Appl. Set-Valued Anal. Optim. 5 (2023), No. 2, pp. 291-321
Available online at http://asvao.biemdas.com
https://doi.org/10.23952/asvao.5.2023.2.12

QUASI-NEWTON METHODS FOR MULTIOBJECTIVE OPTIMIZATION PROBLEMS: A SYSTEMATIC REVIEW

K. KUMAR[1], D. GHOSH[1,*], A. UPADHAYAY[1,2], J.C. YAO[3], X. ZHAO[4]

[1]*Department of Mathematics, Indian Institute of Technology (BHU), Varanasi, Uttar Pradesh, 221 005, India*
[2]*Department of Mathematics, Bareilly College, Bareilly, Uttar Pradesh 243005, India*
[3]*Centre for General Education, China Medical University, Taichung 40402, Taiwan*
[4]*School of Mathematical Sciences, Tiangong University, Tianjin 300387, China*

Dedicated to Henry Wolkowicz on the occasion of his 75th birthday

Abstract. Quasi-Newton method is one of the most popular methods for solving unconstrained single and multiobjective optimization problems. In a quasi-Newton method, the search direction is computed based on a quadratic model of the objective function, where some approximations replace the true Hessian at each iteration. Several Hessian approximation schemes with an adequate line search technique provided higher-order accuracies in approximating the curvature and made the methods more effective in terms of convergence to solution. Considering the aforementioned reasons, we write a survey on quasi-Newton methods for multiobjective optimization problems. We discuss the development of all the variants of the quasi-Newton method for multiobjective optimization problems, along with some of the advantages and disadvantages of the existing methods. We give a brief discussion about the line search for these variants too. Two cases have been considered for BFGS, Huang BFGS, and self-scaling BFGS multiobjective versions of quasi-Newton methods: one is in the presence of the Armijo line search, and the other is in the absence of any line search. Subsequently, a nonmonotone line search version is also explained for multiobjective optimization problems. Commentary is given on the convergence properties of these methods. The rate of convergence of all these methods is highlighted. To prove the convergence of every method, it is reported that every sequence of points generated from the method converges to a critical point of the multiobjective optimization problem under some mild assumptions. Based on the available numerical data, we provide an unbiased opinion on the effectiveness of quasi-Newton methods for multiobjective optimization problems.

Keywords. BFGS; H-BFGS; Multiobjective optimization; Performance profiles; Quasi-Newton method; SS-BFGS.
2020 Mathematics Subject Classification. 90C29, 90C53, 58E17.

*Corresponding author.
E-mail addresses: krishankumar.rs.mat19@itbhu.ac.in (K. Kumar), debdas.mat@iitbhu.ac.in (D. Ghosh), ashutosh.upadhayay.rs.mat18@itbhu.ac.in (A. Upadhayay), yaojc@mail.cmu.edu.tw (J.C. Yao), zhaoxiaopeng.2007@163.com (X. Zhao)
Received January 6, 2023; Accepted March 11, 2023.

1. INTRODUCTION

Many problems from engineering [1–3], management [4,5], environmental analysis [6], optimal control [7,8], game theory [9–11], etc., involve multiple conflicting criteria. These problems can be modeled as multiobjective optimization problems (MOPs). For instance, let us consider simple examples of single and multiobjective optimization problems.

- To find the topper(s) (in terms of the highest total of marks obtained in all subjects) from a class of students is a single objective optimization problem, where the objective function provides the total of marks of a student obtained in all subjects.
- To find the topper(s) from a class of students based on several criteria, namely, in terms of number of attendance, the total of marks obtained in all subjects, number of winning debate competitions, marks in a particular subject, etc., is an MOP, where the objective function is taken a multivalued function whose component functions consist number of attendance, a total of marks obtained in all subjects, number of winning debate competitions, marks in a particular subject, etc.

It can be noticed that comparing a solution to a single objective optimization problem is easy, while it is not so easy for MOPs. Therefore, the conventional concept of optimality has to be replaced by the concept of Pareto optimality or efficiency [12]. In this concept, we say a point is Pareto optimal or efficient if there does not exist another point with the same or smaller objective function values.

To find the Pareto optimal point of an MOP, a few approaches have been introduced. One of them is the scalarization technique, or decomposition [12–15]. In this technique, an MOP is converted into a single objective optimization subproblem, which can be solved by using conventional optimization techniques. The first scalarization technique is the weighted sum method [16]. In this technique, a single objective function is constructed by taking the weighted sum of component functions of the objective function of MOP. After that, the problem is solved repeatedly by using different weights to approximate the non-dominated solution set. However, if the Pareto surface is nonconvex, then not all solutions on the trade-off surface can be found, which is a drawback of the weighted sum method [17]. This drawback has been removed in the adaptive weighted sum method [17]. This method behaves the same as the weighted sum method does. The only difference is the additional constraint is added in the adaptive weighted sum method. One of the drawbacks of this method is its reliance on the solutions obtained from the initial weighted sum step. If the weighted sum method cannot find any effective Pareto solutions in the first iteration, this method may not determine the entire non-dominated set. Further, an ε-constraint method, one of the most popular scalarization methods, is developed [18]. In this method, an MOP is converted into a single objective optimization problem by choosing only one component function of the objective function of MOP while the rest of the component functions are considered as constraints. The predefined ε binds these constraints in this method. This method works well. However, the selection of ε sometimes creates problems. Next, a normal boundary intersection method has been proposed [19]. The idea behind this method is that the non-dominated set is related to the boundary of a feasible objective region towards the minimization of objective functions. One of the known drawbacks of the normal boundary intersection method is that the dominated solutions may also occur in the optimal solution set because the algorithm finds a solution regardless of whether the point is dominated or not [20]. In [20], a method named normal constraint has been proposed. This method is a refinement of normal boundary intersection. Furthermore, a direct search domain method is introduced in [13]. This method works on a search domain based on the local linear transformation of objective functions and searches for the solution within each domain. To guarantee a well-distributed Pareto set, it evenly spreads local search domains. In the case of a nonconvex boundary, it is possible not to obtain any feasible solution in the search domain. An ideal cone method is given in [21]. This method is based upon a cone direction and shifting it until a Pareto optimal point is obtained. Besides these, many scalarization techniques have also been given; for instance, see [3, 15, 22–25] and their references.

Apart from the scalarization techniques, many researchers have extended the classical derivative methods to solve MOPs. The most popular method, steepest descent method, has been proposed [26] for multicriteria optimization and then for vector-valued functions [27]. Subsequently, a projected gradient method has been given for constrained MOPs [28]. In [26] and [27], the methods converge to a critical point of the objective function, while in [28], it has been proved that the projected gradient method converges to critical and weak efficient points for nonconvex and convex MOPs, respectively. These two are not the only methods given in the literature. Some more first-order gradient-based methods have been given for continuously differentiable multiobjective functions; for instance, see [29–31] and their references. A Newton method has also been proposed for MOPs [32]. This method [32] has been extended in [33] with majorizing function techniques for MOPs. Like the conventional Newton method, in the method proposed in [32], the Hessian matrix is used as well. However, it is not an easy task to compute the Hessian matrix at every step of the method [34]. Therefore, a quasi-Newton method has been presented for MOPs by using the approximation of the Hessian matrix at every step [35]. The approximation of the Hessian matrix does not make the calculation easier only, but it makes the algorithm faster as well [34]. After that, many variants of the quasi-Newton method have been proposed by several researchers [34, 36–42]. These quasi-Newton methods have been adopted comprehensively in multiobjective optimization. Therefore, we are giving a brief survey on quasi-Newton methods in this paper.

We start with Section 2, in which preliminaries and terminologies are presented. In Section 3 and Section 4, a brief survey is proposed on quasi-Newton methods for smooth and nonsmooth unconstrained MOPs, respectively. Further, a significant discussion, based on the considered quasi-Newton methods, is given in Section 5.

