contain p and m elements, respectively.

Now we define Pareto optimal solution (efficient solution) for problem P defined in (1.1). A feasible point $x^* \in D$ is said to be an efficient (weakly efficient) solution for P if there is no other $x \in D$ such that $\zeta(x) \leq \zeta(x^*)$ & $\zeta(x) \neq \zeta(x^*)$ $\big(\zeta(x) < \zeta(x^*)\big)$. If x^* is an efficient (weakly efficient) solution, then $\zeta(x^*)$ is called non dominated (weakly non dominated) point, and the set of efficient solution and non dominated point are called efficient set and non dominated set respectively. A feasible point x^* is said to be locally efficient (locally weakly efficient) solution if D is restricted to a neighborhood of x^* as a efficient (weakly efficient) solution i.e., if there is a neighborhood $N \subset D$ of x^* such that x^* is an efficient solution (weakly efficient solution) on N. In case, if $\zeta(x)$ is $\mathbb{R}^m-$convex (i.e., each component of $\zeta(x)$ is convex) then locally efficient solution will be globally efficient solution. Göpfert and Nehse (1990), Luc (1988) given the necessary condition for locally weakly efficient solution for smooth deterministic multiobjective optimization problem (1.1). The necessary condition for a given point x^* to be a locally weakly efficient solution for (1.1) is $I(J(\zeta(x^*))) \cap (-\mathbb{R}^m_{>}) = \emptyset$, where $I(J(\zeta(x^*))$ denotes the image set of Jacobian of ζ at x^*. Fliege and Svaiter (2000) given the definition of critical point for smooth deterministic multiobjective optimization problem (1.1). A point x^* is said to be locally critical point for smooth deterministic multiobjective optimization problem (1.1) if $I(J(\zeta(x^*))) \cap (-\mathbb{R}^m_{>}) = \emptyset$. From the definition of critical point it is clear that if x^* is not a critical point then $\exists$ a $d \in \mathbb{R}^n$ such that $\nabla\zeta_j(x^*)^T d < 0$ for all $j \in \Lambda$.

In this paper, our focus is on finding the solution to $P(U)$ (as defined in equation (1.2)), where the uncertainty set U is a finite nonempty set. To achieve this, we transform the problem into a deterministic multiobjective optimization problem using the worst-case cost type robust counterpart. We then solve this transformed problem using the quasi Newton method. In this case, we assume that U is a finite uncertainty set with p elements. Thus, the uncertain multiobjective optimization problem $P(U)$ with a finite uncertainty set U can be reformulated as follows:

$$P(U) = \{P(\xi_i) : i \in \bar{\Lambda}\},\ U = \{\xi_i : i \in \bar{\Lambda}\}, \tag{2.1}$$

where for any fix $\xi_i \in U$,

$$P(\xi_i) : \min_{x \in \mathbb{R}^n} \zeta(x, \xi_i) = (\zeta_1(x, \xi_i), \zeta_2(x, \xi_i), ..., \zeta_m(x, \xi_i)),\quad \xi_i \in U \subset \mathbb{R}^k,$$

and its objective-wise worst-case cost type robust counterpart(OWC), RC_1^* is given as follows:

$$RC_1^* :\quad \min_{x \in \mathbb{R}^n} F(x),\ \text{where } F(x) = (F_1(x), F_2(x), \ldots, F_m(x)),\ F_j(x) = \max_{i \in \bar{\Lambda}} \zeta_j(x, \xi_i). \tag{2.2}$$

Additionally, let's define $I_j(x)$ as the set of all points in $\bar{\Lambda}$ where $\zeta_j(x, \xi_i)$ achieves its maximum value. In other words, $I_j(x)$ is given by $I_j(x) = \{i \in \bar{\Lambda} : \zeta_j(x, \xi_i) = F_j(x)\}$. Further details can be found in Sun and Yuan (2006).

For the problem $P(U)$, our objective is to find a robust efficient solution or a robust weakly efficient solution, which can be defined as follows:

Definition 2.1. Ehrgott et al. (2014) A feasible point $x^* \in \mathbb{R}^n$ is said to be robust efficient (robust weakly efficient) if there is no $x \in \mathbb{R}^n \setminus \{x^*\}$ such that $\zeta(x; U) \subseteq \zeta(x^*; U) - \mathbb{R}^k_{\geq}$ $\big(\zeta(x; U) \subseteq \zeta(x^*; U) - \mathbb{R}^k_{>}\big)$, where $\zeta(x; U) = \{\zeta(x, \xi) : \xi \in U\}$ set of all possible values of objective function.

Note that RC_1^* represents a deterministic non-smooth multiobjective optimization problem. Solving this problem will yield a solution in the form of Pareto optimal or weak Pareto optimal solutions. To establish the connection between RC_1^* and $P(U)$, Ehrgott et al. presented a theorem that demonstrates that the solution of RC_1^* coincides with the solution of $P(U)$.

Theorem 2.1. Let $P(U)$ be an uncertain multiobjective optimization problem with finite uncertainty set which is defined in (2.1) and RC_1^* be the robust counterpart of $P(U)$. Then:

(a) If $x^* \in \mathbb{R}^n$ is an efficient solution to RC_1^*. Then x^* is robust efficient solution for $P(U)$.

(b) If $\max_{i\in\bar{\Lambda}} \zeta_j(x, \xi_i)$ exist for all $j \in \Lambda$ and all $x \in \mathbb{R}^n$ and x^* is a weakly efficient solution to RC_1^*. Then x^* is robust weakly efficient solution for $P(U)$.

Proof. (a) We can prove the result by contradiction. On contary, we assume that x^* is not a robust efficient solution for $P(U)$ then by definition of robust efficient solution there exists $\bar{x} \in \mathbb{R}^n$ such that

$$\begin{aligned}
&\zeta(\bar{x}; U) \subseteq \zeta(x^*; U) - \mathbb{R}^k_{\geq}, \\
&\implies \zeta(\bar{x}; U) - \mathbb{R}^k_{\geq} \subseteq \zeta(x^*; U) - \mathbb{R}^k_{\geq} \\
&\implies \text{ for all } \ \xi_i \in U \text{ there exist a } \ \xi_l \in U \text{ such that} \\
&\implies \zeta(\bar{x}, \xi_i) \leq \zeta(x^*, \xi_l) \\
&\implies \max_{\xi_i \in U} \zeta_j(\bar{x}, \xi_i) \leq \max_{\xi_l \in U} \zeta_j(x^*, \xi_l) \text{ for all } j \in \Lambda.
\end{aligned}$$

By the last expression, it is clear that $F(x^*)$ is equal to or dominated by $F(\bar{x})$ in RC_1^*, thus x^* is not an efficient solution for RC_1^*, which contradicts the fact that x^* is an efficient solution for RC_1^*. Hence, x^* is a robust efficient solution for $P(U)$.
(b) Similar to item (a), item (b) can be prove by the contradiction. On contrary, we assume that x^* is not a robust weak efficient solution for $P(U)$ then by definition of robust efficient solution there exists $\bar{x} \in \mathbb{R}^n$ such that

$$\begin{aligned}
&\zeta(\bar{x}; U) \subseteq \zeta(x^*; U) - \mathbb{R}^k_{>}, \\
&\implies \zeta(\bar{x}; U) - \mathbb{R}^k_{\geq} \subseteq \zeta(x^*; U) - \mathbb{R}^k_{>} \\
&\implies \text{ for all } \ \xi_i \in U \text{ there exist a } \ \xi_l \in U \text{ such that} \\
&\implies \zeta(\bar{x}, \xi_i) < \zeta(x^*, \xi_l) \\
&\implies \max_{\xi_i \in U} \zeta_j(\bar{x}, \xi_i) < \max_{\xi_l \in U} \zeta_j(x^*, \xi_l) \text{ for all } j \in \Lambda.
\end{aligned}$$

By the last expression, it is clear that $F(x^*)$ is strictly dominated by $F(\bar{x})$ in RC_1^*, thus x^* is not a weak efficient solution for RC_1^*, which contradicts the fact that x^* is a weak efficient solution for RC_1^*. Hence, x^* is a robust weak efficient solution for $P(U)$. □

According to Theorem 2.1, we can observe that solution of RC_1^* is the solution $P(U)$. Therefore, we employ the quasi Newton algorithm to solve RC_1^*. To do so, we first establish the necessary condition for efficiency or Pareto optimality. For a given point $x^* \in \mathbb{R}^n$ to be a locally weakly efficient solution of RC_1^*, it is required that $I(Conv\{\cup_{j\in\Lambda}\partial F_j(x^*)\}) \cap (-\mathbb{R}^n_{>}) = \emptyset$. Motivated by this notion, we introduce the concept of a critical point for RC_1^*.

Definition 2.2. Kumar et al. (2023) A point $x^* \in \mathbb{R}^n$ is said to be critical point for RC_1^* if

$$I(Conv\{\cup_{j\in\Lambda}\partial F_j(x^*)\}) \cap (-\mathbb{R}^n_{>}) = \emptyset.$$

Note that if x^* is a critical point for RC_1^* then there is no $v \in \mathbb{R}^n$ such that $\nabla\zeta_j(x^*, \xi_i)^T v < 0$, for all $i \in I_j(x)$, $j \in \Lambda$, where $I_j(x) = \{i \in \bar{\Lambda} : \zeta_j(x^*, \xi_i) = F_j(x^*)\}$.

Definition 2.3. Nocedal and S.T. (2006) Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function. Then a vector v is said to be descent direction of f at x if and only if there exists $\delta > 0$ such that $f(x + \alpha v) < f(x)$, $\forall \alpha \in (0, \delta)$. If f is continuously differentiable then a vector $v \in \mathbb{R}^n$ is said to be a descent direction for f at x if $\nabla f(x)^T v < 0$.

Definition 2.4. Kumar et al. (2023) In case of RC_1^*, a vector v is said to be descent direction for F at x if

$$\nabla \zeta_j(x, \xi_i)^T v < 0, \ \forall j \in \Lambda$$

and $i \in I_j(x)$. Also if v is descent direction for $F(x)$ at x then there exists $\epsilon > 0$ such that

$$F_j(x + \alpha v) < F_j(x) \ \forall \ j \in \Lambda \text{ and } \alpha \in (0, \epsilon].$$

Theorem 2.2. *Dhara and Dutta (2011) Let $F_j : \mathbb{R}^n \to \mathbb{R}$ be a function such that $F_j(x) = \max_{i \in \bar{\Lambda}} \zeta_j(x, \xi_i)$. Then:*

(i) The directional derivative of F_j at x in the direction v is given by $F_j'(x, v) = \max\limits_{i \in I_j(x)} \nabla \zeta_j(x, \xi_i)^T v$, where $I_j(x) = \{i \in \bar{\Lambda} : \zeta_j(x, \xi_i) = F_j(x)\}$.

(ii) The subdifferential of F_j is

$$\partial F_j(x) = Conv\Big(\bigcup_{i \in I_j(x)} \zeta_j(x, \xi_i) \Big).$$

Also $x^* = arg \min\limits_{x \in \mathbb{R}^n} F_j(x)$ if and only if $0 \in \partial F_j(x^*)$).

Definition 2.5. Miettinen (1999) Let $f : S \subset \mathbb{R}^n \to \mathbb{R}^m$ be a continuously differentiable function such that $f(x) = (f_1(x), f_2(x), \ldots, f_m(x))$ then f is $\mathbb{R}^m -$ strongly convex if and only if there exists $\omega > 0$ such that $\omega \|u\|^2 \leq u^T \nabla^2 f_j(x) u, \forall x \in S, u \in \mathbb{R}^n$ and $j \in \Lambda$.

To prove the necessary and sufficient condition for Pareto optimality for RC_1^* we give lemma 2.1.

Lemma 2.1. *If x^* is critical point for F if and only if $0 \in Conv\{\bigcup\limits_{j \in \Lambda} \partial F_j(x^*)\}$.*

Proof. Since x^* is critical point for F then there must exists $d \in \bigcup_{j \in \Lambda} \partial F_j(x^*)$ such that

$$v^T d \geq 0, \ \forall v \in \mathbb{R}^n. \tag{2.3}$$

On contrary, if we assume $0 \notin Conv\{\bigcup\limits_{j \in \Lambda} \partial F_j(x^*)\}$. Since, $Conv\{\bigcup\limits_{j \in \Lambda} \partial F_j(x^*)\}$ and $\{0\}$ are closed and convex sets then with the help of separation theorem, there exists $v \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that $v^T 0 \geq b$ and $v^T d < b \ \forall d \in Conv\{\bigcup\limits_{j \in \Lambda} \partial F_j(x^*)\}$. Jointly both inequality contradicts (2.3). Hence, $0 \in Conv\{\bigcup\limits_{j \in \Lambda} \partial F_j(x^*)\}$. Conversely, we have to prove that if $0 \in Conv\{\bigcup\limits_{j \in \Lambda} \partial F_j(x^*)\}$ then x^* is critical point for F. For that we define $F'(x) = \max\limits_{j \in \Lambda} F_j(x) - F_j(x^*)$. Then by item (ii) of Theorem 2.2 we have $\partial F'(x) = Conv\Big(\bigcup\limits_{i \in \Lambda} \partial \zeta_j(x) \Big)$. Therefore, by assumption we get $0 \in Conv\Big(\bigcup\limits_{i \in \Lambda} \partial \zeta_j(x) \Big)$ which implies $x^* = \arg \min\limits_{x \in \mathbb{R}^n} F'(x)$. On contrary, if we assume x^* is not a critical point then by Definition 2.2, there exists $s \in \mathbb{R}^n$ such that $\nabla \zeta_j(x^*, \xi_i)^T s < 0$, for all $i \in I_j(x^*)$, $j \in \Lambda$ i.e., $F_j'(x^*, s) < 0$ for all j. Then there exists some $\eta > 0$ sufficiently small such that $F_j(x^* + \eta s) < F_j(x^*)$ for all j which implies $F'(x^* + \eta s) < 0 = F'(x^*)$ holds for some $x^* + \eta s \in \mathbb{R}^n$. Then there is a contradiction to the fact that $x^* = \arg \min\limits_{x \in \mathbb{R}^n} F'(x)$. Therefore our assumption x^* is not a critical point is wrong and hence x^* is a critical point for F. □

We now prove the necessary and sufficient condition for the Pareto optimality for the robust counterpart RC_1^* using Lemma 2.1.

Theorem 2.3. *If $\zeta_j(x,\xi)$ is continuously differentiable and convex for each $j \in \Lambda$ and $\xi \in U$, then x^* is a weakly efficient solution for RC_1^* if and only if*

$$0 \in conv\left(\cup_{j=1}^{m} \partial F_j(x^*)\right).$$

Proof. Let x^* be a weakly efficient solution for RC_1^*. We have to show $0 \in Conv\{\cup_{j\in\Lambda}\partial F_j(x^*)\}$. Since given function $\zeta_j(x,\xi_i)$ is continuously differentiable and convex for each j and $\xi_i \in U$, then $\zeta_j(x,\xi_i)$ will be locally Lipschitz continuous for all $i \in \bar{\Lambda}$. Then, by Theorem 4.3 in Mäkelä et al. (2014) $0 \in Conv\{\cup_{j\in\Lambda}\partial F_j(x^*)\}$.
Conversely, by assumption $0 \in Conv\{\cup_{j\in\Lambda}\partial F_j(x^*)\}$ it is clear that x^* is critical point. Then, for atleast one j^0, we have $F'_{j^0}(x^*,d) \geq 0,\ \forall\ d \in \mathbb{R}^n$. Now, by using the Definition 2.2, we have

$$\nabla\zeta_{j^0}(x^*,\xi_i)^T d \geq 0,\ \forall\ d \in \mathbb{R}^n,\ i \in I_{j^0}(x^*). \tag{2.4}$$

By convexity of F_j and $\zeta_j(x,\xi_i)$, we get

$$\zeta_{j^0}(x,\xi_i) \geq \zeta_{j^0}(x^*,\xi_i) + \nabla\zeta_{j^0}(x^*,\xi_i)^T(x-x^*),\ \forall\ i \in I_{j^0}(x^*) \text{ and } x,\ x^* \in \mathbb{R}^n.$$

By (2.4), we have

$$\zeta_{j^0}(x,\xi_i) \geq \zeta_{j^0}(x^*,\xi_i),\ \forall\ i \in I_{j^0}(x^*),$$

and therefore

$$F_{j^0}(x) \geq F_{j^0}(x^*),\ \forall x \in \mathbb{R}^n,$$

i.e., x^* is weakly efficient solution. □

BFGS (Broyden, Fletcher, Goldfarb and Shanno) is a most popular quasi Newton method for single objective optimization problem. It is a line search method with a descent direction

$$v^k = -(B^k)^{-1}\nabla f(x^k)$$

where $f : S \to \mathbb{R}$ is a continuously differentiable function, $S \subset \mathbb{R}^n$ and B^k is a positive definite symmetric matrix which is updated at every iteration. Also the new iteration is given by

$$x^{k+1} = x^k + \alpha_k v^k$$

where v^k and α_k are the descent direction and steplenth. Now the BFGS update formula is given as follows

$$B^{k+1} = B^k - \frac{B^k u^k (u^k)^T B^k}{(u^k)^T B^k u^k} + \frac{p^k (p^k)^T}{(u^k)^T p^k} \tag{2.5}$$

where $u^k = x^{k+1} - x^k = \alpha_k v^k$ and $p^k = \nabla f(x^{k+1}) - \nabla f(x^k)$. Also from Nocedal and S.T. (2006), B^{k+1} is positive definite whenever B^k is positive definite and α_k satisfies Wolfe line search rule. For the BFGS method two conditions must be satisfied. The seacant equation $B^{k+1}u^k = p^k$ is the first condition which we obtain by multiplying (2.5) by u^k. The curvature condition $(u^k)^T p^k > 0$ is the second necessary condition. Also if f is strongly convex, then for any two points x^k and x^{k+1} the curvature condition is satisfied. BFGS update formula which is given in (2.5) defined for single objective optimization problem, and Povalej (2014), generalized this idea for unconstrained multiobjective optimization problem of strongly convex objective functions. The BFGS update formula, given by Žiga Povalej is as follows

$$B_j^{k+1} = B_j^k - \frac{B_j^k u^k (u^k)^T B_j^k}{(u^k)^T B_j^k u^k} + \frac{p^k (p^k)^T}{(u^k)^T p^k} \tag{2.6}$$

where $u^k = x^{k+1} - x^k = \alpha_k v^k$ and $p^k = \nabla f_j(x^{k+1}) - \nabla f_j(x^k)$.