2. Preliminaries and Terminologies

The following notations are used throughout the paper:

- $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{R}_{++}$ denote the set of real numbers, the set of nonnegative real numbers, and the set of positive real numbers, respectively
- $\mathbb{R}^m = \mathbb{R} \times \mathbb{R} \times \cdots \times \mathbb{R}$ (m-times), $\mathbb{R}^m_+ = \mathbb{R}_+ \times \mathbb{R}_+ \times \cdots \times \mathbb{R}_+$ and $\mathbb{R}^m_{++} = \mathbb{R}_{++} \times \mathbb{R}_{++} \times \cdots \times \mathbb{R}_{++}$
- $\text{int}(A)$ denotes the interior of a set A
- $\text{J}F(x)$ and $\mathscr{R}(\text{J}F(x))$ denote the Jacobian of the function F at x and range or image space of $\text{J}F(x)$, respectively
- $\text{conv}(A)$ represents the convex hull of a set A.

Throughout this paper, we take F from $\mathbb{R}^n$ to $\mathbb{R}^m$ defined by

$$F(x) = (f_1(x), f_2(x), \dots, f_m(x))^\top \text{ for all } x \in \mathbb{R}^n.$$

We consider a general unconstrained MOP:

$$\min_{x \in \mathbb{R}^n} F(x), \tag{2.1}$$

where F is a continuously differentiable function.

The solution of MOP (2.1) is defined using partial ordering in $\mathbb{R}^m$. For any $u, v \in \mathbb{R}^m$, $u \leq v$ if and only if $v - u \in \mathbb{R}^m_+$, and $u < v$ if and only if $v - u \in \text{int}(\mathbb{R}^m_+)$. Now we provide the concept of optimality, i.e., Pareto optimality or efficiency.

Definition 2.1. (Pareto optimality [12]). A point $\bar{x}$ is said to be a Pareto optimal or efficient point of MOP (2.1) if there does not exist $x \in \mathbb{R}^n$ such that $F(x) \leq F(\bar{x})$ and $F(x) \neq F(\bar{x})$ hold. Moreover, the point $\bar{x}$ is a weak Pareto optimum if there is no $x \in \mathbb{R}^n$ such that $F(x) < F(\bar{x})$.

Definition 2.2. (Pareto front [12]). Let X^* be the set of all efficient points of MOP (2.1). Then, $F(X^*)$ is said to be the Pareto front of MOP (2.1).

Definition 2.3. (Critical point [26]). A point $\bar{x} \in \mathbb{R}^n$ is said to be a critical point for F if

$$\mathscr{R}(\mathrm{J}F(\bar{x})) \cap (-\mathbb{R}^m_{++}) = \emptyset. \tag{2.2}$$

Remark 2.1. The condition given in (2.2) is necessary for $\bar{x} \in \mathbb{R}^n$ to be a local efficient point of MOP (2.1).

Definition 2.4. (Descent direction [26]). A vector $d \in \mathbb{R}^n$ is said to be a descent direction of F at $\bar{x}$ if

$$\nabla f_j(\bar{x})^\top d < 0 \text{ for all } j \in I = \{1, 2, \dots, m\},$$

where $\nabla f_j(\bar{x})$ is the gradient of f_j at $\bar{x}$.

We shall use set I as given in Definition 2.4 throughout the paper. Next, we start the survey of quasi-Newton methods for MOPs.

3. Quasi-Newton Methods for Unconstrained Smooth MOPs

The first quasi-Newton method for MOPs was proposed by Qu et al. [35] in 2011. In [35], authors have given an algorithm that is an extension of the classical quasi-Newton method for scalar-valued optimization problems. Like the classical quasi-Newton method, this new quasi-Newton method for MOPs has the following two characteristics.

(i) It does not require computing the Hessian.
(ii) It does not require the convex assumption.

Before presenting the main algorithm of a quasi-Newton method for MOPs, authors in [35] have proved a few results. These results, with a brief discussion, are given below.

Proposition 3.1. ([35]). *A point $\bar{x} \in \mathbb{R}^n$ is critical if and only if either one of the following two conditions is satisfied:*

(i) *There does not exist a $d \in \mathbb{R}^n$ such that $f'_j(\bar{x}; d) < 0$ for all $j \in I$, where $f'_j(\bar{x}; d)$ is the directional derivative of f_j at $\bar{x}$ in the direction of d.*
(ii) *In a special case, there also exists at least one $j_o \in I$ such that $f'_{j_o}(\bar{x}) = 0$, where $f'_{j_o}(\bar{x})$ is the derivative of f_{j_o} at $\bar{x}$.*

Note that Proposition 3.1 explains if $\bar{x}$ is a critical point of F, then there is no descent direction for every f_j. In [35], authors proposed the solution approach based upon (i) of Proposition 3.1, which means that the search stops only when no descent direction is found. In the sequel of the discussion on descent direction and critical point, the following corollary is an important result.

Corollary 3.1. ([35]). Let $x \in \mathbb{R}^n$ be a noncritical point. Then, there exists a descent direction $d \in \mathbb{R}^n$ for f_j at x, i.e., $f_j'(x;d) < 0$.

In Newton method, a Newton direction $d_N(x)$ for MOP (2.1) at x is obtained by solving the following subproblem (see [32])

$$\begin{aligned} &\text{minimize } t\\ &\text{subject to } \nabla f_j(x)^\top d + \tfrac{1}{2} d^\top \nabla^2 f_j(x) d - t \le 0,\ j \in I\\ &\qquad\qquad t \in \mathbb{R}, d \in \mathbb{R}^n, \end{aligned}$$

which involves the quadratic approximation of all component functions of the objective function. Similarly, authors in [35] have presented the following subproblem to find a quasi-Newton direction $d_q(x)$ for quasi-Newton method for MOP (2.1),

$$\begin{aligned} &\text{minimize } t\\ &\text{subject to } \nabla f_j(x)^\top d + \tfrac{1}{2} d^\top B_j(x) d - t \le 0,\ j \in I\\ &\qquad\qquad \|d\| \le 1, t \le -\varepsilon, d \in \mathbb{R}^n, \end{aligned} \tag{3.1}$$

where $B_j(x)$ is the Hessian matrix of f_j at x or its approximation obtained by quasi-Newton method.

The method given in [35] is based on the following observation.

Lemma 3.1. ([35]). *Let $B_j(x)$, $j \in I$, and ε be a positive semidefinite matrix and a sufficiently small positive scalar, respectively. Let the feasible set of subproblem (3.1) be nonempty. Then, x is noncritical, and any feasible point $d_\varepsilon(x)$ is a descent direction for F; otherwise, x is a good estimate of the critical point for F.*

After discussing the elementary information, authors in [35] have summarized the details of the quasi-Newton method in Algorithm 1.

Following is an example that is solved by Algorithm 1. The solution set and Pareto front are shown in Fig. 1.

Problem 1 (FON [43]). This problem is stated to minimize:

$$\begin{aligned} f_1(x) &= 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right),\\ f_2(x) &= 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right). \end{aligned}$$

In this problem, we take the number of decision variables x_i are 2 and $n = 2$. Each decision variable x_i is in $[-1, 1]$.

Algorithm 1 Quasi-Newton Algorithm for Multiobjective Optimization Proposed in [35]

Step 0 (Initial step)
Initially choose x^0, small positive scalar ε, constant $\alpha^0 \in (0,1)$, $\beta \in (0,\frac{1}{2})$ and a positive definite initial matrix $B_j(x^0)$ for all $j \in I$.
Set iteration counter $k = 0$.

Step 1 (Generation of search direction $d_q(x^k)$ and $\tau_q(x^k)$)
Solve subproblem (3.1) at x^k. The solution $d_q(x^k)$ of subproblem (3.1) at x^k is search direction. The optimal objective function value of subproblem (3.1) at x^k is $\tau_q(x^k)$. If subproblem (3.1) is infeasible, then terminate.

Step 2 (Line search)
Compute the step length α^k using the Armijo condition

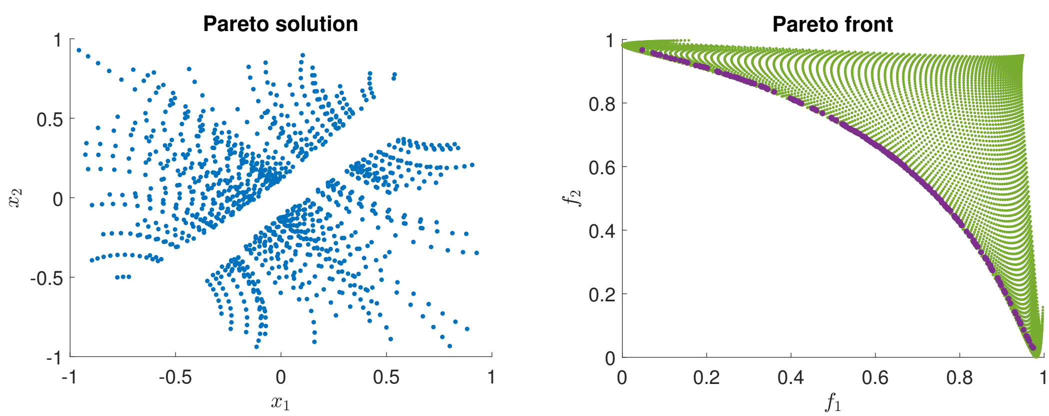
$$f_j(x^k + \alpha^k d_q(x^k)) \leq f_j(x^k) + \beta\alpha^k \tau_q(x^k),\ j \in I.$$

Step 3 (Update of iteration point and Hessian approximations)
Set $x^{k+1} = x^k + \alpha^k d_q(x^k)$ and update the Hessian approximation matrices $B_j(x^k)$ by

$$B_j(x^{k+1}) = \begin{cases} B_j(x^k) - \frac{B_j(x^k)s^k {s^k}^\top B_j(x^k)}{{s^k}^\top B_j(x^k)s^k} + \frac{y_j^k {y_j^k}^\top}{{s^k}^\top y_j^k}, & \text{if } k \in \tilde{K}_j \\ B_j(x^k), & \text{otherwise,} \end{cases}$$

where $\tilde{K}_j = \{k : {y_j^k}^\top s^k \geq \varepsilon \min\{-\tau_q(x^k), 1\}\}$, $s^k = x^{k+1} - x^k$ and $y_j^k = \nabla f_j(x^{k+1}) - \nabla f_j(x^k)$, for each $j \in I$.

Step 4 (Update of iteration counter k)
Set $k = k+1$, and go to **Step 1**.



(A) (B)

FIGURE 1. (A) Pareto solution obtained by Algorithm 1, (B) Feasible region (green dots) and the Pareto critical points (purple dots) obtained by the Algorithm 1 using 200 randomly generated starting points for FON problem

Algorithm 1 converges to a critical point of F under the basic assumptions, which are as follows (see [35]).

A1.1 The level set $S_o = \{x \in \mathbb{R}^n : F(x) \leq F(x_o)\}$ is bounded.

A1.2 For sufficient large k, the step length $\alpha^k = 1$ is accepted.

Remark 3.1. The idea of assumption A1.1 has come from the Newton-type methods. It is commonly used for proving the convergence of scalar and multiobjective optimization problems. Subsequently, assumption A1.2 is taken from the trust region methods in which $\|d\| \leq \Delta$ is replaced by $\|d\| \leq 1$, where Δ is trust region radius.

Next, a convergence theorem for Algorithm 1 is given below.

Theorem 3.1. ([35]). *Suppose that there exists a constant c such that $\|B_j(x)\| \leq c$, for any $x \in S_o$ and $j \in I$. Then, every accumulation point of the sequence $\{x^k\}$ generated by* Algorithm 1 *is critical for F under the assumptions* A1.1 *and* A1.2.

The new quasi-Newton method was proposed by Povalej [34] in 2014. Povalej has given a method for strongly convex unconstrained MOPs. In this method also Hessian matrix is approximated by well known BFGS method [44]. We present an MOP below, which is considered in [34]

$$\min_{x \in \mathbb{R}^n} G(x), \tag{3.2}$$

where $G = (g_1, g_2, \ldots, g_m)^\top : \mathbb{R}^n \to \mathbb{R}^m$ is a strongly convex continuously differentiable multiobjective function.

Throughout this paper, we take G and $g_j, j \in I$ as in MOP (3.2).

Before analyzing the new algorithm, a theorem is presented below, which plays a crucial role in the analysis of the quasi-Newton method proposed in [34].

Theorem 3.2. ([34]). *Let $\bar{x} \in \mathbb{R}^n$ be a stationary point of G. Then, $\bar{x}$ is a weak Pareto optimum. Moreover, if G is twice continuously differentiable and $0 \prec \nabla^2 g_j(x)$ for all $x \in \mathbb{R}^n$ and $j \in I$, then if $\bar{x}$ is a stationary point of G, then $\bar{x}$ is a Pareto optimum.*

In [35], the quasi-Newton direction has been calculated by solving the subproblem (3.12). This problem is based on the trust region method. However, in [34], authors have given a new type subproblem to find the quasi-Newton direction. This is easier to solve and comprehensively used by researchers. We present a brief description of it below.

Analogous to Newton's direction in the Newton method for MOPs, the direction $d(x)$ is a quasi-Newton direction for $x \in \mathbb{R}^n$ if $d(x)$ is the optimal solution of the problem

$$\min_{d \in \mathbb{R}^n} \max_{j \in I} \nabla g_j(x)^\top d + \tfrac{1}{2} d^\top B'_j(x) d, \tag{3.3}$$

where $B'_j(x)$ is some approximation of $\nabla^2 g_j(x), j \in I$. In [34], this approximation matrix is calculated by the BFGS update formula (3.8). Let $\tau(x)$ and $d(x)$ be such that

$$\begin{aligned} \tau(x) &= \min_{d \in \mathbb{R}^n} \max_{j \in I} \nabla g_j(x)^\top d + \tfrac{1}{2} d^\top B'_j(x) d, \\ d(x) &= \operatorname*{argmin}_{d \in \mathbb{R}^n} \max_{j \in I} \nabla g_j(x)^\top d + \tfrac{1}{2} d^\top B'_j(x) d. \end{aligned} \tag{3.4}$$

Sometimes, it is difficult to solve the minimax problem (subproblem (3.3)). Therefore, a very nice convex quadratic optimization problem, which is equivalent to a subproblem (3.3), is formulated in [34]. This problem is as follows:

$$\begin{aligned} &\text{minimize } t \\ &\text{subject to } \nabla g_j(x)^\top d + \tfrac{1}{2} d^\top B'_j(x) d - t \le 0, \; j \in I \\ &\qquad\qquad t \in \mathbb{R}, d \in \mathbb{R}^n. \end{aligned} \tag{3.5}$$

The benefit of (3.5) is that one can solve it by using KKT conditions. The solution is given in [34], which is

$$d(x) = -\left(\sum_{j=1}^{m} \lambda_j(x) B'_j(x)\right)^{-1} \left(\sum_{j=1}^{m} \lambda_j(x) \nabla g_j(x)\right),$$

where $\lambda_j(x), j \in I$ are the KKT multipliers. Note that the algorithm in [35] terminates if subproblem (3.1) is infeasible. However, this is not the case in [34]. To find the terminating condition, a lemma is given in [34] that connects the stationarity of a point x with $d(x)$ and $\tau(x)$.