In the subsequent section, we develop a quasi-Newton method for RC_1^* defined in 2.2. To write the quasi-Newton algorithm, search direction, stepsize in the search direction, and Hessian approximation update formula are the three main required things.

3 Quasi Newton method for RC_1^*

In this section, we establish the necessary results to formulate the quasi Newton algorithm. The algorithm requires three main components: (1) the quasi Newton descent direction, (2) an appropriate step length along the descent direction, and (3) an updated formula to maintain the positive definiteness of the Hessian approximation at the current iteration point.

To obtain these components, we consider RC_1^* with a function $F : \mathbb{R}^n \to \mathbb{R}^m$ defined as $F(x) = (F_1(x), F_2(x), ..., F_m(x))$, where $F_j(x) = \max\limits_{i \in \bar{\Lambda}} \zeta_j(x, \xi_i),\ j \in \Lambda$. Here, $\zeta_j : \mathbb{R}^n \times U \to \mathbb{R}$ represents a twice continuously differentiable and strictly convex function for each x and $\xi_i \in U$.

Next, we construct a subproblem that will provide the quasi Newton descent direction once solved.

3.1 Subproblem to find the quasi Newton descent direction for RC_1^*

$$\min_{s \in \mathbb{R}^n} \Theta_x(s), \tag{3.1}$$

where $\Theta_x(s) = \max\limits_{j \in \Lambda} \max\limits_{i \in \bar{\Lambda}} \{\zeta_j(x, \xi_i) + \nabla \zeta_j(x, \xi_i)^T s + \frac{1}{2} s^T H_j(x, \xi_i) s - F_j(x)\}$.
In the subproblem (3.1), $H_j(x, \xi_i)$ is the approximation of the Hessian $\nabla^2 \zeta_j(x, \xi_i)$, for each $j \in \Lambda$, $i \in \bar{\Lambda}$. Also, we are assuming that $H_j(x, \xi_i)$ is positive definite for each $j \in \Lambda$, $i \in \bar{\Lambda}$. Then the objective function $\Theta_x(s)$ is strongly convex being a maximum of strongly convex function and hence subproblem (3.1) is well defined and has a unique solution. If we denote optimal solution and optimal value of the subproblem (3.1) by $s(x)$ and $\vartheta(x)$ respectively, then we can write

$$s(x) = arg \min_{s \in \mathbb{R}^n} \Theta_x(s) \tag{3.2}$$

$$\vartheta(x) = \Theta_x(s(x)). \tag{3.3}$$

Theorem 3.1. Let at any fix $x \in \mathbb{R}^n$, $H_j(x, \xi_i)$ be a positive definite approximation of the Hessian $\nabla^2 \zeta_j(x, \xi_i)$, for each $j \in \Lambda$, $i \in \bar{\Lambda}$, then the subproblem (3.1) has a unique solution, namely $s(x)$ which is given by

$$s(x) = -\Big(\sum_{j \in \Lambda} \sum_{i \in \bar{\Lambda}} \lambda_{ij} H_j(x, \xi_i) \Big)^{-1} \sum_{j \in \Lambda} \sum_{i \in \bar{\Lambda}} \lambda_{ij} \nabla \zeta_j(x, \xi_i). \tag{3.4}$$

Proof. We want to find the solution of subproblem (3.1) which is an unconstrained real-valued minimization problem. Equivalently, by replacing $\Theta_x(s)$ as $t \in \mathbb{R}$ the subproblem (3.1) can be written as

$$\begin{aligned} & Q_N(x) : \min_{(t,s) \in \mathbb{R} \times \mathbb{R}^n} t \\ s.t. \quad & \zeta_j(x, \xi_i) + \nabla \zeta_j(x, \xi_i)^T s + \tfrac{1}{2} s^T H_j(x, \xi_i) s - F_j(x) \le t, \quad \forall i \in \bar{\Lambda} \text{ and } j \in \Lambda. \end{aligned}$$

Since the problem $Q_N(x)$ is a convex programming problem then the unique solution of $Q_N(x)$ is given by $(t, s) = (\vartheta(x), s(x))$. Also $Q_N(x)$ has a Slater point then solution of this problem is given by the KKT optimality condition. For that the Lagrangian function is given by

$$L(t, s, \lambda) = t + \sum_{j \in \Lambda} \sum_{i \in \bar{\Lambda}} \lambda_{ij} (\zeta_j(x, \xi_i) + \nabla \zeta_j(x, \xi_i)^T s + \frac{1}{2} s^T H_j(x, \xi_i) s - F_j(x) - t).$$

Then the KKT optimality conditions are

$$\sum_{j\in\Lambda}\sum_{i\in\bar{\Lambda}}\lambda_{ij}=1 \tag{3.5}$$

$$\sum_{j\in\Lambda}\sum_{i\in\bar{\Lambda}}\lambda_{ij}\nabla\zeta_j(x,\xi_i)+\sum_{j\in\Lambda}\sum_{i\in\bar{\Lambda}}\lambda_{ij}H_j(x,\xi_i)^T s=0 \tag{3.6}$$

$$\zeta_j(x,\xi_i)+\nabla\zeta_j(x,\xi_i)^T s+\frac{1}{2}s^T H_j(x,\xi_i)s-F_j(x)-t\leq 0,\quad \forall i\in\bar{\Lambda}\text{ and }j\in\Lambda \tag{3.7}$$

$$\lambda_{ij}(\zeta_j(x,\xi_i)+\nabla\zeta_j(x,\xi_i)^T v+\frac{1}{2}s^T H_j(x,\xi_i)s-F_j(x)-t)=0,\quad \lambda_{ij}\geq 0. \tag{3.8}$$

From (3.6),

$$s(x)=-\Big(\sum_{j\in\Lambda}\sum_{i\in\bar{\Lambda}}\lambda_{ij}H_j(x,\xi_i)\Big)^{-1}\sum_{j\in\Lambda}\sum_{i\in\bar{\Lambda}}\lambda_{ij}\nabla\zeta_j(x,\xi_i). \tag{3.9}$$

Since $H_j(x,\xi_i)$ is positive definite for all $x\in\mathbb{R}^n$, $i\in\bar{\Lambda}$, and $j\in\Lambda$, and $\lambda_{ij}\geq 0$ and x is not a critical point, it follows from equation (3.5) that there must exists at least one $\lambda_{ij}>0$. Consequently, the inverse of $\sum\limits_{j\in\Lambda}\sum\limits_{i\in\bar{\Lambda}}\lambda_{ij}H_j(x,\xi_i)$ exists. Therefore $s(x)$ is well defined. Moreover, the optimal value of the subproblem can be expressed as follows:

$$\vartheta(x)=\max_{j\in\Lambda}\max_{i\in\bar{\Lambda}}\{\zeta_j(x,\xi_i)+\nabla\zeta_j(x,\xi_i)^T s(x)+\frac{1}{2}s(x)^T H_j(x,\xi_i)s(x)-F_j(x)\}. \tag{3.10}$$

□

Theorem 3.2. Let $s(x)$ and $\vartheta(x)$ be the solution and optimal value of the subproblem $Q_N(x)$ respectively. Then the following results will hold:

1. $s(x)$ is bounded on compact subset W of $\mathbb{R}^n$ and $\vartheta(x)\leq 0$.
2. The following conditions are equivalent :
   (a) The point x is not a critical point.
   (b) $\vartheta(x)<0$.
   (c) $s(x)\neq 0$.
   (d) $s(x)$ is a descent direction for F at x for the problem RC_1^* .

In particular x is critical point iff $\vartheta(x)=0$.

Proof. (1) Let W be a compact subset of $\mathbb{R}^n$. Since for each $j\in\Lambda$, $\zeta_j(x,\xi_i)$ is twice continuously differentiable for all $x\in\mathbb{R}^n$ and $\xi_i\in U$, then its first and second order derivative will be bounded on compact set W. Also, $H_j(x,\xi)$ is the Hessian approximation of true Hessian of $\zeta_j(x,\xi_i)$ and $H_j(x,\xi)$ positive definite for each $i\in\bar{\Lambda}$ and $j\in\Lambda$. Hence, $s(x)$ is bounded on compact set W.
Since $t=1$ and $s=\bar{0}=(0,0,\ldots,0)\in\mathbb{R}^n$ lies in the feasible region then we have

$$\vartheta(x)\leq\max_{j\in\Lambda}\{\max_{i\in\bar{\Lambda}}\zeta_j(x,\xi_i)+\max_{i\in\bar{\Lambda}}\nabla\zeta_j(x,\xi_i)^T\bar{0}+\max_{i\in\bar{\Lambda}}\frac{1}{2}\bar{0}^T H_j(x,\xi_i)\bar{0}-\max_{i\in\bar{\Lambda}}F_j(x)\}=0.$$

Hence $\vartheta(x)\leq 0$.
(2) : $(a)\implies(b)$
Assume x is not a critical point. Then there exists $\bar{s}\in\mathbb{R}^n$ such that

$$\nabla\zeta_j(x,\xi_i)^T\bar{s}<0,\ \forall j\in\Lambda\text{ and }i\in I_j(x).$$

Since $\vartheta(x)$ is the optimal value for the subproblem (3.1), then $\forall\delta>0$ we have

$$\vartheta(x) \leq \max_{j\in\Lambda}\{\max_{i\in\bar{\Lambda}} \zeta_j(x,\xi_i) + \max_{i\in\bar{\Lambda}}\{\nabla\zeta_j(x,\xi_i)^T(\delta\bar{s}) + \frac{1}{2}(\delta\bar{s})^T H_j(x,\xi_i)(\delta\bar{s})\} - \max_{i\in\bar{\Lambda}} F_j(x)\}\}.$$

Then we have

$$\vartheta(x) \leq \delta\Big(\max_{j\in\Lambda}\max_{i\in\bar{\Lambda}}\{\nabla\zeta_j(x,\xi_i)^T\bar{s} + \delta\frac{1}{2}\bar{s}^T H_j(x,\xi_i)\bar{s}\}\Big).$$

For small enough $\delta > 0$ right hand side of above inequality will be negative because of $\nabla\zeta_j(x,\xi_i)^T\bar{s} < 0$, $\frac{1}{2}\bar{s}^T H_j(x,\xi_i)\bar{s} > 0$ and also $\vartheta(x) \leq 0$. Thus $\vartheta(x) < 0$.
$(b) \implies (c)$
Since $\vartheta(x)$ is the optimal value of the subproblem (3.1) and from (b) it is strictly negative so we will get $s(x) \neq 0$. Because if $s(x) = 0$ then $\vartheta(x)$ will be zero which is not possible from (b). Hence if $\vartheta(x) < 0$ then $s(x) \neq 0$.
$(c) \implies (d)$
Let $s(x) \neq 0$. Then, $\vartheta(x) \neq 0$. Since $\vartheta(x) \leq 0$ and $s(x) \neq 0$ then $\vartheta(x) < 0$. Thus,

$$\begin{aligned}
&\theta(x) \leq \max_{j\in\Lambda}\{\max_{i\in\bar{\Lambda}} \zeta_j(x,\xi_i) + \max_{i\in\bar{\Lambda}}\{\nabla\zeta_j(x,\xi_i)^T s(x) + \frac{1}{2}s^T(x)H_j(x,\xi_i)s(x)\} - \max_{i\in\bar{\Lambda}} F_j(x)\} < 0\\
&\implies \max_{j\in\Lambda}\max_{i\in\bar{\Lambda}}\{\nabla\zeta_j(x,\xi_i)^T s(x) + \frac{1}{2}s^T(x)H_j(x,\xi_i)s(x)\} < 0\\
&\implies \nabla\zeta_j(x,\xi_i)^T s(x) < 0,\ \forall\ j\in\Lambda \text{ and } i\in\bar{\Lambda}\\
&\implies \nabla\zeta_j(x,\xi_i)^T v(x) < 0,\ \forall\ j\in\Lambda \text{ and } i\in I_j(x)\\
&\implies s(x) \text{ is a descent direction for } F \text{ at } x \text{ for problem } RC_1^*.
\end{aligned}$$

$(d) \implies (a)$
Since $s(x)$ is a descent direction for F at x, then we have

$$\nabla\zeta_j(x,\xi_i)^T s(x) < 0,\ \forall\ j\in\Lambda \text{ and } i\in I_j(x). \text{ Hence } x \text{ is not a critical point.}$$

Also if $\vartheta(x) < 0$ then $s(x) \neq 0$ and also if $s(x) \neq 0$ then $\vartheta(x) < 0$. Thus, we have x is critical point if and only if $\vartheta(x) = 0$. □

From the Theorem 3.2, it is clear that $s(x) \approx 0$ or $\vartheta(x) \approx 0$ can be used as a stopping criterion in the quasi Newton algorithm. Now in the next theorem we will prove $\vartheta(x)$ is continuous for every $x \in \mathbb{R}^n$.

Theorem 3.3. Let $\vartheta : \mathbb{R}^n \to \mathbb{R}$ be a function which is defined in (3.10). Then, $\vartheta(x)$ is continuous.