Lemma 3.2. (Stationarity [34]). *Let $B'_j(x)$ be a positive definite matrix for all $x \in \mathbb{R}^n$ and consider τ as defined in (3.4). Then,*

(i) *for all $x \in \mathbb{R}^n, \tau(x) \le 0$.*
(ii) *the following conditions are equivalent.*
  (a) *The point x is not stationary.*
  (b) $\tau(x) < 0$.
  (c) $d(x) \neq 0$.
(iii) *the function τ is continuous.*

After defining the descent direction $d(x)$, the next step is to choose step length α. In [34], the author has used the classical Armijo condition for the scalar-valued function $g_j, j \in I$. The Armijo condition for the quasi-Newton search direction $d(x)$ is given by

$$g_j(x + \alpha d(x)) \le g_j(x) + \rho \alpha \nabla g_j(x)^\top d(x), \tag{3.6}$$

where $\rho \in (0,1)$. The condition (3.6) is redefined in [34] to accept a full quasi-Newton step, i.e., $\alpha = 1$ close to a local minimum. The redefined Armijo condition is

$$g_j(x + \alpha d(x)) \le g_j(x) + \tfrac{1}{2} c \alpha \nabla g_j(x)^\top d(x), j \in I, \tag{3.7}$$

where $c = 2b \in (0,1)$ with $b \in (0, \frac{1}{2})$. To relate Armijo condition (3.7) with the stationarity of a point $x \in \mathbb{R}^n$ for G, a theorem is given in [34].

Theorem 3.3. ([34]). *Let $x \in \mathbb{R}^n$ be a nonstationary point of G. Then, for any $0 < c < 1$ there exists $\alpha^0 \in (0,1]$ such that (3.7) holds for all $\alpha \in [0, \alpha^0]$ and $j \in I$.*

Since the direction and step length have been discussed for the quasi-Newton method proposed in [34], we present the algorithm (Algorithm 2) for this method below. One thing we want to mention here is that as we have given a complete step-by-step algorithm (Algorithm 1), from now onward, we shall mention only those steps which are different from the standard algorithm, Algorithm 1.

Algorithm 2 Quasi-Newton Algorithm for Multiobjective Optimization Proposed in [34]

Step 1 (Generation of search direction $d(x^k)$ and $\tau(x^k)$)
Solve subproblem (3.3) at x^k. The solution $d(x^k)$ of subproblem (3.3) at x^k is a search direction. The optimal objective function value of subproblem (3.3) at x^k is $\tau(x^k)$. If $\tau(x^k) = 0$, then stop.

Step 2 (Line search)
Compute the step length α^k using the Armijo condition

$$g_j(x^k + \alpha^k d(x^k)) \leq g_j(x^k) + \beta\alpha^k\tau(x^k), \; j \in I.$$

Step 3 (Update of iteration point and Hessian approximations)
Set $x^{k+1} = x^k + \alpha^k d(x^k)$ and update the Hessian approximation matrices $B'_j(x^k)$ by

$$B'_j(x^{k+1}) = B'_j(x^k) - \frac{B'_j(x^k)s_k s_k^\top B'_j(x^k)}{s_k^\top B'_j(x^k)s_k} + \frac{y_j^k y_j^{k\top}}{s_k^\top y_j^k}, \tag{3.8}$$

where $s_k = x^{k+1} - x^k$ and $y_j^k = \nabla g_j(x^{k+1}) - \nabla g_j(x^k)$ for each $j \in I$.

Following is an example solved by Algorithm 2. The solution set and Pareto front are shown in Fig. 2.

Problem 2 (PNR [45]).This problem is as follows:

$$\text{minimize } (x_1^4 + x_2^4 - x_1^2 + x_2^2 + 20, x_1^2 + x_2^2)^\top,$$
$$\text{subject to } -1 \leq x_i \leq 1 \text{ for } i = 1,2.$$

The number of decision variables x_i in this problem is 2. Each decision variable is in between -1 to 1.

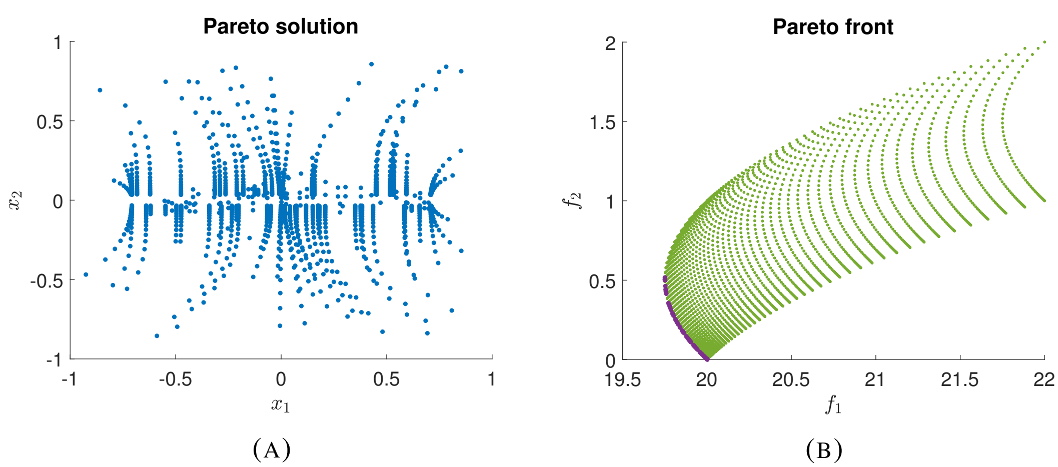


FIGURE 2. (A) Pareto solution obtained by Algorithm 2, (B) Feasible region (green dots) and the Pareto critical points (purple dots) obtained by the Algorithm 2 using 200 randomly generated starting points for PNR problem

The convergence of Algorithm 2 is proved with the help of some lemmas. Below, we first mention these lemmas, and after that, a theorem claims that the convergence of the Algorithm 2 is superlinear.

Lemma 3.3. ([34]). *For $\delta, \varepsilon \in \mathbb{R}_+$ and $x, y \in \mathbb{R}^n$ with $\|y - x\| < \delta$,*

$$\|\nabla^2 g_j(y) - \nabla^2 g_j(x)\| < \tfrac{\varepsilon}{2}, j \in I \tag{3.9}$$

holds. Under the assumption (3.9), we have

$$\|\nabla g_j(y) - (\nabla g_j(x) + \nabla^2 g_j(x)(y - x))\| < \tfrac{\varepsilon}{2}\|y - x\|,$$
$$\text{and } |g_j(y) - (g_j(x) + \nabla g_j(x)^\top (y - x) + \tfrac{1}{2}(y - x)^\top \nabla^2 g_j(y - x))| < \tfrac{\varepsilon}{4}\|y - x\|^2.$$

Lemma 3.4. ([34]). *Let $\{x^k\}$ be a sequence generated by Algorithm 2 and $\{B'_j(x^k)\}, j \in I$, sequence of BFGS updates. For $\varepsilon > 0$, let us assume there exists $k_o \in \mathbb{N}$ such that for all $k_o \leq k$*

$$\frac{\|(\nabla^2 g_j(x^k) - B'_j(x^k))(y - x^k)\|}{\|y - x^k\|} < \tfrac{\varepsilon}{2}, j \in I. \tag{3.10}$$

Under this assumption, for any $x^k, k_o \leq k$ and $y \in \mathbb{R}^n$ such that $\|y - x^k\| < \delta$, we have

$$\|\nabla g_j(y) - (\nabla g_j(x^k) + B_j(x^k)(y - x^k)\| < \varepsilon \|y - x^k\|,$$
$$\text{and } |g_j(y) - (g_j(x^k) + \nabla^k g_j^\top (y - x^k) + \tfrac{1}{2}(y - x^k)^\top B_j(x^k)(y - x^k))| < \tfrac{\varepsilon}{2}\|y - x^k\|^2, j \in I.$$

Lemma 3.5. ([34]). *Let $x \in \mathbb{R}^n$ and $a', b' \in \mathbb{R}_+$ such that $a' \leq b'$. If*

$$aI_{id} \leq B'_j(x) \leq bI_{id}, \; j \in I,$$

where I_{id} is an identity matrix of order n, then

(a) $\frac{a'}{2}\|d(x)\|^2 \leq |\tau(x)| \leq \frac{b'}{2}\|d(x)\|^2$,

(b) $|\tau(x)| \leq \frac{1}{2a'}\left\|\sum_{j=1}^{m} \lambda_j(x) \nabla g_j(x)\right\|^2$ *for all* $0 \leq \lambda_j(x), j \in I$ *with* $\sum_{j=1}^{m} \lambda_j(x) = 1$.