Proof. To prove continuity of $\vartheta(x)$, it is suffice to prove that $\vartheta(x)$ is continuous in any arbitrary compact subset W of $\mathbb{R}^n$. Since, by Theorem 3.2, $\vartheta(x) \leq 0$ then for every $j \in \Lambda$ and $i \in \bar{\Lambda}$ we have

$$\zeta_j(x,\xi_i) + \nabla\zeta_j(x,\xi_i)^T s(x) + \frac{1}{2}s(x)^T H_j(x,\xi_i)s(x) - F_j(x) \leq 0.$$

Since, at the active indices $\zeta_j(x,\xi_i)$ attains its maximum value i.e., $\zeta_j(x,\xi_i) = F_j(x)$ then from the above equation we get

$$\nabla\zeta_j(x,\xi_i)^T s(x) + \frac{1}{2}s(x)^T H_j(x,\xi_i)s(x) \leq 0,$$

which implies

$$\frac{1}{2}s(x)^T H_j(x,\xi_i)s(x) \leq -\nabla\zeta_j(x,\xi_i)^T s(x),\ \text{for every } j\in\Lambda,\ i\in\bar{\Lambda}. \tag{3.11}$$

For every $j \in \Lambda$ and $i \in \bar{\Lambda}$, approximation of true Hessian of $\zeta_j(x,\xi_i)$ is positive definite for every $x \in \mathbb{R}^n$. Also, W is a compact subset of $\mathbb{R}^n$ then, eigenvalues of approximation of true Hessian of $\zeta_j(x,\xi_i)$ are bounded on C. So, there exists $\nu > 0$ and $\bar{\nu}$, such that

$$\bar{\nu} = \max_{x\in C,\ j\in\Lambda}\max_{i\in\bar{\Lambda}} \|\zeta_j(x,\xi_i)\| \tag{3.12}$$

and

$$\nu\|w\|^2 \leq w^T H_j(x,\xi_i)w, \text{ for every } j \in \Lambda,\ i \in \bar{\Lambda} \text{ and } \forall x \in C, w \in \mathbb{R}^n. \tag{3.13}$$

Now from (3.11), (3.12), (3.13) and Cauchy-Schwartz inequality we have

$$\nu\|s(x)\|^2 \leq \|\nabla\zeta_j(x,\xi_i)\|\|s(x)\| \leq \bar{\nu}\|s(x)\| \ \forall y \in C,\ j \in \Lambda \text{ and } i \in \bar{\Lambda}.$$

Which implies that

$$\|s(x)\| \leq \frac{\bar{\nu}}{\nu}, \quad \forall x \in C$$

i.e., $s(x)$, the Newton's directions are bounded on W.
Now for $x \in C$, $j \in \Lambda$ and $i \in \bar{\Lambda}$ we define

$$E_{x,j,i} : C \to \mathbb{R}$$

$$s.t. \qquad z \to \zeta_j(z,\xi_i) + \nabla\zeta_j(z,\xi_i)s(x) + \frac{1}{2}s(x)^T H_j(z,\xi_i)s(x) - F_j(z).$$

Thus the family $\{E_{x,j,i}\}_{x\in C,\ j\in\Lambda, i\in\bar{\Lambda}}$ is equicontinuous. Therefore, $\{\Delta_x = \max\limits_{j\in\Lambda}\max\limits_{i\in\bar{\Lambda}} E_{x,j,i}\}_{x\in C}$ is also equicontinuous.
Now if we take $\epsilon > 0$ there exists $\delta > 0$ such that $\forall u, w \in C$

$$\|u - w\| < \delta \implies |\Delta_x(u) - \Delta_x(w)| < \epsilon, \forall x \in C.$$

Hence, for $\|u - w\| < \delta$,

$$\begin{aligned}
\vartheta(w) &\leq \zeta_j(w,\xi_i) + \nabla\zeta_j(w,\xi_i)^T s(u) + s(u)^T\nabla^2\zeta_j(w,\xi_i)^T s(u) - F_j(w) \\
&= \Delta_u(w) \\
&\leq \Delta_u(u) + |\Delta_u(w) - \Delta_u(u)| \\
&< \vartheta(u) + \epsilon,
\end{aligned}$$

thus, we get $\vartheta(w) - \vartheta(u) < \epsilon$, also if we interchange u and w then we get $\vartheta(w) - \vartheta(u) > -\epsilon$. Therefore,

$$|\vartheta(u) - \vartheta(w)| < \epsilon, \text{ whenever } \|u - w\| < \delta.$$

Thus, $\vartheta(x)$ is continuous in W. Since, W is any arbitrary compact subset of $\mathbb{R}^n$ and hence, $\vartheta(x)$ is continuous. □

To find the step size in the search direction, we develop an Armijo type inexact line search technique in the following subsection.

3.2 Armijo type inexact line search for quasi Newton method

Theorem 3.4. Assume that x is not critical point of F. Then for any $\beta > 0$ and $\epsilon \in (0, 1]$ there exists an $\alpha \in [0, \epsilon]$ such that

$$F_j(x + \alpha s) \leq F_j(x) + \alpha\beta\vartheta(x),$$

where $s(x)$ and $\vartheta(x)$ are given in the equation (3.9) and (3.10) respectively.

Proof. By (3.2), we have $\vartheta(x) \leq 0$, and

$$\vartheta(x) = \max_{j\in\Lambda}\max_{i\in\bar{\Lambda}}\{\zeta(x,\xi_i) + \nabla\zeta_j(x,\xi_i)^T s(x) + \frac{1}{2}s(x)^T H_j(x,\xi_i)s(x) - F_j(x)\}.$$

Now we consider an auxiliary function $F_j''^*(x,\alpha v)$ to find the step length size which is given by

$$F_j''^*(x,\alpha s) = \max_{i\in\bar{\Lambda}}\{\zeta_j(x,\xi_i) + \alpha\nabla\zeta_j(x,\xi_i)^T s + \frac{1}{2}\alpha^2 s^T H_j(x,\xi_i)^T s\} - F_j(x) - F_j'(x,\alpha s), \tag{3.14}$$

where $\alpha \in [0,\epsilon]$ and $\epsilon < 1$. For every $j \in \Lambda$, (3.14) can be written as

$$F_j''^*(x,\alpha s) + F_j'(x,\alpha s) \le \max_{i\in\bar{\Lambda}}\zeta_j(x,\xi_i) + \max_{i\in\bar{\Lambda}}\{\alpha\nabla\zeta_j(x,\xi_i)^T s + \frac{1}{2}\alpha^2 s^T H_j(x,\xi_i)^T s\} - F_j(x),$$

which implies

$$F_j''^*(x,\alpha s) + F_j'(x,\alpha s) \le \max_{i\in\bar{\Lambda}}\{\alpha\nabla\zeta_j(x,\xi_i)^T s + \frac{1}{2}\alpha^2 s^T H_j(x,\xi_i)^T s\}. \tag{3.15}$$

Since $\zeta_j(x,\xi_i)$, is twice continuously differentiable and convex. Then $\zeta_j(x,\xi_i)$ will be upper uniformly twice continuously differentiable (see Bazaraa and Goode (1982), for the definition of uniformly differentiable function) and convex. Hence there exists $k_i^j > 0$ such that

$$\zeta_j(x+s,\xi_i) \le \zeta_j(x,\xi_i) + \nabla\zeta_j(x,\xi_i)^T s + \tfrac{1}{2}s^T H_j(x,\xi_i)^T s + \tfrac{1}{3!}k_i^j\|s\|^3$$

since above inequality holds for every $s \in \mathbb{R}^n$ then it will be hold for αs. Therefore,

$$\begin{aligned}
\zeta_j(x+\alpha s,\xi_i) &\le \zeta_j(x,\xi_i) + \alpha\nabla\zeta_j(x,\xi_i)^T s + \frac{1}{2}\alpha^2 s^T H_j(x,\xi_i)^T s + \frac{1}{3!}k_i^j\|\alpha s\|^3 \\
&\le \max_{i\in\bar{\Lambda}}\{\zeta_j(x,\xi_i) + \alpha\nabla\zeta_j(x,\xi_i)^T s + \frac{1}{2}\alpha^2 s^T H_j(x,\xi_i)^T s + \frac{1}{3!}k_i^j\|\alpha s\|^3\} \\
&\le \max_{i\in\bar{\Lambda}}\{\zeta_j(x,\xi_i) + \alpha\nabla\zeta_j(x,\xi_i)^T s + \frac{1}{2}\alpha^2 s^T H_j(x,\xi_i)^T s\} + \frac{1}{3!}\alpha^3 K^j\|s\|^3,
\end{aligned}$$

where $K^j = \max_{i\in\bar{\Lambda}} k_i^j$. Now from equation (3.14), we have

$$\zeta_j(x+\alpha s,\xi_i) \le F_j'(x,\alpha s) + F_j(x) + F_j''^*(x,\alpha s) + \frac{1}{3!}\alpha^3 K^j\|s\|^3.$$

Since above inequality hold for each $i \in \bar{\Lambda}$, therefore,

$$\max_{i\in\bar{\Lambda}}\zeta_j(x+\alpha s,\xi_i) \le F_j'(x,\alpha s) + F_j(x) + F_j''^*(x,\alpha s) + \frac{1}{3!}\alpha^3 K^j\|s\|^3. \tag{3.16}$$

Now, by (3.15)

$$\begin{aligned}
F_j''^*(x,\alpha s) + F_j'(x,s) &\le \max_{i\in\bar{\Lambda}}\{\alpha\nabla\zeta_j(x,\xi_i)^T s + \frac{1}{2}\alpha^2 s^T H_j(x,\xi_i)s\} \\
&= \max_{i\in\bar{\Lambda}}\{\alpha\zeta_j(x,\xi_i) + \alpha\nabla\zeta_j(x,\xi_i)^T s + \frac{1}{2}\alpha^2 s^T H_j(x,\xi_i)s - \alpha F_j(x)\} \\
&\le \max_{j\in\Lambda}\max_{i\in\bar{\Lambda}}\{\alpha\zeta_j(x,\xi_i) + \alpha\nabla\zeta_j(x,\xi_i)^T s + \frac{1}{2}\alpha^2 s^T H_j(x,\xi_i)s - \alpha F_j(x)\} \\
&\le \alpha\max_{j\in\Lambda}\max_{i\in\bar{\Lambda}}\{\zeta_j(x) + \nabla\zeta_j(x,\xi_i)^T s + \frac{1}{2}s^T H_j(x,\xi_i)s - F_j(x)\} \\
&+ \frac{\alpha(\alpha-1)}{2}\max_{j\in\Lambda}\max_{i\in\bar{\Lambda}} s^T H_j(x,\xi_i)^T s \\
&\le \alpha\theta(x) + \frac{1}{2}\alpha(\alpha-1)\max_{j\in\Lambda}\max_{i\in\bar{\Lambda}}\{s^T H_j(x,\xi_i)^T s\}.
\end{aligned} \tag{3.17}$$

Since x is not a critical point then by Theorem 3.2, (3.15), (3.16) and (3.17), we get

$$F_j(x+\alpha v) \leq F_j(x) + \alpha\vartheta(x) + \frac{1}{2}\alpha(\alpha-1)\max_{j\in\Lambda}\max_{i\in\bar{\Lambda}}\{s^T H_j(x,\xi_i)s\} + \frac{1}{3!}\alpha^3 K^j\|s\|^3. \tag{3.18}$$

Since for each $i \in \bar{\Lambda}$ and $j \in \Lambda$, $H_j(x,\xi_i)$ is positive definite $\forall x \in \mathbb{R}^n$. Then $s^T H_j(x,\xi_i)^T s > 0$, $0 \neq s \in \mathbb{R}^n$. As $\alpha < 1$, then the third term of R.H.S of (3.18), will be negative. Then (3.18), can be written as

$$F_j(x+\alpha s) \leq F_j(x) + \alpha\vartheta(x) + \alpha^3\frac{1}{3!}K^j\|s\|^3.$$

If $\alpha > 0$ is sufficiently small implies $\alpha^3 \approxeq 0$. Then, we have

$$F_j(x+\alpha s) \leq F_j(x) + \alpha\vartheta(x) \tag{3.19}$$

and also for some $\beta \in (0,1)$ we get

$$\vartheta(x) < \beta\vartheta(x). \tag{3.20}$$

Now, from (3.19) and (3.20) we get

$$F_j(x+\alpha s) \leq F_j(x) + \alpha\beta\vartheta(x). \tag{3.21}$$

The inequality (3.21) represents the Armijo type inexact line search for quasi Newton's descent algorithm for the RC_1^*. □

3.3 A modified BFGS update of quasi newton method for RC_1^*

The BFGS update formula proposed by Žiga Povalej (2014) is generally applicable for smooth multi-objective optimization problems. However, in the case of RC_1^*, which is a nonsmooth multiobjective optimization problem, there is no guarantee that the BFGS update (2.6) satisfies the curvature condition. Additionally, using the BFGS method to update $H_j(x^k,\xi_i)$ may fail to maintain the positive definiteness of $H_j(x^{k+1},\xi_i)$.

Therefore, we need to modify the BFGS update (2.6) for RC_1^*. To do this, we introduce the concept of $H_j(x^{k+1},\xi_i)$ for the problem RC_1^*. We assume that $H_j(x^k,\xi_i)$ is a symmetric positive definite matrix for each $\xi_i \in U$ and $j \in \Lambda$. Instead of using p^k as in the traditional BFGS update, we consider a vector of the form:

$$\eta^k = \sigma p^k + (1-\sigma)H(x^k,\xi_i)u^k,$$

$0 \leq \sigma \leq 1$ i.e., closest to p_k subject to the condition

$$u^{k^T}\eta^k \geq 0.2u^{k^T}H(x^k,\xi_i)u^k, \tag{3.22}$$

where $u^k = x^{k+1} - x^k = \alpha_k v^k$, $p^k = \nabla\zeta_j(x^{k+1},\xi_i) - \nabla\zeta_j(x^k,\xi_i)$, α_k satisfy (3.21) at x^k and the factor 0.2 chosen based on empirical evidence. Thus the value of σ is given by

$$\sigma = \left\{ \begin{array}{ll} 1, & \text{if } u^{k^T}p^k \geq 0.2u^{k^T}H(x^k,\xi_i)u^k, \\ \frac{0.8u^{k^T}H(x^k,\xi_i)u^k}{u^{k^T}H(x^k,\xi_i)u^k - u^{k^T}p^k}, & \text{if } u^{k^T}p^k < 0.2u^{k^T}H(x^k,\xi_i)u^k \end{array} \right\}. \tag{3.23}$$

Thus the new modified BFGS update formula for RC_1^* is given by

$$H_j(x^{k+1},\xi_i) = H_j(x^k,\xi_i) - \frac{H_j(x^k,\xi_i)u^k(u^k)^T H_j(x^k,\xi_i)}{(u^k)^T H_j(x^k,\xi_i)u^k} + \frac{\eta^k(\eta^k)^T}{(u^k)^T\eta^k}. \tag{3.24}$$

After multiplying (3.24) by u^k we get

$$H_j(x^{k+1},\xi_i)u^k = H_j(x^k,\xi_i)u^k - \frac{H_j(x^k,\xi_i)u^k(u^k)^T H_j(x^k,\xi_i)u^k}{(u^k)^T H_j(x^k,\xi_i)u^k} + \frac{\eta^k {u^k}^T \eta^k}{(u^k)^T\eta^k},$$

which implies,

$$H_j(x^{k+1},\xi_i)u^k = \eta^k. \tag{3.25}$$

Equation (3.25), represents the quasi Newton condition or secant equation. After multiply (3.25) by u^k and with the help of (3.22) we get

$${u^k}^T H_j(x^{k+1},\xi_i)u^k = {u^k}^T\eta^k > 0. \tag{3.26}$$

Equation (3.26) represents the curvature condition which ensures that $H_j(x^{k+1},\xi_i)$ is positive definite for each ξ_i and j. Thus, we have new modified BFGS update formula (3.24) for Hessian update which satisfies quasi Newton condition and curvature condition.

With the derived quasi Newton descent direction, step length size, and the modified BFGS update formula for updating the Hessian, we are now equipped to present the quasi Newton algorithm for solving RC_1^*. Using these components, we can outline the steps of the algorithm as follows:

Algorithm 3.1. (**Quasi Newton algorithm for** RC_1^*)

Step 1 Choose $\epsilon > 0$, $\beta \in (0,1)$ and $x^0 \in \mathbb{R}^n$ and $H_j(x^0,\xi_i)$ is symmetric and positive definite matrix for each ξ_i and j. Set $k := 0$.

Step 2 Solve $Q_N(x^k)$ at x^k to obtain $s^k = s(x^k)$ and $\vartheta(x^k)$. If $|\vartheta(x^k)| < \epsilon$ or $\|s(x^k)\| < \epsilon$ then stop. Otherwise, proceed to Step 3.

Step 3 Choose α_k as the largest $\alpha \in \{\frac{1}{2^r} : r = 1,2,3,\ldots\}$, which satisfies (3.21).

Step 4 Set $x^{k+1} := x^k + \alpha_k s^k$. Update: Generate a positive definite matrix $H_j(x^{k+1},\xi_i)$ with the help of (3.24). Set $k := k+1$, go to Step 2 and repeat this process until we do not reach the stopping criteria.

Upon examining Algorithm 3.1, we can observe that at any given point $x^k \in \mathbb{R}^n$, $Q_N(x^k)$ is a convex programming quadratic subproblem due to the positive definiteness of $H_j(x^k,\xi_i)$ for each j and i. As a result, the unique solution of $Q_N(x^k)$, denoted as $s(x^k)$, exists, ensuring that Step 2 is well-defined. According to Theorem 3.2, the stopping criterion for Algorithm 3.1 can be set as $\|s(x^k)\| < \epsilon$ or $|\vartheta(x^k)| < \epsilon$. If, at iteration k, the stopping criterion is not met (i.e., x^k is not an approximate critical point), we proceed to Step 3. In this step, we find α_k such that α_k is the largest $\alpha \in \{\frac{1}{2^r} : r = 1,2,3,\ldots\}$ satisfying equation (3.21). Next, we calculate $x^{k+1} = x^k + \alpha_k s^k$ in Step 4, and update $H_j(x^k,\xi_i)$ to obtain $H_j(x^{k+1},\xi_i)$ using equation (3.24). The updated $H_j(x^{k+1},\xi_i)$ is a symmetric and positive definite matrix for each j and ξ_i. Finally, we set $k := k+1$ and return to Step 2, repeating this process until an approximate critical point is reached. There is one more thing to know about Algorithm 3.1. When we reach an approximative critical point after reaching the stopping criteria, that critical point will be the approximative weak Pareto optimal solution of RC_1^*.

3.4 Convergence analysis of algorithm 3.1

Initially, we prove Lemma 3.1, which helps to prove the convergence of the sequence generated by Algorithm 3.1.

Lemma 3.1. *If $\vartheta(x)$ is given by (3.9), then for all $k = 0,1,2,\ldots$ and $j \in \Lambda$,*

$$\sum_{r=0}^{k} \alpha_r |\vartheta(x^r)| \leq \beta^{-1}(F_j(x^0) - F_j(x^{k+1})),$$

where $F_j(x^k) = \max_{i \in \bar{\Lambda}} \zeta_j(x^k, \xi_i)$.

Proof. Assume that $\{x^k\}$ is a sequence generated by Algorithm 3.1. Then, by Step 3 of Algorithm 3.1

$$-\alpha_k \vartheta(x^k) \leq \beta^{-1}(F_j(x^k) - F_j(x^{k+1})).$$

Thus we have

$$-\sum_{r=0}^{k} \alpha_r \vartheta(x^r) \leq \beta^{-1}(F_j(x^0) - F_j(x^{k+1})).$$

Also

$$\sum_{r=0}^{k} \alpha_r \vartheta(x^r) \geq -\beta^{-1}(F_j(x^0) - F_j(x^{k+1})). \tag{3.27}$$

Since, we know $\vartheta(x^r) < 0$, then equation (3.27), can be written as

$$\beta^{-1}(F_j(x^0) - F_j(x^{k+1})) \geq \sum_{r=0}^{k} \alpha_r \vartheta(x^r). \tag{3.28}$$

Now, from (3.27) and (3.28)

$$\sum_{r=0}^{k} \alpha_r |\vartheta(x^r)| \leq \beta^{-1}(F_j(x^0) - F_j(x^{k+1})) \tag{3.29}$$

hence the lemma. □

Theorem 3.5. Let $s(x)$ and $\vartheta(x)$ be given by (3.9) and (3.10), respectively, which represent the optimal value and optimal solution of the subproblem $Q_N(x)$. At any given point $x^k \in \mathbb{R}^n$, in the direction $s(x^k)$ the Armijo type inexact line search is given by

$$F_j(x^k + \alpha s^k) \leq F_j(x^k) + \alpha_k \beta \vartheta(x^k). \tag{3.30}$$

Then

1. if $\{x^k\}$ is a sequence generated by Algorithm 3.1. Then accumulation point of $\{x^k\}$ is a Pareto point for F;
2. moreover, if x^0 lies in a compact level-set of F, and $\{x^k\}$ be a sequence generated by Algorithm 3.1. Then $\{x^k\}$ converges to a Pareto optimum $x^* \in \mathbb{R}^n$ for F for the problem RC_1^*.

Proof. To prove 1, we assume x^* is an accumulation point of the sequence $\{x^k\}$ generated by Algorithm 3.1. Then there exists a subsequence $\{x^{k_l}\}$ such that

$$\lim_{l \to \infty} x^{k_l} = x^*.$$

Using the lemma 3.1, with $l = k_l$ and taking the limit $l \to \infty$ from the continuity of F_j

$$\sum_{l=0}^{\infty} \alpha_l |\vartheta(x^l)| < \infty.$$

Therefore

$$\lim_{k \to \infty} \alpha_k \vartheta(x^k) = 0.$$

And in particular, we have

$$\lim_{l \to \infty} \alpha_{k_l} \vartheta(x^{k_l}) = 0. \tag{3.31}$$

Suppose that x^* is non critical point, then

$$\vartheta(x^*) < 0 \quad and \quad s^* \neq 0. \tag{3.32}$$

Now define

$$G : \mathbb{R}^m \to \mathbb{R}, \; G(u) = \max_{j \in \Lambda} u_j.$$

Using the definition of $\vartheta(.)$ and descent direction s^*, we can conclude that there exists a non negative integer q such that

$$G(F(x^* + 2^{-q}s^*) - F(x^*)) < \beta 2^{-q}\vartheta(x^*).$$

Since $G(.)$ and $\vartheta(x)$ are continuous, then

$$\lim_{l \to \infty} s^{k_l} = s^*, \quad \lim_{l \to \infty} \vartheta(x^{k_l}) = \vartheta(x^*).$$

And for l large enough

$$G(F(x^{k_l} + 2^{-q}s^{k_l}) - F(x^{k_l})) < \beta 2^{-q}\vartheta(x^{k_l}), \tag{3.33}$$

which implies

$$F_j(x^{k_l} + 2^{-q}s^{k_l}) - F_j(x^{k_l}) < \beta 2^{-q}\vartheta(x^{k_l}). \tag{3.34}$$

By the Algorithm 3.1, *Step* 3 we have

$$F_j(x^{k_l} + \alpha_{k_l}s^{k_l}) \leq F_j(x^{k_l}) + \beta\alpha_{k_l}\vartheta(x^{k_l}). \tag{3.35}$$

Since α_{k_l} is the largest of $\{\frac{1}{2^n} : n = 1, 2, 3, ...\}$, and $\sigma \in (0, 1)$. Then by equation (3.34) and (3.35) we have

$$\beta\alpha_{k_l}\vartheta(x^{k_l}) \leq \beta 2^{-q}\vartheta(x^{k_l}). \tag{3.36}$$

Which implies

$$\alpha_{k_l} \geq 2^{-q} > 0, \quad \text{for } l \text{ large enough.} \tag{3.37}$$

Hence

$$\lim_{l \to \infty} \alpha_{k_l}\vartheta(x^{k_l}) > 0. \tag{3.38}$$

Which contradicts the equation (3.31). Therefore, the condition in equation (3.32) is not true. Thus x^* is a critical point. And F is strictly convex, so x^* is a Pareto optimum, hence the item 1.
Now to prove the item 2, let L_0 be the $F(x^0)$-level set of F, i.e.,

$$L_0 = \{x \in \mathbb{R}^n : F(x) \leq F(x^0)\}.$$

Since the sequence $F(x^k)$ is $\mathbb{R}^m$-decreasing, we have $x^k \in L_0$ for all k. Therefore, $\{x^k\}$ is bounded, and all its accumulation points lie in L_0. By applying the above theorem, we can conclude that these accumulation points are all Pareto optima for F. Furthermore, since for every $x^k \in L_0$ we have $F(x^{k+1}) \leq F(x^k)$, we can use Lemma 3.8 in Fukuda and Drummond (2014) to show that for any accumulation point x^* of $\{x^k\}$, we have $F(x^*) \leq F(x^k)$ and $\lim_{k\to\infty} F(x^k) = F(x^*)$. Additionally, $F(x)$ is constant among the accumulation points of $\{x^k\}$. Since F is strongly convex, meaning it is strictly $\mathbb{R}^m$-convex, there exists at least one accumulation point of $\{x^k\}$, denoted as x^*. Thus, the proof is complete.

□

3.5 Convergence rate of Algorithm 3.1

Under the usual assumptions, we will prove the convergence of the infinite sequence generated by Algorithm 3.1 to the Pareto optimum with a superlinear rate when full quasi Newton steps are performed, i.e., $\alpha_k = 1$ for sufficiently large k. First, we assume a bound for the descent direction $s(x)$. Next, we prove a bound for $\vartheta(x^{k+1})$ using information from the previous iteration point x^k. Finally, we show that

full quasi Newton steps are performed when k is large enough, i.e., to say $\alpha_k = 1$.

Theorem 3.6. Let x^k be a sequence generated by Algorithm 3.1.

1. If for any k, $\exists\ \lambda_{ij}$ such that
$$\sum_{j\in\Lambda}\sum_{i\in\bar{\Lambda}}\lambda_{ij} = 1,$$
then
$$\vartheta(x^{k+1}) \geq -\frac{1}{2\omega}\|\sum_{j\in\Lambda}\sum_{i\in\bar{\Lambda}}\lambda_{ij}^k\nabla\zeta_j(x^{k+1},\xi_i)\|^2$$
where $\omega > 0$ defined in Definition 2.5.

2. If we assume for any $x \in \mathbb{R}^n$,
$$\|s(x)\|^2 \leq \frac{2}{\omega}\mid\vartheta(x)\mid, \tag{3.39}$$
where $\omega > 0$ defined in Definition 2.5, and let $x^0 \in \mathbb{R}^n$ be in compact level set of F. Then $\{x^k\}$ converges to a Pareto optimum point x^* moreover $\alpha_k = 1$ for k large enough and rate of convergence of $\{x^k\}$ to x^* is superlinear.

Proof. Assume the sequence $\{x^k\}$ is generated by Algorithm 3.1. By (3.10) and (3.5) we have

$$\begin{aligned}
\vartheta(x^{k+1}) &= \sum_{j\in\Lambda}\sum_{i\in\bar{\Lambda}}\lambda_{ij}^k\max_{j\in\Lambda}\max_{i\in\bar{\Lambda}}\{\nabla\zeta_j(x^{k+1},\xi_i)^Ts(x^{k+1}) + \frac{1}{2}v(x^{k+1})^TH_j(x^{k+1},\xi_i)s(x^{k+1})\}\\
&\geq \sum_{j\in\Lambda}\sum_{i\in\bar{\Lambda}}\lambda_{ij}^k\{\nabla\zeta_j(x^{k+1},\xi_i)^Ts(x^{k+1}) + \frac{1}{2}s(x^{k+1})^TH_j(x^{k+1},\xi_i)s(x^{k+1})\}\\
&\geq \min_s\sum_{j\in\Lambda}\sum_{i\in\bar{\Lambda}}\lambda_{ij}^k\{\nabla\zeta_j(x^{k+1},\xi_i)^Ts + \frac{1}{2}s^TH_j(x^{k+1},\xi_i)s\}.
\end{aligned}$$

Therefore,
$$\vartheta(x^{k+1}) \geq \min_s\sum_{j\in\Lambda}\sum_{i\in\bar{\Lambda}}\lambda_{ij}^k\big(\nabla\zeta_j(x^{k+1},\xi_i)^Ts + \frac{\omega}{2}\|s\|^2\big). \tag{3.40}$$

Solving the R.H.S. of inequality (3.40) we get
$$\min_s\sum_{j\in\Lambda}\sum_{i\in\bar{\Lambda}}\lambda_{ij}^k\big(\nabla\zeta_j(x^{k+1},\xi_i)^Ts + \frac{1}{2}\|s\|^2\big) = -\frac{1}{2\omega}\|\sum_{j\in\Lambda}\sum_{i\in\bar{\Lambda}}\lambda_{ij}^k\nabla\zeta_j(x^{k+1},\xi_i)\|^2 \tag{3.41}$$

then by (3.40) and (3.41)
$$\vartheta(x^{k+1}) \geq -\frac{1}{2\omega}\|\sum_{j\in\Lambda}\sum_{i\in\bar{\Lambda}}\lambda_{ij}^k\nabla\zeta_j(x^{k+1},\xi_i)\|^2. \tag{3.42}$$

Hence the proof of item 1.
Now we prove item 2. Since $\{x^k\}$ be a sequence generated by Algorithm 3.1 whose initial point x^0 belongs to compact level set of F. Therefore by Theorem 3.5, $\{x^k\}$ converges to x^*. Here we prove $\{x^k\}$ converges to x^* with superlinear rate. Since x^* is critical point then by the definition there is no descent direction at x^* i.e., $s(x^*) = 0$. Since $\zeta_j(x,\xi_i)$ is twice continuously differantiable for each x and ξ_i, we get

$$\begin{aligned}\zeta_j(x^k+v^k,\xi_i) &\le \zeta_j(x^k,\xi_i)+\nabla\zeta_j(x^k,\xi_i)^T s^k+\frac{1}{2}(s^k)^T H_j(x^k,\xi_i)(s^k)+\frac{\epsilon}{2}\|s^k\|^2\\ &\le \max_{i\in\bar\Lambda}\{\zeta_j(x^k,\xi_i)+\nabla\zeta_j(x^k,\xi_i)^T s^k+\frac{1}{2}(s^k)^T H_j(x^k,\xi_i)(s^k)+\frac{\epsilon}{2}\|s^k\|^2\}\\ &\le \max_{i\in\bar\Lambda}\zeta_j(x^k,\xi_i)+\max_{j\in\Lambda}\max_{i\in\bar\Lambda}\{\nabla\zeta_j(x^k,\xi_i)^T s^k+\frac{1}{2}(s^k)^T H_j(x^k,\xi_i)(s^k)\}+\frac{\epsilon}{2}\|s^k\|^2.\end{aligned}$$

Then

$$\zeta_j(x^k+s^k,\xi_i)\le F_j(x^k)+\vartheta(x^k)+\frac{\epsilon}{2}\|s^k\|^2 \text{ holds for each } \xi_i.$$

Then

$$\max_{i\in\bar\Lambda}\zeta_j(x^k+s^k,\xi_i)\le F_j(x^k)+\vartheta(x^k)+\frac{\epsilon}{2}\|s^k\|^2$$

$$\text{and}\quad F_j(x^k+s^k)-F_j(x^k)\le\vartheta(x^k)+\frac{\epsilon}{2}\|s^k\|^2.$$

Then by (3.39),

$$\begin{aligned}F_j(x^k+s^k)-F_j(x^k) &\le \sigma\theta(x^k)+(1-\sigma)\vartheta(x^k)+\frac{\epsilon}{2}\|s^k\|^2\\ &\le \sigma\vartheta(x^k)+(\epsilon-\omega(1-\sigma))\frac{\|s^k\|^2}{2},\ \text{for all } k\ge k_\epsilon.\end{aligned}$$

This inequality shows that if $\epsilon<\omega(1-\sigma)$, then by Algorithm 3.1, in *Step* 3, $\alpha_k=1$ is accepted for $k\ge k_\epsilon$.

For superlinear convergence suppose that $\epsilon<\omega(1-\sigma)$. Use the Taylor's first order expansion of $\sum_{j\in\Lambda}\sum_{i\in\bar\Lambda}\lambda_{ij}^k\nabla\zeta_j(x^{k+1},\xi_i)$ at x^k, in s^k, we conclude that for any $k\ge k_\epsilon$, it holds

$$\|\sum_{j\in\Lambda}\sum_{i\in\bar\Lambda}\lambda_{ij}^k\nabla\zeta_j(x^{k+1},\xi_i)\|=\|\sum_{j\in\Lambda}\sum_{i\in\bar\Lambda}\lambda_{ij}^k\nabla\zeta_j(x^{k+1},\xi_i)-0\|.$$

By (3.6),

$$\begin{aligned}\|\sum_{j\in\Lambda}\sum_{i\in\bar\Lambda}\lambda_{ij}^k\nabla\zeta_j(x^{k+1},\xi_i)\| &= \|\sum_{j\in\Lambda}\sum_{i\in\bar\Lambda}\lambda_{ij}^k\nabla\zeta_j(x^{k+1},\xi_i)-\big(\sum_{j\in\Lambda}\sum_{i\in\bar\Lambda}\lambda_{ij}^k\nabla\zeta_j(x^k,\xi_i)+\sum_{j\in\Lambda}\sum_{i\in\bar\Lambda}\lambda_{ij}^k H_j(x^k,\xi_l)s^k\big)\|\\ &\le \|\sum_{j\in\Lambda}\sum_{i\in\bar\Lambda}\lambda_{ij}^k\nabla\zeta_j(x^k+s^k,\xi_l)-(\sum_{j\in\Lambda}\sum_{i\in\bar\Lambda}\lambda_{ij}^k\nabla\zeta_j(x^k,\xi_l)\\ &+\sum_{j\in\Lambda}\sum_{i\in\bar\Lambda}\lambda_{ij}^k H_j(x^k,\xi_i)^T s^k)\|\\ &\le \sum_{j\in\Lambda}\sum_{i\in\bar\Lambda}\lambda_{ij}^k\|\epsilon s^k\|.\end{aligned}$$

Thus we get

$$\|\sum_{j\in\Lambda}\sum_{i\in i\in\bar\Lambda}\lambda_{ij}^k\nabla\zeta_j(x^{k+1},\xi_i)\|\le\epsilon\|s^k\| \tag{3.43}$$

By (3.42) and (3.43) we get

$$\vartheta(x^{k+1})\ge-\frac{1}{2\omega}(\epsilon\|s^k\|)^2\quad\text{and}\quad|\,\vartheta(x^{k+1})\,|\le\frac{1}{2\omega}(\epsilon\|s^k\|)^2. \tag{3.44}$$