Now we give a theorem to show the convergence of Algorithm 2 is superlinear.

Theorem 3.4. (Superlinear convergence [34]). *Let $\{x^k\}$ be a sequence generated in Algorithm 2. Let us assume $c \in (0,1), k_o \in \mathbb{N}, a', b', r, \delta, \varepsilon \in \mathbb{R}_+, x, y \in V$, where V is an open convex subset of $\mathbb{R}^n$ and*

(i) $a'I_{id} \leq B'_j(x) \leq b'I_{id}, a'I_{id} \leq \nabla^2 g_j(x) \leq b'I_{id}, \; j \in I$,

(ii) $\|\nabla^2 g_j(y) - \nabla^2 g_j(x)\| < \frac{\varepsilon}{2}$ *with* $\|y - x\| < \delta, j \in I$,

(iii) $\frac{\|(\nabla^2 g_j(x^k) - B'_j(x^k))(y - x^k)\|}{\|y - x^k\|} < \frac{\varepsilon}{2}, j \in I$,

(iv) $\frac{\varepsilon}{a'} \leq 1 - c$,

(v) $\|d(x^{k_o})\| < \min\{\delta, r\left(1 - \frac{\varepsilon}{a'}\right)\}$.

Then, for $k_o \leq k$,

(a) $\|x^k - x^{k_o}\| \leq (1 - (\frac{\varepsilon}{a'})^{k-k_o})\|d(x^{k_o})\|$,

(b) $\|d(x^k)\| \leq (\frac{\varepsilon}{a})^{k-k_o}\|d(x^{k_o})\|$,

(c) $\alpha^k = 1$,

(d) $\|d(x^{k+1})\| \leq \frac{\varepsilon}{a'}\|d(x^k)\|$, *and the sequence* $\{x^k\}$ *converges to some local Pareto optimum* $\bar{x} \in \mathbb{R}^n$. *Also, the convergence is superlinear.*

After a significant theory and analysis of the quasi-Newton method for MOPs given in [34], Ansary and Panda [36] have modified this quasi-Newton method. The highlighted points of the method given in [36] are as follows.

- Assumption of convexity of each objective function is not necessary for this method.
- This method computes a common positive definite matrix instead of computing it for every component function of the objective function.
- The algorithm converges not only to critical but also local weak efficient points under some mild assumptions.
- It is also proved that the method does not depend upon the initial point.

The work given in [36] is a modified version of [35] and [34]. Therefore, the fundamental theory is common in these works. Considering this, we directly start from the main modifications and changes provided in [36].

To define the quasi-Newton direction, the authors in [36] have used a common positive definite matrix for each objective function. They have reformulated subproblems (3.3), and (3.5) in terms of a given symmetric positive definite matrix $B(x)$ at x, which are as follows:

$$\min_{d \in \mathbb{R}^n} \max_{j \in I} \nabla f_j(x)^\top d + \tfrac{1}{2} d^\top B(x) d, \tag{3.11}$$

and

$$\begin{aligned} &\text{minimize } \ t \\ &\text{subject to } \nabla f_j(x)^\top d + \tfrac{1}{2} d^\top B(x) d - t \leq 0, \ j \in I \\ &\qquad\qquad\quad t \in \mathbb{R}, d \in \mathbb{R}^n, \end{aligned} \tag{3.12}$$

respectively. Subproblem (3.12) is a convex programming problem. Using KKT optimality conditions, its solution can be found. The KKT optimality conditions for subproblem (3.12) are

$$\sum_{j \in I} \lambda_j (\nabla f_j(x) + B(x) d) = 0, \tag{3.13}$$

$$\sum_{j \in I} \lambda_j = 1, \tag{3.14}$$

$$\lambda_j \geq 0, \nabla f_j(x)^\top d + \tfrac{1}{2} B(x) d \leq t \text{ for all } j \in I, \tag{3.15}$$

$$\lambda_j (\nabla f_j(x)^\top d + \tfrac{1}{2} d^\top B(x) d - t) = 0 \text{ for all } j \in I, \tag{3.16}$$

where $\lambda_j = \lambda_j(x)$. Let $d_{mod}(x)$ satisfy (3.13)-(3.16). Then,

$$d_{mod}(x) = -B(x)^{-1} \sum_{j \in I} \lambda_j \nabla f_j(x). \tag{3.17}$$

To show that this $d_{mod}(x)$ is a descent direction at x, where x is a noncritical point, a theorem has been proved in [36]. We present this theorem below.

Theorem 3.5. ([36]). *Let x be a noncritical point for F. Then, $d_{mod}(x)$ as given in (3.17) is a descent direction at x.*

We still did not mention here how a common positive matrix $B(x)$ is generated instead of having m positive definite matrices for m component functions of the objective function. The next step is how to generate a positive definite matrix at every step. For this, authors in [36] have modified the conventional secant conditions and generated a positive definite matrix at every step. We provide the description of this calculation below.

To make calculation easy, we denote $d^k_{mod} = d_{mod}(x^k), B^k = B(x^k), \nabla f^k_j = \nabla f_j(x^k), f^k_j = f_j(x^k), \lambda^k_j = \lambda_j(x^k)$, where $\lambda_j(x^k)$ are Lagrange multipliers obtained by solving (3.13)-(3.16) at x^k.

Let $x^{k+1} = x^k + \alpha^k d^k_{mod}$ for some $\alpha^k > 0$. Consider the following quadratic forms in x,

$$f^k_j + (x - x^k)^\top \nabla f^k_j + \tfrac{1}{2}(x - x^k)^\top B^k (x - x^k) \text{ for all } j \in I \tag{3.18}$$

and

$$f^{k+1}_j + (x - x^{k+1})^\top \nabla f^{k+1}_j + \tfrac{1}{2}(x - x^{k+1})^\top B^{k+1} (x - x^{k+1}) \text{ for all } j \in I. \tag{3.19}$$

From (3.14), we have $\sum\limits_{j \in I} \lambda^k_j = 1$. Consider the summation over j in (3.18) and (3.19) and let

$$m^k(x) = \sum_{j \in I} \lambda^k_j (f^k_j + (x - x^k)^\top \nabla f^k_j) + \tfrac{1}{2}(x - x^k)^\top B^k (x - x^k)$$

and

$$m^{k+1}(x) = \sum_{j \in I} \lambda^k_j (f^{k+1}_j + (x - x^{k+1})^\top \nabla f^{k+1}_j) + \tfrac{1}{2}(x - x^{k+1})^\top B^{k+1} (x - x^{k+1}).$$

The updated matrix B^{k+1} will be a good approximation of $\sum\limits_{j \in I} \lambda^k_j \nabla^2 f_j(x^{k+1})$ if the following condition holds:

$$\nabla m^k(x^k) = \nabla m^{k+1}(x^k). \tag{3.20}$$

After simplifying (3.20), we have

$$\sum_{j \in I} \lambda^k_j \nabla f^k_j = \sum_{j \in I} \lambda^k_j \nabla f^{k+1}_j + B^{k+1}(x^k - x^{k+1}).$$

Thus,

$$B^{k+1}(x^{k+1} - x^k) = \sum_{j \in I} \lambda^k_j (\nabla f^{k+1}_j - \nabla f^k_j). \tag{3.21}$$

Assuming $\delta^k = x^{k+1} - x^k, \gamma^k_j = \nabla f^{k+1}_j - \nabla f^k_j$ for $j \in I$, $\sum\limits_{j \in I} \lambda^k_j \gamma^k_j = u^k$, then (3.21) becomes $B^{k+1}\delta^k = u^k$, which is a modified secant condition.

Lemma 3.6. ([36]). *If* $\nabla f^{k+1^\top}_j d^k_{mod} \geq c \nabla f^{k^\top}_j d^k_{mod}$*, for some* $c < 1$*, for all* $j \in I$*, then* $u^{k^\top} \delta^k > 0$.

In Lemma 3.6, the condition $u^{k^\top} \delta^k > 0$ has been proved. Therefore, by using the same steps of rank one DFP formula for the conventional quasi-Newton method for the scalar-valued objective problem, the corresponding rank one DFP formula of generating B^k for MOP is defined in [36] by

$$B^{k+1} = (I_{id} - \gamma^k u^k \delta^k) B^k (I_{id} - \gamma^k \delta^k u^k) + \gamma^k u^k u^{k^\top},$$

where $\gamma^k = \frac{1}{u^{k^\top}\delta^k} = \frac{1}{\sum\limits_{j \in I} \lambda^k_j (\gamma^{k^\top}_j \delta^k)}, u^{k^\top}\delta^k > 0$. Similarly, the modified rank two DFP update formula for MOP is given by

$$B^{k+1} = B^k + \frac{u^k u^{k^\top}}{u^{k^\top}\delta^k} - \frac{B^k \delta^k \delta^{k^\top} B^k}{\delta^{k^\top} B^k \delta^k}. \tag{3.22}$$

Remark 3.2. We know that the usual curvature condition in the quasi-Newton method is ${\delta^k}^\top \gamma^k > 0$. However, in [36], γ^k is replaced by u^k in usual curvature condition. Therefore, the sequence $\{B^k\}$ is of positive definite matrices, provided initial matrix B^0 is a positive definite matrix.

Remark 3.3. Note that for convex component function $g_j, j \in I$, the condition proved in Lemma 3.6 is satisfied easily because

$$ {u^k}^\top \delta^k = \sum_{j \in I} \lambda_j^k (\nabla g_j^{k+1} - \nabla g_j^k)^\top \delta^k = \sum_{j \in I} \lambda_j^k {\delta^k}^\top \nabla^2 G_j \left(x^k + \theta \delta^k\right) \delta^k > 0, $$

where $0 < \theta < 1$. Therefore, the sequence $\{B^k\}$ can be generated straightforwardly if component functions are convex. However, Lemma 3.6 claims that $\{B^k\}$ can be generated without using the convexity of component functions.

Now we present the algorithm proposed in [36].

Algorithm 3 First quasi-Newton Algorithm for Multiobjective Optimization Proposed in [36]

Step 1 (Generation of search direction $d_{mod}(x^k)$ and $\tau_{mod}(x^k)$)
Solve subproblem (3.11) at x^k. The solution $d_{mod}(x^k)$ of subproblem (3.11) at x^k is a search direction. The optimal objective function value of subproblem (3.11) at x^k is $\tau_{mod}(x^k)$. If $\tau_{mod}(x^k) = 0$, then stop.

Step 2 (Line search)
Compute the step length α^k using the Armijo condition

$$ f_j(x^k + \alpha^k d_{mod}(x^k)) \leq f_j(x^k) + c\alpha^k \sum_{j \in I} \lambda_j(x^k) \nabla f_j(x^k)^\top d_{mod}(x^k), \ j \in I. \tag{3.23} $$

Step 3 (Update iteration point and Hessian approximation)
Set $x^{k+1} = x^k + \alpha^k d_{mod}(x^k)$ and update Hessian approximation matrix $B(x^k)$ by (3.22).

Following is an example solved by Algorithm 3. The solution set and Pareto front are shown in Fig. 3.

Problem 3 (COMET [46]). This problem is stated to minimize:

$$ \begin{aligned} f_1(x) &= (1 + g(x_3))(x_1^3 x_2^2 - 10x_1 - 4x_2), \\ f_2(x) &= (1 + g(x_3))(x_1^3 x_2^2 - 10x_1 + 4x_2), \\ f_3(x) &= 3(1 + g(x_3))x_1^2, \\ g(x_3) &\geq 0. \end{aligned} $$

This problem is a nonconvex tri-objective optimization problem in which the decision variable x_1 takes the value from 1 to 3.5 and x_2 from -2 to 2. We have chosen $g(x_3) = x_3$ and $0 \leq x_3 \leq 1$. The Pareto optimal surface we obtained from the proposed method corresponds to $\bar{x}_3 = 0$ and for $-2 \leq \bar{x}_1 \bar{x}_2 \leq 2$ with $1 \leq \bar{x}_1 \leq 3.5$.

One of the many properties of the quasi-Newton method given in [36] is that this method converges to the critical and local weak efficient solutions according to some mild assumptions. Considering this, two theorems have been given below.

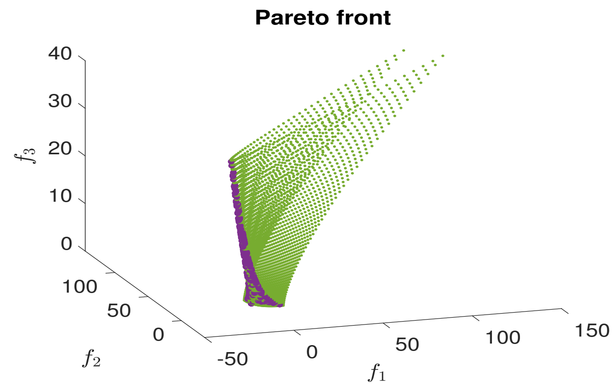


FIGURE 3. Feasible region (green dots) and the Pareto critical points (purple dots) obtained by the Algorithm 3 using 200 randomly generated starting points for Comet problem

Theorem 3.6. ([36]). *Let $x^{k+1} = x^k + \alpha^k d^k_{mod}$ be a sequence generated by Algorithm 3, where $\alpha > 0$ is taken from Armijo line rule and d^k_{mod} is a quasi-Newton direction at iteration k. Moreover, let assumption A1.1 be satisfied and $f_j(x)$ is bounded below for at least one $j \in I$. Then, every accumulation point of the sequence $\{x^k\}$ is a critical point for F.*

Algorithm 3 converges to a critical point for F. However, there is another version of Algorithm 3 given in [36] as well with minor changes in **Step 2**. Therefore, we are just giving **Step 2** of this algorithm below.

Algorithm 4 Second quasi-Newton Algorithm for Multiobjective Optimization Proposed in [36]

Step 2 (Line search)
If $\cos^2(\theta_j(x^k)) > \delta$ for all $j \in I$ and for some $\delta > 0$, where $\theta_j(x^k)$ is the angle between ∇f^k_j and d^k_{mod}, then choose appropriate step length α^k satisfies (3.23) and (3.24). Otherwise, choose α^k satisfies (3.23) only.