Since $\alpha_k = 1$ and $x^{k+1} = x^k + s^k$, from (3.39) and (3.44) we have

$$\|x^{k+1} - x^{k+2}\| = \|s^{k+1}\| \leq \frac{\epsilon}{\omega}\|s^k\|.$$

Thus if $k \geq k_\epsilon$ and $j \geq 1$ then

$$\|x^{k+j} - x^{k+j+1}\| \leq (\frac{\epsilon}{\omega})^j \|x^k - x^{k+1}\|. \tag{3.45}$$

To prove the superlinear convergence rate take $0 < \aleph < 1$ and define

$$\epsilon^* = \min\{1 - \sigma, \frac{\aleph}{1 + 2\aleph}\}\omega.$$

If $\epsilon < \epsilon^*$ and $k \geq k_\epsilon$, then by (3.45) and convergence of $\{x^k\}$ to x^*, we have

$$\begin{aligned}\|x^* - x^{k+1}\| &\leq \sum_{j=1}^{\infty} \|x^{k+j} - x^{k+j+1}\| \\ &\leq \sum_{j=1}^{\infty} (\frac{\aleph}{1 + 2\aleph})^j \|x^k - x^{k+1}\| \\ &= (\frac{\aleph}{1 + \aleph})\|x^k - x^{k+1}\|.\end{aligned}$$

$$\|x^* - x^{k+1}\| \leq (\frac{\aleph}{1 + \aleph})\|x^k - x^{k+1}\|. \tag{3.46}$$

Now with the help of triangle inequality and (3.46) we get

$$\begin{aligned}\|x^* - x^k\| &\leq \|x^k - x^{k+1}\| - \|x^{k+1} - x^*\| \\ &\leq \|x^k - x^{k+1}\| - (\frac{\aleph}{1 + \aleph})\|x^k - x^{k+1}\|\end{aligned}$$

therefore

$$\|x^* - x^k\| \leq (\frac{1}{1 + \aleph})\|x^k - x^{k+1}\|. \tag{3.47}$$

Now by (3.46) and (3.47), we conclude if $\epsilon < \epsilon^*$ and $k \geq k_\epsilon$, then

$$\|x^* - x^{k+1}\| \leq \aleph\|x^* - x^k\|,$$

$$\text{i.e.,} \quad \frac{\|x^* - x^{k+1}\|}{\|x^* - x^k\|} \leq \aleph.$$

Since $\aleph$ was arbitrary in $(0, 1)$, thus the above quotient tends to zero and hence $\{x^k\}$ converges to x^* with superlinear rate. □

3.6 Numerical results

In this subsection, we consider some numerical examples, and solve with the help of quasi Newton algorithm (Algorithm 3.1) and compare with weighted sum method. To solve numerical examples with the help of Algorithm 3.1, a *Python* code is developed. In our computations, we consider $\epsilon = 10^{-4}$ as tolerance or maximum 5000 iterations is considered as stopping criteria. Thus, in our computations, we use $\|s^k\| < 10^{-4}$ or $|\vartheta| < 10^{-4}$ or maximum 5000 number of iterations as a stopping criteria.

Since we know the solution of multiobjective optimization problem is not an isolated point but a set of Pareto optimal solutions. We used a multi-start technique to generate a well distributed approximation of Pareto front. Here 100 uniformly distributed random point is chosen between lb and ub (where

$lb, ub \in \mathbb{R}^n$ and $lb < ub$) and Algorithm 3.1 is executed separately. The nondominated set of collection of critical points is considered as an approximate Pareto front. We have compared the approximate Pareto fronts obtained by Algorithm 3.1 with the approximate Pareto fronts obtained by weighted sum method. In weighted sum method, we solve the following single objective optimization problem

$$\min_{x\in\mathbb{R}^n} \left(w_1\phi_1(x) + w_2\phi_2(x) + \cdots + w_m\phi_m(x)\right),$$

where $(w_1, w_2, \ldots, w_m) = w \geq 0$, $w \neq 0$ using the technique developed in Bazaraa and Goode (1982) with initial approximation $x^0 = \frac{1}{2}(lb + ub)$. For bi-objective optimization problems, we have considered weights $(1, 0)$, $(0, 1)$, and 98 random weights uniformly distributed in the square area of $[0, 1] \times [0, 1]$ (i.e., any 98 random weights uniformly distributed in this area). On the other hand, for three-objective optimization problems, we have considered four types of weights: $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$, and 97 random weights uniformly distributed in the cubic area of $[0, 1] \times [0, 1] \times [0, 1]$ (i.e., any 97 random weights uniformly distributed in this area).

To find the descent direction at k^{th} iteration we solve subproblem

$$\begin{gathered} Q_N(x^k) : \min_{(t,s)\in\mathbb{R}\times\mathbb{R}^n} t \\ s.t. \qquad \zeta_j(x^k, \xi_i) + \nabla\zeta_j(x^k, \xi_i)^T s + \tfrac{1}{2}s^T H_j(x^k, \xi_i)s - F_j(x) \leq t, \quad \forall i \in \bar{\Lambda} \text{ and } j \in \Lambda, \\ lb \leq x + s \leq ub. \end{gathered}$$

For step length size α_k, we choose α_k, as largest $\alpha \in \{\frac{1}{2^r} : r = 1, 2, 3, \ldots\}$, which satisfies (3.21) i.e.,

$$F_j(x^k + \alpha_k s) \leq F_j(x^k) + \alpha_k\beta\vartheta(x^k).$$

For Positive definite Hessian update we use modified BFGS formula which is given in (3.24).

Example 3.1. (Two dimensional bi-objective non-convex problem under uncertainty set of two elements)
Consider the uncertain bi-objective optimization problem

$$P(U) = \{P(\xi) : \xi \in U\}$$

such that

$$P(\xi) : \qquad \min_{x\in\mathbb{R}^2} \zeta(x, \xi),$$

where $\zeta : \mathbb{R}^2 \times U \to \mathbb{R}^2$ such that $\zeta(x, \xi) = (\zeta_1(x, \xi), \zeta_2(x, \xi))$.
Consider $U = \{(1, 2),\ (2, 3)\} \subset \mathbb{R}^2$ such that $\xi^1 = (\xi^1_1, \xi^1_2) = (1, 2)$ $\xi^2 = (\xi^2_1, \xi^2_2) = (2, 3)$, and $\bar{\Lambda} = \{i : \xi_i \in U\} = \{1, 2\}$.

$$\zeta(x, \xi^i) = \left(\frac{1}{4}\big((x_1 - \xi^i_1)^4 + 2(x_2 - \xi^i_2)^4\big),\ (\xi^i_1 x_2 - \xi^i_2 x^2_1)^2 + (1 - \xi^i_1 x_1)^2\right),$$

$$\text{and} \quad \zeta_1(x, \xi^i) = \frac{1}{4}(x_1 - \xi^i_1)^4 + 2(x_2 - \xi^i_2)^4,\ \zeta_2(x, \xi^i) = (\xi^i_1 x_2 - \xi^i_2 x^2_1)^2 + (1 - \xi^i_1 x_1)^2,\ i = 1,\ 2.$$

objective-wise worst-case cost type robust counterpart to $P(U)$ is given by

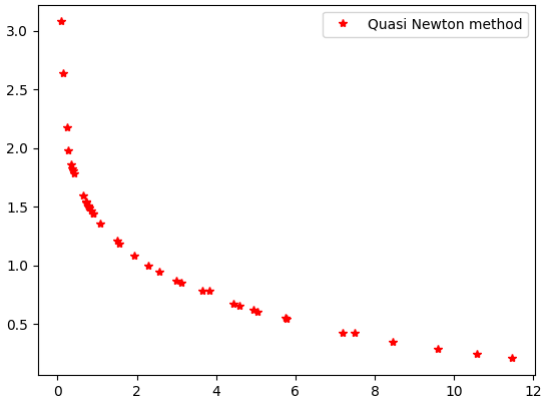
$$RC^*_1 : \qquad \min_{x\in\mathbb{R}^2} F(x), \tag{3.48}$$

$$\text{where } F(x) = (F_1(x), F_2(x)) \text{ and}$$

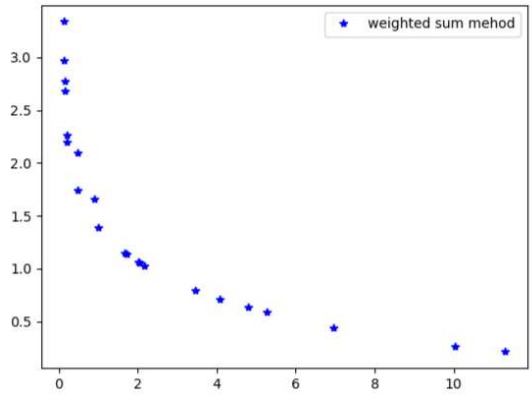
$$F_1(x) = \max_{i\in\{1,2\}} \zeta_1(x, \xi^i) = \max\{\frac{1}{4}\big((x_1 - 1)^4 + 2(x_2 - 2)^4\big),\ \frac{1}{4}\big((x_1 - 2)^4 + 2(x_2 - 3)^4\big)\},$$

$$F_2(x) = \max_{i\in\{1,2\}} \zeta_2(x, \xi^i) = \max\{(x_2 - 2x^2_1)^2 + (1 - x_1)^2, (2x_2 - 3x^2_1)^2 + (1 - 2x_1)^2\}.$$

Now we solve RC^*_1 with the help of Algorithm 3.1 (quasi Newton method). In Algorithm 3.1, we



(a) Approximate Pareto front generated by quasi Newton algorithm



(b) Approximate Pareto front generated by weighted sum method

Figure 1: Comarison of aproximate Pareto fronts generated by quasi Newton algorithm and weighted sum method.

consider the initial point $x^0 = (1.10203444,\ 1.93225526)$, $\epsilon = 10^{-4}$, and for $j = 1, 2$, $i = 1, 2$ $H_j(x^0, \xi^i)$ is taken as follows

$$H_1(x^0,\xi^1) = \begin{pmatrix} 0.15716692 & 0.08803005 \\ 0.08803005 & 0.0844797 \end{pmatrix},\ H_1(x^0,\xi^2) = \begin{pmatrix} 5.02235556 & 1.7801104 \\ 1.7801104 & 8.00564985 \end{pmatrix},$$

$$H_2(x^0,\xi^1) = \begin{pmatrix} 44.14712932 & -9.00388107 \\ -9.00388107 & 1.85298502 \end{pmatrix},\ H_2(x^0,\xi^2) = \begin{pmatrix} 91.45875562 & -26.67973288 \\ -26.67973288 & 7.78744484 \end{pmatrix}.$$

Now, we solve subproblem $Q_N(x)$ at x^0 and we get $s^0 = (-6.48808533e-03,\ 9.42476577e-03)$, $t^0 = -7.58041271e-06$ $\alpha_0 = 1$. Then we find x^1 such as follows: $x^1 = x^0 + \alpha_0 s^0 = (1.09554636,\ 1.94168003)$. Now we find $F(x^0)$ and $F(x^1)$ as follows

$$F_1(x^0) = \max\{\zeta_1(x^0,\xi^1),\ \zeta_1(x^0,\xi^2)\},$$

$$\text{where}\ \ \zeta_1(x^0,\xi^1) = 3.76284389e-05,\ \zeta_1(x^0,\xi^2) = 8.12436783e-01.$$

Thus, $F_1(x^0) = 0.81243678$. Also,

$$F_2(x^0) = \max\{\zeta_2(x^0,\xi^1),\ \zeta_2(x^0,\xi^2)\},$$

$$\text{where}\ \ \zeta_2(x^0,\xi^1) = 2.57126451e-01,\ \zeta_2(x^0,\xi^2) = 1.49865417.$$

Thus, $F_2(x^0) = 1.49865417$. Hence, $F(x^0) = (0.81243678,\ 1.49865417)$.

$$F_1(x^1) = \max\{\zeta_1(x^1,\xi^1),\ \zeta_1(x^1,\xi^2)\},$$

$$\text{where}\ \ \zeta_1(x^1,\xi^1) = 2.66192924e-05,\ \zeta_1(x^1,\xi^2) = 7.94541995e-01.$$

Thus, $F_1(x^1) = 0.79454199$.

$$F_2(x^1) = \max\{\zeta_2(x^1,\xi^1),\ \zeta_2(x^1,\xi^2)\}$$

$$\text{where}\ \ \zeta_2(x^1,\xi^1) = 2.19593161e-01,\ \zeta_2(x^1,\xi^2) = 1.49861808.$$

Thus, $F_2(x^1) = 1.49861808$. Hence, $F(x^1) = (0.79454199,\ 1.49861808)$. We can observe that $F(x^1) = (0.79454199,\ 1.49861808) < F(x^0) = (0.81243678,\ 1.49865417)$ i.e., the function value decreases com-

ponent wise in the descent direction. Now we update Hessian matrix with the help of $H_j(x^0, \xi^i)$.

$$H_1(x^1,\xi_1) = \begin{pmatrix} 0.0352717 & 0.02121372 \\ 0.02121372 & 0.06300125 \end{pmatrix},\ H_1(x^1,\xi_2) = \begin{pmatrix} 3.18805989 & 1.47432974 \\ 1.47432974 & 9.75153934 \end{pmatrix},$$

$$H_2(x^1,\xi_1) = \begin{pmatrix} 43.57305567 & -8.82511574 \\ -8.82511574 & 1.80401205 \end{pmatrix},\ H_2(x^1,\xi_2) = \begin{pmatrix} 89.91329472 & -26.32939674 \\ -26.32939674 & 7.71467964 \end{pmatrix}.$$

With the help of x^1, $H_j(x^1,\xi_i)$ for $j = 1,2$ and $i = 1,2$, we solve subproblem and we get $t^1 = -5.70455549e-06$, $s^1 = (-8.44268112e-03,\ 4.07150091e-03)$ and after 3 iteration we get critical point $x^* = (1.08710368,\ 1.94575153)$, which will be the approximate weak Pareto optimal solution to the problem RC_1^* and the corresponding nondominated point to x^* is given by $F(x^*) = (0.79127969,\ 1.49856185)$. Also, by the Theorem 2.1 and Algorithm 3.1, x^* will be the robust weak Pareto optimal solution to $P(U)$.

Now we prove $x^* = (1.08710368,\ 1.94575153)$ is a weak Pareto optimal solution to the problem RC_1^*. By observation we have $\zeta_1(x^*,\xi^1) = 1.87211543e-05$, $\zeta_1(x^*,\xi^2) = 7.91279695e-01$, $\zeta_2(x^*,\xi^1) = 1.82175041e-01$, $\zeta_2(x^*,\xi^2) = 1.49856185$ and $F(x^*) = (0.79127969,\ 1.49856185)$. These imply $I_1(x^*) = \{2\}$ and $I_2(x^*) = \{2\}$. The sub differential of $F_1(x)$ and $F_2(x)$ at x^* are given as follows:
$\partial F_1(x^*) = conv\{\nabla\zeta_1(x,\xi^i) : i \in I_1(x^*))\} = conv\{(-0.7607892531,\ -2.3434674942)\}$
$\partial F_2(x^*) = conv\{\nabla\zeta_2(x,\xi^i) : i \in I_2(x^*))\} = conv\{(0.1816117909,\ 1.38447930717)\}$. Also $\partial F(x) = conv\left(\cup_{j=1}^2 \partial F_j(x^*)\right) = conv\{(-0.7607892531, -2.3434674942),\ (0.1816117909, 1.38447930717)\}$.
Thus $0 \in conv\left(\cup_{j=1}^2 \partial F_j(x^*)\right)$ then x^* will be approximate weak Pareto optimal solution to the problem RC_1^* and the corresponding nondominated point to x^* is given by $F(x^*) = (0.79127969,\ 1.49856185)$. Also, by the Theorem 2.1, x^* will be the approximate robust weak Pareto optimal solution to $P(U)$. Comparison of approximate Pareto front generated by quasi Newton method and weighted sum method of the Example 3.1 is given in the Figure 1.

Example 3.2. (Two dimensional three-objective non-convex problem under uncertainty set of three elements)
Consider the uncertain three objective optimization problem

$$P(U) = \{P(\xi) : \xi \in U\}$$

such that

$$P(\xi): \qquad \min_{x\in\mathbb{R}^2} \zeta(x,\xi),$$

where $\zeta(x,\xi) = (\zeta_1(x,\xi), \zeta_2(x,\xi), \zeta_3(x,\xi))$, $\zeta : \mathbb{R}^2 \times U \to \mathbb{R}^3$. Consider $U = \{(5,3),\ (5,6),\ (4,1)\}$ such that $\xi^1 = (\xi_1^1, \xi_2^1) = (5,3)$, $\xi^2 = (\xi_1^2, \xi_2^2) = (5,6)$, $\xi^3 = (\xi_1^3, \xi_2^3) = (4,1)$, and $\bar{\Lambda} = \{i : \xi_i \in U\} = \{1,2,3\}$.

$$\zeta(x,\xi^i) = (x_1^2 + \xi_1^i x_2^4 + \xi_1^i\xi_2^i x_1x_2,\ 5x_1^2 + \xi_1^i x_2^2 + \xi_2^i x_1^4 x_2,\ e^{-\xi_1^i x_1 + \xi_2^i x_2} + x_1^2 - \xi_1^i x_2^2),\ \text{where } \xi^i = (\xi_1^i, \xi_2^i).$$

Therefore,

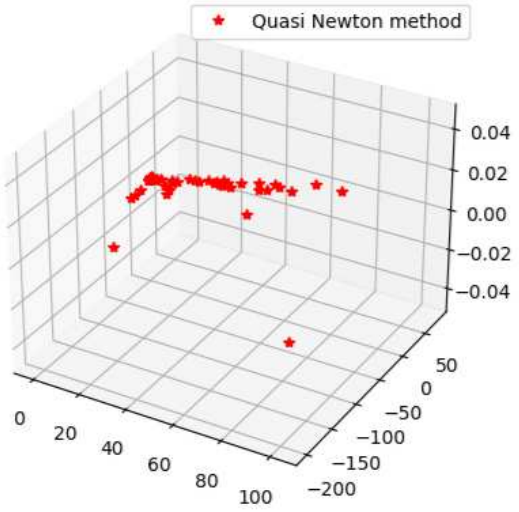
$$\zeta_1(x,\xi^i) = x_1^2 + \xi_1^i x_2^4 + \xi_1^i\xi_2^i x_1x_2,\ i = 1,2,3,$$
$$\zeta_2(x,\xi^i) = 5x_1^2 + \xi_1^i x_2^2 + \xi_2^i x_1^4 x_2,\ i = 1,2,3,$$
$$\zeta_3(x,\xi^i) = e^{-\xi_1^i x_1 + \xi_2^i x_2} + x_1^2 - \xi_1^i x_2^2,\ i = 1,2,3.$$

objective-wise worst-case cost type robust counterpart to $P(U)$ is given by

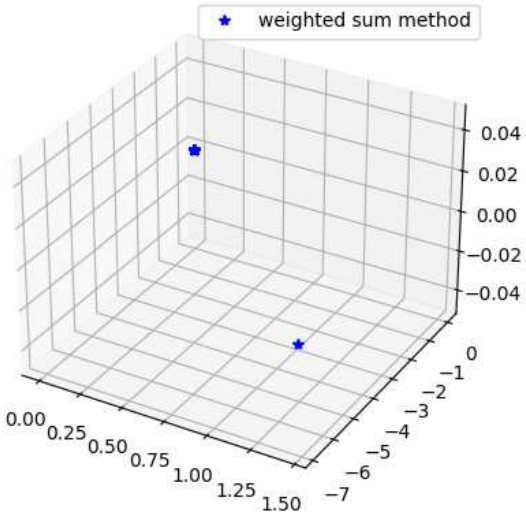
$$RC_1^*: \qquad \min_{x\in\mathbb{R}^2} F(x), \tag{3.49}$$

where $F(x) = (F_1(x), F_2(x))$ and

$$F_1(x) = \max_{i\in\bar{\Lambda}} \zeta_1(x,\xi^i) = \max_{i\in\bar{\Lambda}}\{x_1^2 + 5x_2^4 + 15x_1x_2,\ x_1^2 + 5x_2^4 + 30x_1x_2,\ x_1^2 + 4x_2^4 + 4x_1x_2\},$$



(a) Approximate Pareto front generated by quasi Newton algorithm for RC_1^*



(b) Approximate Pareto front generated by weighted sum method for RC_1^*

Figure 2: Comarison of aproximate Pareto fronts generated by quasi Newton algorithm and weighted sum method for Example 3.2

$$F_2(x) = \max_{i\in\bar{\Lambda}} \zeta_2(x,\xi^i) = \max\{5x_1^2 + 5x_2^2 + 3x_1^4x_2,\ 5x_1^2 + 5x_2^2 + 6x_1^4x_2,\ 5x_1^2 + 4x_2^2 + x_1^4x_2\}$$

$$F_3(x) = \max_{i\in\bar{\Lambda}} \zeta_3(x,\xi^i) = \max\{e^{-5x_1+3x_2} + x_1^2 - 5x_2^2,\ e^{-5x_1+6x_2} + x_1^2 - 5x_2^2,\ e^{-4x_1+x_2} + x_1^2 - 4x_2^2\}.$$

Example 3.3. ***(Two dimensional three objective convex problem under uncertainty set of three elements)***
Consider the uncertain bi-objective optimization problem

$$P(U) = \{P(\xi) : \xi \in U\}$$

such that

$$P(\xi): \qquad \min_{x\in\mathbb{R}^2} \zeta(x,\xi),$$

where $\zeta(x,\xi) = (\zeta_1(x,\xi), \zeta_2(x,\xi))$, $\zeta : \mathbb{R}^2 \times U \to \mathbb{R}^3$. Consider $U = \{(4,1),\ (5,2),\ (6,4)\}$ such that $\xi^1 = (\xi_1^1, \xi_2^1) = (5,3)$, $\xi^2 = (\xi_1^2, \xi_2^2) = (5,6)$, $\xi^3 = (\xi_1^3, \xi_2^3) = (4,1)$, and $\bar{\Lambda} = \{i : \xi_i \in U\} = \{1,2,3\}$.

$$\zeta(x,\xi^i) = (100\xi_1(x_2 - x_1^2)^2 + \xi_2^i(1-x_1)^2, (x_2 - \xi_1^i)^2 + \xi_2^i x_1^2,\ \xi_1^i x_1^2 + 3\xi_2^i x_2^2),$$

where $\xi = (\xi_1^i, \xi_2^i)$.

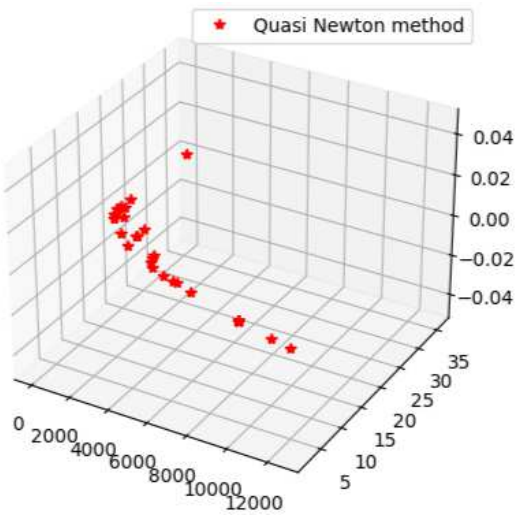
$$\zeta_1(x,\xi^i) = 100\xi_1^i(x_2 - x_1^2)^2 + \xi_2^i(1-x_1)^2,\ i = 1,2,3,$$
$$\zeta_2(x,\xi^i) = (x_2 - \xi_1^i)^2 + \xi_2^i x_1^2,\ i = 1,2,3,$$
$$\zeta_3(x,\xi^i) = \xi_1^i x_1^2 + 3\xi_2^i x_2^2\ i = 1,2,3.$$

objective-wise worst-case cost type robust counterpart to $P(U)$ is given by

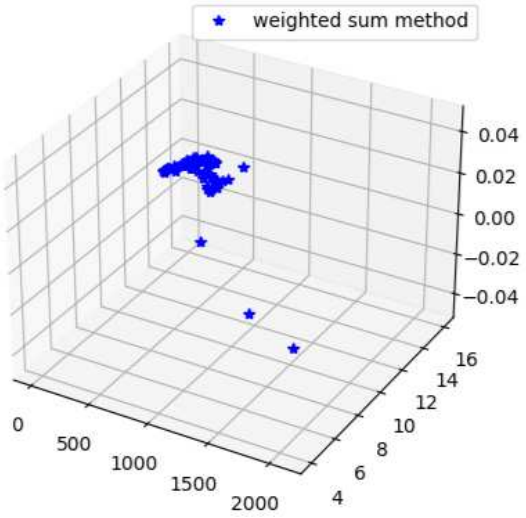
$$RC_1^*: \qquad \min_{x\in\mathbb{R}^2} F(x), \tag{3.50}$$

where $F(x) = (F_1(x), F_2(x), F_3(x))$ and

$$F_1(x) = \max_{i\in\bar{\Lambda}} \zeta_1(x,\xi^i) = \max\{400(x_2 - x_1^2)^2 + (1-x_1)^2,\ 500(x_2 - x_1^2)^2 + 2(1-x_1)^2,\ 600(x_2 - x_1^2)^2 + 4(1-x_1)^2\},$$



(a) Approximate Pareto front generated by quasi Newton algorithm for RC_1^*



(b) Approximate Pareto front generated by weighted sum method for RC_1^*

Figure 3: Comarison of aproximate Pareto fronts generated by quasi Newton algorithm and weighted sum method for Example 3.3

$$F_2(x) = \max_{i\in\bar{\Lambda}} \zeta_2(x,\xi^i) = \max\{(x_2-4)^2 + x_1^2,\ (x_2-5)^2 + 2x_1^2,\ \ (x_2-6)^2 + 4x_1^2\}$$

$$F_3(x) = \max_{i\in\bar{\Lambda}} \zeta_3(x,\xi^i) = \max\{4x_1^2 + 3x_2^2,\ 5x_1^2 + 6x_2^2,\ 6x_1^2 + 12x_2^2\}.$$

Based on the examples discussed above, we can observe that the objective functions in Example 3.1 and Example 3.2 are nonconvex, while those in Example 3.3 are non convex. From the figures associated with each example, it is evident that Algorithm 3.1 successfully generates good approximations of the Pareto fronts in both convex and non convex cases. However, when examining the Figure of Example 3.2, it becomes clear that the weighted sum method fails to produce even an approximate front in the non convex case.

In addition to these examples 20 test problems are constructed and the weighted sum method and Algorithm 3.1 are compared using performance profiles.

Test Problems: The above two are included in a set of test problems that are generated and executed using both techniques. Table 1 provides details of these test problems. Table 1's columns m and n stand for the number of objective functions and the size of the decision variable (x). One can see that in our execution, we took into account 2 or 3 objective 1, 2, 3, 5, and 10 dimensional problems. Each test problem's explicit form is provided in Appendix A.

Remark 3.1. *Since we developed a quasi Newton method for RC_1^* which is the robust counterpart of $P(U)$. Therefore, RC_1^* is considered as the test problem corresponding to some $P(U)$ in the Appendix A.*

The weighted sum method and Algorithm 3.1 are compared using performance profiles. Performance profiles are employed to contrast various approaches (see Ansary and Panda (2015, 2019, 2020) for more details of performance profiles). The cumulative function $\rho(\tau)$ that represents the performance ratio in relation to a specific metric and a set of methods is known as a performance profile. Give a set of methods $\mathcal{SO}$ and a set of problems $\mathcal{P}$, let $\varsigma_{p,s}$ be the performance of solver s on solving p. The performance ratio is defined as $r_{p,s} = \varsigma_{p,s}/\min_{s\in\mathcal{SO}} \varsigma_{p,s}$. The cumulative function $\rho_s(\tau)$ $(s \in \mathcal{SO})$ is defined as

$$\rho_s(\tau) = \frac{|\{p \in \mathcal{P} : r_{p,s} \le \tau\}|}{|P|}.$$

To justify how much well-distributed this set is, the following metrics are considered for computing

Problem	(m, n, p)	lb^T	ub^T
TP1	$(2, 2, 2)$	$(-2, -2)^T$	$(5, 5)^T$
TP2	$(3, 2, 3)$	$(-1, -1)^T$	$(5, 2)^T$
TP3	$(3, 2, 3)$	$(-1, -1)^T$	$(5, 2)^T$
TP4	$(2, 2, 2)$	$(-5, -5)^T$	$(5, 5)^T$
TP5	$(2, 1, 2)$	-5	5
TP6	$(2, 3, 3)$	$(0, 0, 0)^T$	$(1, 1, 1)^T$
TP7	$(2, 1, 2)$	-3	3
TP8	$(2, 2, 2)$	$(-4, -4)$	$(4, 4)$
TP9	$(2, 3, 3)$	$(-1, -2, -1)^T$	$(1, 1, 2)^T$
TP10	(3,3,3)	$(1, -2, 0)^T$	$(3.5, 2.0, 1.0)^T$
TP11	(2,2,2)	$(-6, -6)^T$	$(6, 4)^T$
TP12	(3,3,3)	$(-1, -1, -1)$	$(5, 5, 5)$
TP13	(3,2,3)	$(-1, -1, -1)$	$(5, 5, 5)$
TP14	(2,1,2)	-100	100
TP15	(2,2,2)	$(-2, -2)$	$(5, 5)$
TP16	(3,3,3)	$(0, 0, 0)$	$(1, 1, 1)$
TP17	(3,2,3)	$(-4, -4)$	$(5, 5)$
TP18	(2,2,2)	$(.01, .001)$	$(1, 1)$
TP19	(2,5,2)	$(.001, ..., 001)$	$(1, ..., 1)$
TP20	(3,10,3)	$(0.001, .., 0.001)$	$(1, ..., 1)$

Table 1: Details of test problems

performance profile.

Δ-spread metric: Let $x^1, x^2, \ldots, x^N$ be the set of points obtained by a solver s for problem p and let these points be sorted by $f_j(x^i) \leq f_j(x^{i+1})$ $(i = 1, 2, \ldots, N-1)$. Assume that x^0, calculated over all the approximated Pareto fronts obtained by various solvers, is the best known approximation of the global minimum of f_j, and x^{N+1}, the best known approximation of the global maximum of f_j. Define $\bar{\delta_j}$ as the average of the distances $\delta_{i,j}$, $i = 1, 2, \ldots, N-1$. For an algorithm s and a problem p, the spread metric $\Delta_{p,s}$ is

$$\Delta_{p,s} := \max_{j \in \Lambda_m} \left(\frac{\delta_{0,j} + \delta_{N,j} + \Sigma_{i=1}^{N-1} |\delta_{i,j} - \bar{\delta_j}|}{\delta_{0,j} + \delta_{N,j} + (N-1)\bar{\delta_j}} \right).$$

Hypervolume metric: Hypervolume metric of an approximate Pareto front with respect to a reference point P_{ref} is defined as the volume of the entire region dominated by the efficient solutions obtained by a method with respect to the reference point. We have used the codes from https://github.com/anyoptimization/pymoo to calculate hypervolume metric. Higher values of $hv_{p,s}$ indicate better performance using hypervolume metric. So while using the performance profile of the solvers measured by hypervolume metric we need to set $\widetilde{hv_{p,s}} = \frac{1}{hv_{p,s}}$.
Performance profile using Δ spread metric and hypervolume metric are given in Figures 4 and 5 respectively.
In addition, using the number of iterations and function evaluations, we computed performance profiles. Both methods use the forward difference formula to calculate gradients, which necessitates n additional function evaluations. If $\#$ *Iter* and $\#$ *Fun* denote the number of iterations and number of function evaluations required to solve a problem respectively then total function evaluation is $\#Fun + n\#Iter$. Figures 6 and 7 show the performance profiles using the number of iterations and the number of function evaluations, respectively. The performance profile figures show that the Algorithm 3.1 typically outperforms the weighted sum method.