Theorem 3.7. ([36]). *Let the assumptions of Theorem 3.6 hold and*

(i)
$$\sum_{j\in I} \lambda^k_j \nabla f^{k+1^\top}_j d^k_{mod} \geq c_2 \sum_{j\in I} \lambda^k_j \nabla f^{k^\top}_j d^k_{mod}, \quad c_1 < c_2 < 1, \tag{3.24}$$

(ii) *∇f_j are Lipschitz continuous for all $j \in I$,*

(iii) *$\cos^2 \theta_j(x^k) \geq \delta$ for some $\delta > 0$ and for all $j \in I$, where $\theta_j(x^k)$ is the angle between ∇f^k_j and d^k_{mod}.*

Then, every accumulation point of $\{x^k\}$ is a locally weak efficient solution for MOP (3.2).

Remark 3.4. The authors in [36] have given two convergence theorems, but they did not make any conclusion on the rate of convergence. However, it is mentioned without any proof in [36] that the modified quasi-Newton method has at least superlinear convergence under some reasonable assumptions. Proving or disproving this statement might be a future problem for interested researchers.

It is easy to notice that in quasi-Newton methods for MOPs, the BFGS method is commonly used to approximate the Hessian matrix of the objective multiobjective function. In [39], new variants of the quasi-Newton method have been proposed using the self-scaling BFGS (SS-BFGS) and the Huang BFGS (H-BFGS) formulae instead of using the conventional BFGS formula. The methods given in [39] are similar to the [34] without the restriction of convexity on the objective function of MOP. The difference is in using the update formula for approximate matrix B_j^k at every iteration. Authors have considered MOP (2.1) in [39] to solve. As mentioned above, in [39], authors have used the following formulae for SS-BFGS and H-BFGS

$$B_j^{k+1} = \frac{{\gamma_j^k}^\top \delta^k}{\delta^\top B_j^k \delta^k}\left(B_j^k - \frac{B_j^k \delta^k {\delta^k}^\top B_j^k}{\delta^\top B_j^k \delta^k}\right) + \frac{\gamma_j^k {\gamma_j^k}^\top}{{\gamma_j^k}^\top \delta^k}, \tag{3.25}$$

and

$$B_j^{k+1} = B_j^k - \frac{B_j^k \delta^k {\delta^k}^\top B_j^k}{\delta^\top B_j^k \delta^k} + \frac{\hat{\gamma}_j^k {\hat{\gamma}_j^k}^\top}{{\hat{\gamma}_j^k}^\top \delta^k}, \text{ respectively,} \tag{3.26}$$

where notations γ_j^k, δ^k are the same as in (3.8) and $\hat{\gamma}_j^k = \gamma_j^k + \frac{\theta_k}{{\delta^k}^\top \gamma_j^k}\gamma_j^k$ with $\theta_k = 6(f^k - f^{k+1}) + 3(\nabla f^k + \nabla f^{k+1})^\top \delta^k$.

In order to find search directions, similar to the previous quasi-Newton methods, the same subproblems along with SS-BFGS and H-BFGS updates of an approximate matrix $B_j(x)$ have been solved in [39]. The next task is to find the step length. Generally, step length is found with the Armijo line search. The authors of [39] have given two algorithms. One is in the presence of Armijo linear search, and the other is in the absence of any line search. We start with the quasi-Newton method with the Armijo line search presented in [39]. This algorithm is very much similar to the algorithm proposed in [34] except for the update of Hessian approximation. The algorithm is as follows:

Algorithm 5 Quasi-Newton Algorithm with Armijo Line Search for Multiobjective Optimization Proposed in [39]

Step 3 (Update of iteration point and Hessian approximations)
Set $x^{k+1} = x^k + \alpha^k d(x^k)$ and update the Hessian approximation matrices $B_j(x^k)$ by (3.8), (3.25), and (3.26).

Now we describe the algorithm without any line search. In this algorithm, there is no update in step length in the line search step. The authors have assumed that $\alpha^k = 1$ at every iteration. So, the updated point at kth iteration is $x^{k+1} = x^k + d(x^k)$.

Algorithm 6 Quasi-Newton Algorithm without any Line Search for Multiobjective Optimization Proposed in [39]

Step 2 (Line search)
Take $\alpha^k = 1$.

Step 3 (Update of iteration point and Hessian approximations)
Set $x^{k+1} = x^k + d(x^k)$ and update the Hessian approximation matrices $B_j(x^k)$ by (3.8), (3.25), and (3.26).

Remark 3.5. In [39], one of the key points is that the authors have observed that when employing the Armijo-like line search rule in MOPs, more than one inequality should be satisfied simultaneously (see Line search steps in algorithms). Therefore, this rule imposes large computational efforts and yields a small step length. Subsequently, the use of the Armijo-like line search rule for finding a step length reduces the speed of the convergence. Hence, two different methods using Armijo line search and without any line search have been proposed in [39].

From here, a few questions may arise in readers' minds. Does the algorithm without line search work properly? Does it converge to a critical point? If yes, then what about the speed of the convergence or the number of iterations taken by the algorithm? To answer these kinds of questions, sufficient numerical data and theory are provided in [39]. First, we give the theory and then a brief description of the numerical results later.

The convergence analysis for Algorithm 5 has been proved in [34]. However, the convergence of Algorithm 6 is given in [39] using Lemma 3.4 and Lemma 3.5. First, a theorem similar to Theorem 3.4 has been given in [39]. After that, using this theorem, the convergence of Algorithm 6 has been proved.

Theorem 3.8. ([39]). *Let $\{x^k\}$ be a sequence generated by Algorithm 6 and x^0 be an initial point of this sequence. Let $\mathbb{B}[x^0,r]$ be a closed ball with radius $r>0$ and centre x^0. If we have $a,\delta,\varepsilon>0$ with $\varepsilon<a$ such that*

(i) $aI_{id}\leq B_j(x)$ *and* $aI_{id}\leq\nabla^2 f_j(x)$ *for all* $j\in I$ *and* $x\in\mathbb{B}[x^0,r]$,
(ii) $\|\nabla^2 f_j(y)-\nabla^2 f_j(x)\|<\varepsilon$ *for all* $j\in I$ *and* $x,y\in\mathbb{B}[x^0,r]$ *with* $\|y-x\|<\delta$,
(iii) $\frac{\|\left(\nabla^2 f_j(x^k)-B_j(x^k)\right)(y-x^k)\|}{\|y-x\|}<\frac{\varepsilon}{2}$ *for all* $j\in I$, $y\in\mathbb{B}[x^0,r]$ *and* $k\geq k_0$,
(iv) $\|d(x^0)\|<\min\{\delta,r(1-\frac{\varepsilon}{a})\}$, *then for any* $k\in\mathbb{N}$,
  (a) $x^k\in\mathbb{B}[x^0,r]$,
  (b) $\|d(x^k)\|<(\frac{\varepsilon}{a})^k\|d(x^0)\|$.

Corollary 3.2. ([39]). *The sequence generated by Algorithm 6 converges to a critical point for F under the assumptions of Theorem 3.8.*

In [39], quasi-Newton methods (with BFGS, SS-BFGS and H-BFGS updates) have been compared as well. The performance of the algorithms has been compared using three performance assessment criteria: the purity metric, the spread metric and the epsilon indicator. Moreover, authors in [39] have depicted numerical results using performance profile to have a good visual comparison. We provide a brief description of these performance profiles and assessments below.