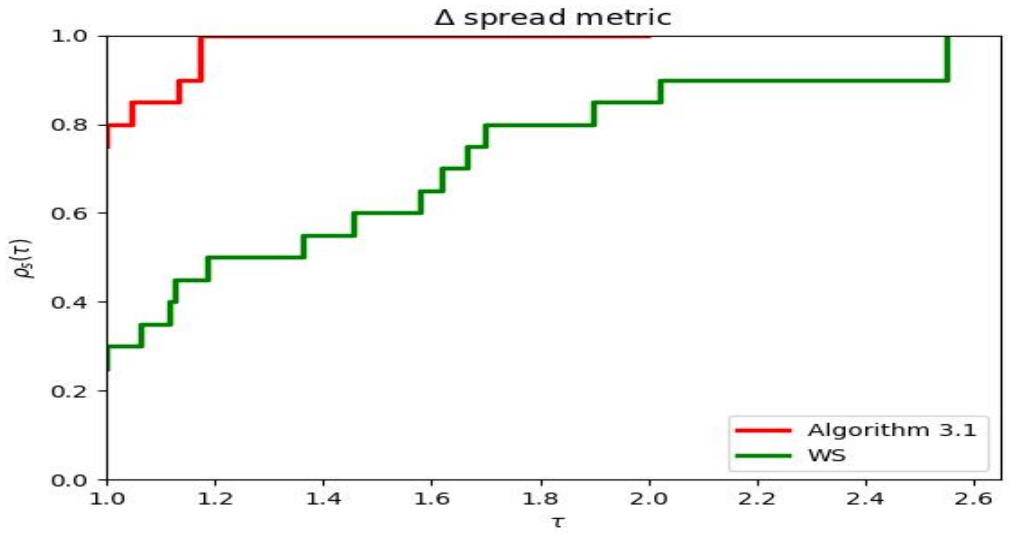


Figure 4: Performance profile using Δ spread metric

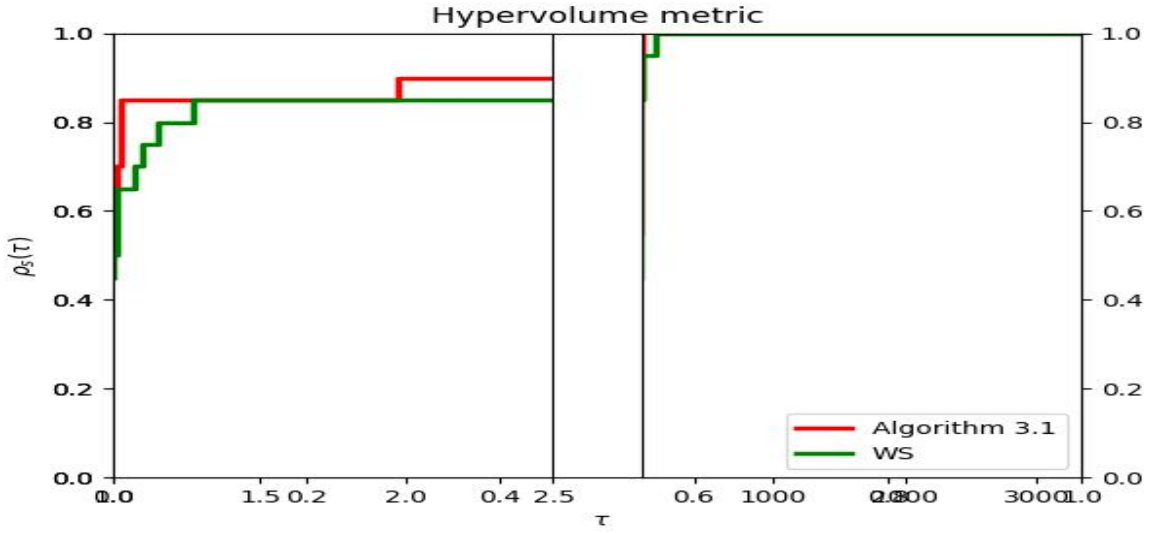


Figure 5: Performance profile using Hypervolume metric

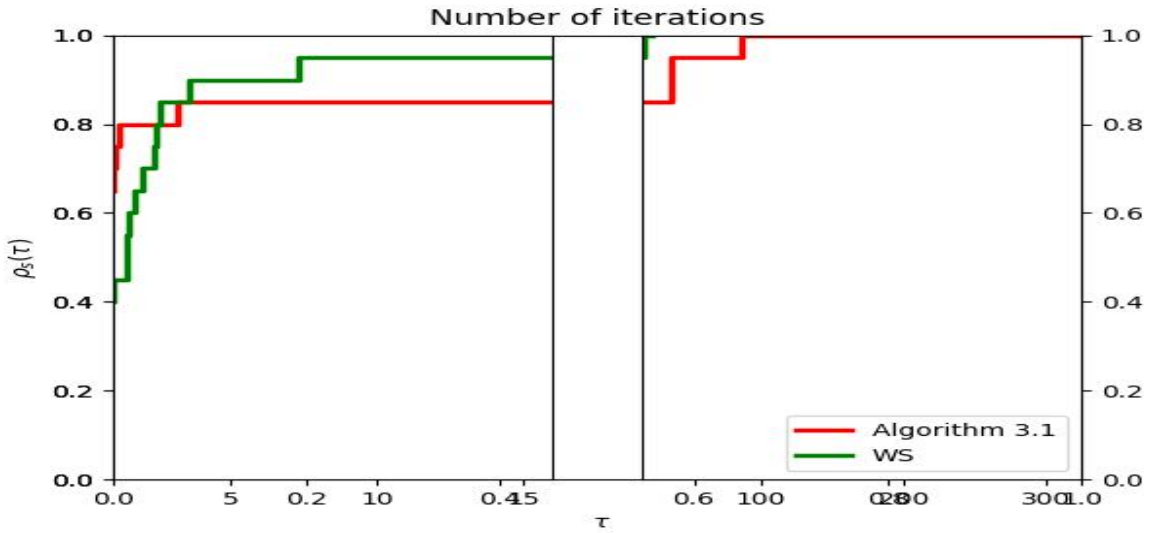


Figure 6: Performance profile using number of iterations

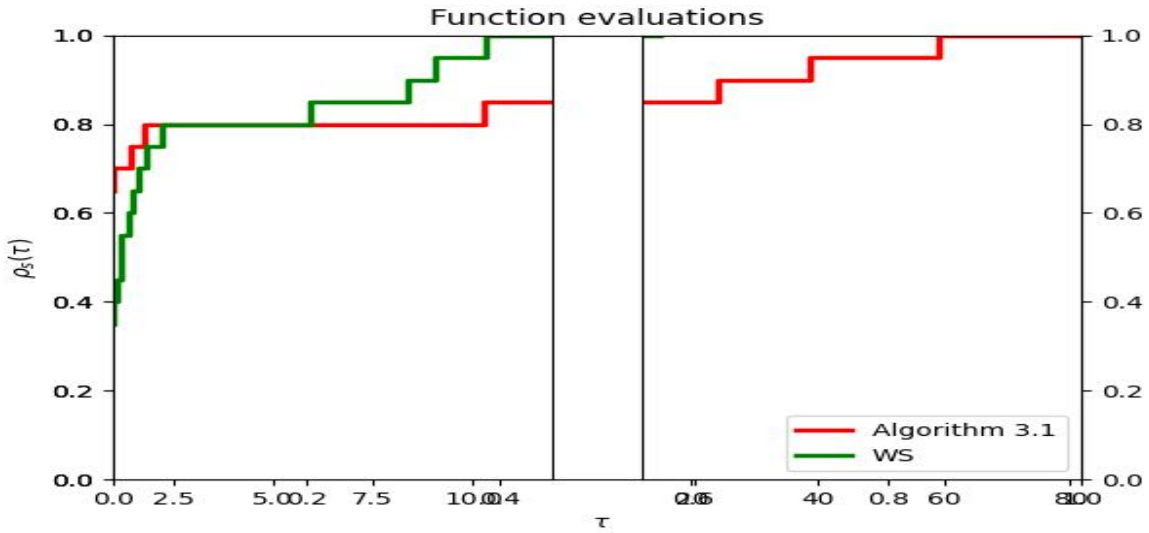


Figure 7: Performance profile using function evaluations

The advantage of Algorithm 3.1 is that it eliminates the need to choose pre-order weights, which is a challenge in the weighted sum method. Consequently, Algorithm 3.1 enables us to obtain good approximations of Pareto fronts without the requirement of pre-order weight selection. Theorem 2.1 demonstrate that the Pareto optimal solution for the problem RC_1^* is the robust Pareto optimal solution to the $P(U)$. Consequently, the approximate Pareto front obtained for RC_1^* is also an approximate Pareto front for $P(U)$. By utilizing RC_1^* and Algorithm 3.1, our main objective of discovering the approximate Pareto front for the multiobjective optimization problem $P(U)$ with a finite parameter uncertainty set has been achieved.

4 Conclusions

We solved an uncertain multiobjective optimization problem $P(U)$ defined in (2.1), using its robust counterpart RC_1^*, which is a deterministic multiobjective optimization problem. By solving RC_1^*, we obtain the solution for $P(U)$, eliminating the need to directly solve $P(U)$. To tackle RC_1^*, we employed a quasi Newton method, assuming strong convexity of the objective function. This assumption ensures that the necessary condition for optimality is also sufficient for Pareto optimality. Our approach involved developing a quasi Newton algorithm to find critical points. This algorithm solves a subproblem to determine the quasi Newton descent direction and employs an Armijo-type inexact line search technique. We also incorporated a modified BFGS formula. By combining these elements, we constructed a quasi Newton algorithm that generates a sequence converging to a critical point. This critical point serves as the Pareto optimal solution for RC_1^* and the robust Pareto optimal solution for the uncertain multiobjective optimization problem $P(U)$. Under certain assumptions, we proved that the sequence generated by the quasi Newton algorithm converges to a critical point with a superlinear rate. Finally, we validated the algorithm through numerical examples. We compared the approximate Pareto front obtained by the quasi Newton algorithm with the one generated by the weighted sum method. We observed that Algorithm 3.1 (quasi Newton method) successfully generates good approximations of the Pareto fronts in both convex and non convex cases. However, the weighted sum method fails to produce even an approximate front in the non convex case. In addition to this, we have computed performance profiles using number of iterations, function evaluations, Δ spread metric and hypervolume metric. On the basis of performance profile figures that quasi Newton method performs better than weighted sum method in most cases.

In this work, we were focused on finding the solution to an uncertain multiobjective optimization problem under a finite uncertainty set. The solution of an uncertain multiobjective optimization problem under an infinite uncertainty set is left for future scope.

Acknowledgment:

This research is supported by Govt. of India CSIR fellowship, Program No. 09/1174(0006)/2019-EMR-I, New Delhi.

Conflict of interest:

The authors declared that they have no conflict of interests.

References

Ansary, M. A. and Panda, G. (2015). A modified quasi-newton method for vector optimization problem. *Optimization*, 64(11):2289–2306.

Ansary, M. A. T. and Panda, G. (2019). A sequential quadratically constrained quadratic programming technique for a multi-objective optimization problem. *Engineering Optimization*, 51(1):22–41.

Ansary, M. A. T. and Panda, G. (2020). A sequential quadratic programming method for constrained multi-objective optimization problems. *Journal of Applied Mathematics and Computing*, 64(1-2):379–397.

Ansary, M. A. T. and Panda, G. (2021). A globally convergent SQCQP method for multiobjective optimization problems. *SIAM Journal on Optimization*, 31(1):91–113.

Bazaraa, M. S. and Goode, J. J. (1982). An algorithm for solving linearly constrained minimax problems. *European Journal of Operational Research*, 11(2):158–166.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). Robust optimization. Princeton and Oxford: Princeton University Press.

Bhaskar, V., Gupta, S. K., and Ray, A. K. (2000). Applications of multiobjective optimization in chemical engineering. *Reviews in chemical engineering*, 16(1):1–54.

Bonnel, H., Iusem, A. N., and Svaiter, B. F. (2005). Proximal methods in vector optimization. *SIAM Journal on Optimization*, 15(4):953–970.

Ceng, L., Mordukhovich, B. S., and Yao, J.-C. (2010). Hybrid approximate proximal method with auxiliary variational inequality for vector optimization. *Journal of optimization theory and applications*, 146(2):267–303.

Cruz, J. B., Pérez, L. L., and Melo, J. (2011). Convergence of the projected gradient method for quasiconvex multiobjective optimization. *Nonlinear Analysis: Theory, Methods & Applications*, 74(16):5268–5273.

Davidon, W. Variable metric method for minimization, argonne national labo-ratory report anl-5990, november 1959. *DavidonVariable Metric Method for Minimization1959*.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2002). Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 1, pages 825–830. IEEE.

Dhara, A. and Dutta, J. (2011). Optimality conditions in convex optimization: a finite-dimensional view. Boca Raton: CRC Press; 2011.

Drummond, L. G. and Iusem, A. N. (2004). A projected gradient method for vector optimization problems. *Computational Optimization and applications*, 28:5–29.

Drummond, L. G. and Svaiter, B. F. (2005). A steepest descent method for vector optimization. *Journal of computational and applied mathematics*, 175(2):395–414.

Ehrgott, M. (2005). Multicriteria optimization. Berlin, Heidelberg: Springer.

Ehrgott, M., Ide, J., and Schöbel, A. (2014). Minmax robustness for multi-objective optimization problems. *European Journal of Operational Research*, 239(1):17–31.

Eichfelder, G. Adaptive scalarization methods in multiobjective optimization (vector optimization). 2008. Berlin, Heidelberg: Springer.

Fazzio, N. S. and Schuverdt, M. L. (2019). Convergence analysis of a nonmonotone projected gradient method for multiobjective optimization problems. *Optimization Letters*, 13:1365–1379.

Fletcher, R. and Powell, M. J. (1963). A rapidly convergent descent method for minimization. *The computer journal*, 6(2):163–168.

Fliege, J., Drummond, L. G., and Svaiter, B. F. (2009). Newton's method for multiobjective optimization. *SIAM Journal on Optimization*, 20(2):602–626.

Fliege, J. and Svaiter, B. F. (2000). Steepest descent methods for multicriteria optimization. *Mathematical methods of operations research*, 51:479–494.

Fukuda, E. H. and Drummond, L. G. (2011). On the convergence of the projected gradient method for vector optimization. *Optimization*, 60(8-9):1009–1021.

Fukuda, E. H. and Drummond, L. M. G. (2014). A survey on multiobjective descent methods. *Pesquisa Operacional*, 34:585–620.

Fukuda, E. H. and Graña Drummond, L. (2013). Inexact projected gradient method for vector optimization. *Computational Optimization and Applications*, 54:473–493.

Gass, S. and Saaty, T. (1955). The computational algorithm for the parametric objective function. *Naval research logistics quarterly*, 2(1-2):39–45.

Gonçalves, M. L. and Prudente, L. (2020). On the extension of the hager–zhang conjugate gradient method for vector optimization. *Computational Optimization and Applications*, 76(3):889–916.

Göpfert, A. and Nehse, R. (1990). Vektoroptimierung: theorie, verfahren und anwendungen. Leipzig, Germany: Teubner.

Kumar, S., Ansary, M. A. T., Mahato, N. K., and Ghosh, D. (2024). Steepest descent method for uncertain multiobjective optimization problems under finite uncertainty set. *Applicable Analysis*, pages 1–22.

Kumar, S., Ansary, M. A. T., Mahato, N. K., Ghosh, D., and Shehu, Y. (2023). Newton's method for uncertain multiobjective optimization problems under finite uncertainty set. *Journal of Nonlinear and Variational Analysis*, 7(5):785–809.

Kuroiwa, D. and Lee, G. M. (2012). On robust multiobjective optimization. *Vietnam J. Math*, 40(2-3):305–317.

Lai, K. K., Mishra, S. K., and Ram, B. (2020). On q-quasi-newton's method for unconstrained multiobjective optimization problems. *Mathematics*, 8(4):616.

Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. (2002). Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary computation*, 10(3):263–282.

Luc, D. T. (1988). Theory of vector optimization. Berlin: Springer.

Lucambio Pérez, L. and Prudente, L. (2018). Nonlinear conjugate gradient methods for vector optimization. *SIAM Journal on Optimization*, 28(3):2690–2720.

Mahdavi-Amiri, N. and Salehi Sadaghiani, F. (2020). A superlinearly convergent nonmonotone quasi-newton method for unconstrained multiobjective optimization. *Optimization Methods and Software*, 35(6):1223–1247.

Mäkelä, M. M., Eronen, V.-P., and Karmitsa, N. (2014). On nonsmooth multiobjective optimality conditions with generalized convexities. *Optimization in Science and Engineering: In Honor of the 60th Birthday of Panos M. Pardalos*, pages 333–357.

Miettinen, K. (1999). Nonlinear multiobjective optimization. New York: Springer.

Morovati, V., Basirzadeh, H., and Pourkarimi, L. (2018). Quasi-newton methods for multiobjective optimization problems. *4OR*, 16:261–294.

Nocedal, J. and S.T., W. (2006). Numerical optimization. New York: Springer.

Povalej, Ž. (2014). Quasi-newton's method for multiobjective optimization. *Journal of Computational and Applied Mathematics*, 255:765–777.

Qu, S., Goh, M., and Chan, F. T. (2011). Quasi-newton methods for solving multiobjective optimization. *Operations Research Letters*, 39(5):397–399.

Soyster, A. L. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations research*, 21(5):1154–1157.

Stewart, T., Bandte, O., Braun, H., Chakraborti, N., Ehrgott, M., Göbelt, M., Jin, Y., Nakayama, H., Poles, S., and Di Stefano, D. (2008). Real-world applications of multiobjective optimization. *Multiobjective optimization: interactive and evolutionary approaches*, pages 285–327.

Sun, W. and Yuan, Y.-X. (2006). Optimization theory and methods: nonlinear programming. New York: Springer.

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., Tiwari, S., et al. (2008). Multiobjective optimization test instances for the cec 2009 special session and competition.

A Details of test problems

Appendix A: Details of test problems

(TP1) $F : \mathbb{R}^2 \to \mathbb{R}^2$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2\}$, $j = 1, 2$, where $\xi^i = (\xi^i_1, \xi^i_2)$ and $\xi^1 = (1, 2)$, $\xi^2 = (2, 3)$.

$$\zeta_1(x, \xi^i) = \frac{1}{4}(x_1 - \xi^i_1)^4 + 2(x_2 - \xi^i_2)^4 \quad i = 1, 2$$
$$\text{and} \quad \zeta_2(x, \xi^i) = (\xi^i_1 x_2 - \xi^i_2 x_1^2)^2 + (1 - \xi^i_1 x_1)^2, \ i = 1, \ 2.$$

(TP2) $F : \mathbb{R}^2 \to \mathbb{R}^3$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2, 3\}$, $j = 1, 2, 3$ where $\xi^i = (\xi^i_1, \xi^i_2)$ and $\xi^1 = (5, 3)$, $\xi^2 = (5, 6)$, $\xi^3 = (4, 1)$.

$$\zeta_1(x, \xi^i) = x_1^2 + \xi^i_1 x_2^4 + \xi^i_1 \xi^i_2 x_1 x_2,$$
$$\zeta_2(x, \xi^i) = 5x_1^2 + \xi^i_1 x_2^2 + \xi^i_2 x_1^4 x_2,$$
$$\zeta_3(x, \xi^i) = e^{-\xi^i_1 x_1 + \xi^i_2 x_2} + x_1^2 - \xi^i_1 x_2^2.$$

(TP3) $F : \mathbb{R}^2 \to \mathbb{R}^3$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2, 3\}$, $j = 1, 2, 3$ where $\xi^i = (\xi^i_1, \xi^i_2)$ and $\xi^1 = (4, 1)$, $\xi^2 = (5, 2)$, $\xi^3 = (6, 4)$.

$$\zeta_1(x, \xi^i) = 100\xi^i_1(x_2 - x_1^2)^2 + \xi^i_2(1 - x_1)^2,$$
$$\zeta_2(x, \xi^i) = (x_2 - \xi^i_1)^2 + \xi^i_2 x_1^2,$$
$$\zeta_3(x, \xi^i) = \xi^i_1 x_1^2 + 3\xi^i_2 x_2^2.$$

(TP4) $F : \mathbb{R}^2 \to \mathbb{R}^2$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2\}$ $j = 1, 2$, where $\xi^1 = (1, 3)^T$, $\xi^2 = (3, 1)^T$. Note that $\zeta_1(x, \xi^1) = (x_1 - 1)^2 + (x_2 - 3)^2$, and $\zeta_1(x, \xi^2) = (x_1 - 3)^2 + (x_2 - 1)^2$, $\zeta_2(x, \xi^1) = x_1^2 + 3x_2^2$, and $\zeta_2(x, \xi^2) = 3x_1^2 + x_2^2$.

(TP5) $F : \mathbb{R} \to \mathbb{R}^2$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2\}$ $j = 1, 2$, where $\xi^1 = -1$, $\xi^2 = 3$, $\zeta_1(x, \xi^i) = (x - \xi^i)^2$ and $\zeta_2(x, \xi^i) = x^2 + \xi^i x$ for $i = 1, 2$.

(TP6) $F : \mathbb{R}^3 \to \mathbb{R}^2$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2, 3\}$, $j = 1, 2$. Here $\xi^i = (\xi^i_1, \xi^i_2, \xi^i_3)$ and therefore $\xi^1 = (1, 1, 1)^T$, $\xi^2 = (1, -1, 1)^T$, $\xi^3 = (1, -2, 2)^T$,

$$\zeta_1(x, \xi^i) = \quad 1 - e^{-\sum_{j=1}^3 \xi^j_i \left(x_j - \frac{1}{\sqrt{3}}\right)^2} \quad i = 1, 2, 3$$
$$\text{and} \quad \zeta_2(x, \xi^i) = \quad 1 - e^{-\sum_{j=1}^3 \xi^j_i \left(x_j + \frac{1}{\sqrt{3}}\right)^2} \quad i = 1, 2, 3.$$

(TP7) $F : \mathbb{R} \to \mathbb{R}^2$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2\}$ $j = 1, 2$, where $\xi^1 = -4$, $\xi^2 = 7$, $\zeta_1(x, \xi^i) = (x - \xi^i)^2$ and $\zeta_2(x, \xi^i) = x^2 + \xi^i x$ for $i = 1, 2$.

(TP8) $F : \mathbb{R}^2 \to \mathbb{R}^2$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2\}$ $j = 1, 2$, where $\xi^1 = (1, 3)^T$, $\xi^2 = (3, 1)^T$. Note that $\zeta_1(x, \xi^1) = (x_1 - 1)^2 + (x_2 - 3)^2$, and $\zeta_1(x, \xi^2) = (x_1 - 3)^2 + (x_2 - 1)^2$, $\zeta_2(x, \xi^1) = x_1^2 + 3x_2^2$, and $\zeta_2(x, \xi^2) = 3x_1^2 + x_2^2$.

(TP9) $F : \mathbb{R}^3 \to \mathbb{R}^2$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2, 3\}$, $j = 1, 2$. Here $\xi^i = (\xi^i_1, \xi^i_2, \xi^i_3)$ and therefore $\xi^1 = (1, 1, 1)^T$, $\xi^2 = (1, -1, 1)^T$, $\xi^3 = (1, -2, 2)^T$,

$$\zeta_1(x, \xi^i) = \quad 1 - e^{-\sum_{j=1}^3 \xi^j_i \left(x_j - \frac{1}{\sqrt{3}}\right)^2} \quad i = 1, 2, 3$$
$$\text{and} \quad \zeta_2(x, \xi^i) = \quad 1 - e^{-\sum_{j=1}^3 \xi^j_i \left(x_j + \frac{1}{\sqrt{3}}\right)^2} \quad i = 1, 2, 3.$$

(TP10) $F : \mathbb{R}^3 \to \mathbb{R}^3$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2, 3\}$ $j = 1, 2, 3$, where $\xi^i = (\xi^i_1, \xi^i_2, \xi^i_3)$ and $\xi^1 = (1, 1, 1)^T$, $\xi^2 = (1, -1, 1)^T$, $\xi^3 = (1, -2, 2)^T$.

$$\begin{aligned} \zeta_1(x, \xi^i) &= \left(1 + \xi^i_3 x_3\right)\left(\xi^i_1 \xi^i_2 x^3_1 x^3_2 - 10\xi^i_1 x_1 - 4\xi^i_2 x_2\right) \quad i = 1, 2, 3 \\ \zeta_2(x, \xi^i) &= \left(1 + \xi^i_3 x_3\right)\left(\xi^i_1 \xi^i_2 x^3_1 x^3_2 - 10\xi^i_1 x_1 + 4\xi^i_2 x_2\right) \quad i = 1, 2, 3 \\ \text{and} \quad \zeta_3(x, \xi^i) &= \left(1 + \xi^i_3 x_3\right)\xi^i_1 x^2_1 \quad i = 1, 2, 3. \end{aligned}$$

(TP11) $F : \mathbb{R}^2 \to \mathbb{R}^2$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2\}$, $j = 1, 2$, where $\xi^i = (\xi^i_1, \xi^i_2)$ and $\xi^1 = \{(2, 2)$, $\xi^2 = (0, 4)$.

$$\begin{aligned} \zeta_1(x, \xi^i) &= (x_1 - \xi^i_1)^2 + (x_2 + \xi^i_2)^2 \quad i = 1, 2 \\ \text{and} \quad \zeta_2(x, \xi^i) &= (\xi^i_1 x_1 + \xi^i_2 x_2)^2 \quad i = 1, 2. \end{aligned}$$

(TP12) $F : \mathbb{R}^3 \to \mathbb{R}^3$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2, 3\}$, $j = 1, 2, 3$, where $\xi^i = (\xi^i_1, \xi^i_2)$ and $\xi^1 = (4, 1)$ $\xi^2 = (0, 2)$, $\xi^3 = (1, 0)$.

$$\begin{aligned} \zeta_1(x, \xi^i) &= x^2_1 + (x_2 - \xi^i_1)^2 - \xi^i_2 x^2_3 \quad i = 1, 2, 3 \\ \zeta_2(x, \xi^i) &= \xi^i_1 x_1 + \xi^i_2 x^2_2 + x_3 + 4\xi^i_1 \xi^i_2 \quad i = 1, 2, 3 \\ \text{and} \quad \zeta_3(x, \xi^i) &= \xi^i_1 x^2_1 + 6x^2_2 + 25(x_3 - \xi^i_2 x_1)^2. \end{aligned}$$

(TP13) $F : \mathbb{R}^2 \to \mathbb{R}^3$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2, 3\}$, $j = 1, 2, 3$, where $\xi^i = (\xi^1_i, \xi^2_i)$ and $\xi^1 = (2, 3)$ $\xi^2 = (4, 5)$, $\xi^3 = (2, 0)$.

$$\begin{aligned} \zeta_1(x, \xi^i) &= x^2_1 + \xi^i_1 x^4_2 + \xi^i_1 \xi^i_2 x_1 x_2, \quad i = 1, 2, 3 \\ \zeta_2(x, \xi^i) &= 5x^2_1 + \xi^i_1 x^2_2 + \xi^i_2 x^4_1 x_2, \quad i = 1, 2, 3 \\ \text{and} \quad f_3(x, \xi^i) &= e^{-\xi^i_1 x_1 + \xi^i_2 x_2} + x^2_1 - \xi^i_1 x^2_2, \quad i = 1, 2, 3. \end{aligned}$$

(TP14) $F : \mathbb{R} \to \mathbb{R}^2$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2\}$, $j = 1, 2$, where $\xi^1 = -3$ $\xi^2 = 8$.

$$\begin{aligned} \zeta_1(x, \xi^i) &= (x - \xi^i)^2, \quad i = 1, 2 \\ \text{and} \quad \zeta_2(x, \xi^i) &= -x^2 - \xi^i x. \end{aligned}$$

(TP15) $F : \mathbb{R}^2 \to \mathbb{R}^2$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2\}$, $j = 1, 2$, where $\xi^i = (\xi^i_1, \xi^i_2)$ and $\xi^1 = (1, 1)$ $\xi^2 = (0, 2)$.

$$\begin{aligned} \zeta_1(x, \xi^i) &= (x_1 - \xi^i_1)^2 + (x_2 + \xi^i_2)^2, \quad i = 1,\ 2 \\ \text{and} \quad \zeta_2(x, \xi^i) &= (\xi^i_1 x_1 + \xi^i_2 x_2)^2 \quad i = 1,\ 2. \end{aligned}$$

(TP16) $F : \mathbb{R}^3 \to \mathbb{R}^3$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2, 3\}$, $j = 1, 2, 3$, where $\xi^i = (\xi^i_1, \xi^i_2)$ and $\xi^1 = (5, 4)$ $\xi^2 = (0, 8)$, $\xi^3 = (4, 0)$.

$$\begin{aligned} \zeta_1(x, \xi^i) &= 3x^2_1 + (x_2 - \xi^i_1)^2 + \xi^i_2 x^2_3 \quad i = 1, 2, 3 \\ \zeta_2(x, \xi^i) &= 2\xi^i_1 x_1 + \xi^i_2 x^2_2 + 3x_3 + 4\xi^i_1 \xi^i_2 \quad i = 1, 2, 3 \\ \text{and } \zeta_3(x, \xi^i) &= \xi^i_1 x^2_1 + 6x^2_2 + 20(x_3 - \xi^i_2 x_1)^2) \quad i = 1, 2, 3. \end{aligned}$$

(TP17) $F : \mathbb{R}^2 \to \mathbb{R}^3$ is defined as $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2, 3\}$, $j = 1, 2, 3$, where $\xi^1 =$

$(3.0, 3.0, \ldots, 3.0)^T$, $\xi^2 = (6.0, 6.0, \ldots, 6.0)^T$, and $\xi^3 = (9.0, 9.0, \ldots 9.0)^T$. Define

$$\begin{aligned} I_1 &= \{k \in \{2, 3, \ldots, n\} | k \mod 2 = 1\} \\ I_1 &= \{k \in \{2, 3, \ldots, n\} | k \mod 2 = 0\} \\ \zeta_1(x, \xi^i) &= x_1 + \frac{2}{|I_1|} \sum_{k \in I_1} \left(x_k - \sin(\xi_k^i \pi x_1 + \frac{k\pi}{n}) \right)^2 \quad i = 1, 2, 3 \\ \zeta_2(x, \xi^i) &= 1 - \sqrt{x_1} + \frac{2}{I_2} \sum_{k \in I_2} \left(x_k - \sin(\xi_k^i \pi x_1 + \frac{k\pi}{n}) \right)^2 \quad i = 1, 2, 3. \end{aligned}$$

Note: Test problem TP17 is formulated using CEC09_1 in Zhang et al. (2008).

(TP18) $F : \mathbb{R}^2 \to \mathbb{R}^2$ is defined by $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2\}$, $j = 1, 2$.

$$\begin{aligned} \zeta_1 = \max\{\zeta_1(x, \xi^1), \zeta_1(x, \xi^2)\} = \max\{1 + x_1^{\frac{1}{4}}, 1 + x_2^2\} \\ F_2 = \max\{\zeta_2(x, \xi^1), \zeta_2(x, \xi^2)\} = \left\{ 1 - \left(\frac{x_1}{1 + x_1^{\frac{1}{4}}} \right)^2, 1 - \left(\frac{x_1}{1 + x_2} \right)^2 \right\} \end{aligned}$$

(TP19) $F : \mathbb{R}^n \to \mathbb{R}^m$ is defined by $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2, 3\}$, $j = 1, 2, \ldots, m$, where $\xi^1 = (0.25, 0.25, \ldots, 0.25)^T \in \mathbb{R}^n$, $\xi^2 = (0.5, 0.5, \ldots, 0.5)^T \in \mathbb{R}^n$, $\xi^3 = (0.75, 0.75, \ldots, 0.75)^T \in \mathbb{R}^n$. Define

$$gx_i = 100 \left(K + \sum_{k=m}^{n} \left((x_i - \xi_k^i)^2 - \cos(20\pi(x_k - \xi_k^i)) \right) \right)$$

for $i = 1, 2, 3$ where $K = m + n - 1$. $\zeta_j(x, \xi^i)$ for $j = 1, 2, ..., m$ are defined as

$$\begin{aligned} \zeta_1(x, \xi^i) &= 0.5(1 + gx_i) \prod_{k=1}^{m-1} x_k \\ \zeta_j(x, \xi^i) &= 0.5(1 + gx_i) \prod_{k=1}^{m-j} x_k (1 - x_{m-j+1}) \quad j = 2, \ldots, m \end{aligned}$$

(TP20) $F : \mathbb{R}^n \to \mathbb{R}^m$ is defined by $F_j(x) = \max\{\zeta_j(x, \xi^i) : i = 1, 2, 3\}$, $j = 1, 2, \ldots, m$, where $\xi^1 = (0.4, 0.4, \ldots, 0.4)^T \in \mathbb{R}^n$, $\xi^2 = (0.5, 0.5, \ldots, 0.5)^T \in \mathbb{R}^n$, $\xi^3 = (0.6, 0.6, \ldots, 0.6)^T \in \mathbb{R}^n$. Define

$$gx_i = \sum_{k=m}^{n} (x_i - \xi_k^i)^2$$

for $i = 1, 2, 3$. $f_j(x, \xi^i)$ for $j = 1, 2, ..., m$ are defined as

$$\begin{aligned} \zeta_1(x, \xi^i) &= (1 + gx_i) \prod_{k=1}^{m-1} \cos(0.5\pi x_k) \\ \zeta_j(x, \xi^i) &= 0.5(1 + gx_i) \prod_{k=1}^{m-j} \cos(0.5\pi x_k) \sin(0.5\pi x_{m-k+1}) \quad j = 2, \ldots, m \end{aligned}$$

Note: TP19 and TP20 are are constructed using the test problems DTLZ1 and DTLZ2 respectively from Deb et al. (2002).