Performance profiles

The performance profiles are introduced by Dolan and Moré in [47] as a means to evaluate and compare the performance of a set of algorithms $\mathscr{A}$ on a set of test problems $\mathscr{P}$. In this procedure, first performance ratio r_{p,a_l} for the algorithm $a_l\in\mathscr{A}$ in solving the problem $p\in\mathscr{P}$ is calculated by the formula

$$r_{p,a_l}=\frac{t_{p,a_l}}{\min\{t_{p,a_l}:a_l\in\mathscr{A}\}},$$

where t_{p,a_l} denotes the performance of the algorithm $a_l \in \mathscr{A}$ in solving the problem $p \in \mathscr{P}$ such that better performances correspond to lower values of t_{p,a_l}. Using the performance ratio, the performance of the algorithm for a particular problem can be compared. However, to obtain an overall assessment of the performance of the algorithm, a formula is given

$$\rho_{a_l}(\omega) = \tfrac{1}{|\mathscr{P}|}|p \in \mathscr{P} : r_{p,a_l} \leq \omega|.$$

$\rho_{a_l}(\omega)$ is the probability for the algorithm $a_l \in \mathscr{A}$ whose performance ratio is within the interval $[1,\omega]$. The value of $\rho_{a_l}(1)$ is the probability that the algorithm a_l will win over the rest of the algorithms. Thus, if we are interested only in the number of wins, we need only to compare the values of $\rho_{a_l}(1)$ for all of the algorithms. For example, $\rho_{a_{l_o}}(1) = 0.6$ means that the algorithm a_{l_o} has the best performance in solving 60 percent of the test problems in contrast to the rest of the algorithms. Moreover, algorithms with the largest probabilities $\rho_{a_l}(\omega)$ for large values of ω have a better performance than the other algorithms. In [39], authors have compared performance profiles for many values of ω (see Fig.1-Fig.4 of [39]). Next, assessment criteria are given.

Purity metric

The purity metric [48] is a criterion that is used to compare the non-dominated frontiers obtained by the algorithms. Let F_{p,a_l} be the non-dominated frontier obtained by the algorithm $a_l \in \mathscr{A}$ for problem $p \in \mathscr{P}$. After removing the dominated elements from the set $\cup_{a_l \in \mathscr{A}} F_{p,a_l}$, a set, namely the non-dominated reference frontier, denoted by F_p, is obtained. The purity metric for algorithm $a_l \in \mathscr{A}$ and problem $p \in \mathscr{P}$ is defined as

$$\bar{t}_{p,a_l} = \frac{c'_{p,a_l}}{c_{p,a_l}},$$

where $c'_{p,a_l} = |F_{p,a_l} \cap F_p|$ and $c_{p,a_l} = |F_{p,a_l}|$. It is obvious that the greater value of $\bar{t}_{p,a_l}$ indicates the higher percentage of non-dominated points for problem $p \in \mathscr{P}$. Therefore, in order to use performance profile, $t_{p,a_l} = \frac{1}{\bar{t}_{p,a_l}}$ has been considered in [39] instead of $\bar{t}_{p,a_l}$.

Spread metric

This metric is used to measure the largest gap in the obtained non-dominated frontiers [48]. To find the metric, in [39], first F_p is calculated. Then, assumed that the algorithm $a_l \in \mathscr{A}$ has obtained a set N having points indexed by $1,2,\ldots,N$ as the approximated non-dominated set. Also, $N+1, N+2, \ldots, N+k$ extreme points are added to these points. Let $f^{p,a_l}_{i,j}$ denotes the jth component of the ith point of this collection. The spread metric of the algorithm $a_l \in \mathscr{A}$ in solving the problem p is denoted by Γ_{p,a_l} and is defined by

$$\Gamma_{p,a_l} = \max_{j \in I}\left(\max_{i \in \{1,2,\ldots,N+k\}}\{\delta^{p,a_l}_{i,j}\}\right),$$

where $\delta^{p,a_l}_{i,j} = (f^{p,a_l}_{i+1,j} - f^{p,a_l}_{i,j})$.

Epsilon indicator

The epsilon indicator is a criterion that represents the quality of the obtained non-dominated frontiers. The epsilon indicator given in [39], denoted by $I_{\varepsilon+}(\mathscr{N})$, is defined by with respect to set F_p

$$I_{\varepsilon+}(\mathscr{N}) = \inf\{\varepsilon : \text{ for all } y \in F_p \text{ there exists } z \in \mathscr{N} \text{ such that } z \preceq_{\varepsilon+} y\},$$

where $\mathscr{N}$ is the obtained non-dominated set for the specified algorithm and $\preceq_{\varepsilon+}$ is the additive ε-dominance inequality defined as

$$z^1 \preceq_{\varepsilon+} z^2 \iff \text{ for all } j \in I, z_j^1 \leq_{\varepsilon} \varepsilon z_j^2.$$

There are many figures given in [39] to check the performance profiles and assessment. Fig.6 in [39] compares the methods (BFGS, SS-BFGS, and H-BFGS) using the purity metric, spread metric, and epsilon indicator. Moreover, Fig.7-Fig.10 in [39] show the comparison of methods based upon the obtained non-dominated frontiers related to some of the test problems.

After so much work on step length and approximation of the Hessian matrix in the quasi-Newton methods, authors in [38] extended their thoughts and proposed a non-monotone quasi-Newton method for unconstrained strongly convex MOP. In this new method, a non-monotone line search has been used instead of the conventional monotone Armijo line search. This method is similar to the method given in [34] except for the line search procedure. The quasi-Newton direction and approximate matrix are calculated in the same way as in [34]. The algorithm for this quasi-Newton method, proposed in [38], is given below.

Algorithm 7 Quasi-Newton Algorithm for Multiobjective Optimization Proposed in [38]

Step 0 (Initial step)

Initially choose x^0, constants $c, \rho, \alpha_0 \in (0,1)$, $C_j^0 = f_j(x^0)$, $Q_0 = 1$ and a positive definite initial matrix $B_j(x^0)$ for all $j \in I$.

Set iteration counter $k = 0$.

Step 2 (Line search)

Compute the step length α^k satisfies the non-monotone Armijo condition: $\alpha^k = \rho^{h_k}$ and h_k is the least nonnegative integer such that the following condition holds

$$g_j(x^k + \alpha^k d(x^k)) \leq C_j^k + c\alpha^k \tau(x^k), \; j \in I. \tag{3.27}$$

Choose $\eta_k \in [0,1]$, and set

$$Q_{k+1} = \eta_k Q_k + 1, \; C_j^{k+1} = \frac{\eta_k Q_k C_j^k + g_j(x^{k+1})}{Q_{k+1}}.$$

Remark 3.6. It can be easily seen that if $\eta_k = 0$, then $C^k = G(x^k)$ for all k. In this case, the line search is a conventional monotone Armijo line search. Similarly, if $\eta_k = 1$, then $C^k = A^k$, where $A^k = \frac{G(x^0)+G(x^1)+\cdots+G(x^k)}{k+1}$.

To check the well-definedness of Algorithm 7, a lemma has been given in [38], which is as follows.

Lemma 3.7. ([38]). *Under the consideration of Algorithm 7, C^k lies between the $G(x^k)$ and A^k for each iteration k in Algorithm 7, where $A^k = \frac{G(x^0)+G(x^1)+\cdots+G(x^k)}{k+1}$. Moreover, if x^k is a noncritical point for G, then there exists α^k satisfying the nonmonotone Armijo condition* (3.27).

Like in [36], global and local convergences for the Algorithm 7 are given in [38] under the different assumptions. The global convergence has been proved with the help of the following lemma, which is as follows.

Lemma 3.8. ([38]). *Let $\{x^k\}$ be a sequence generated by Algorithm 7. Suppose ∇g_j are such that $\|\nabla g_j(x) - \nabla g_j(x^k)\| \leq K\|x - x^k\|$ for all x on the segment connecting x^k and $x^k + \frac{\alpha^k}{\rho} d^k$, where $\alpha^k \leq \sigma$. If the non-monotone Armijo conditions are satisfied, then*

$$\alpha^k \geq \min\left\{\rho, \frac{2\rho(1-\mu)}{K} \frac{|\tau(x^k)|}{\|d(x^k)\|^2}\right\},$$

where all the parameters are the same as in Algorithm 7.

Theorem 3.9. ([38]). *Suppose $g_j, j \in I$ are bounded from below, $\eta < 1$ and there exists $c_1 > 0$ such that $|\tau(x^k)| \geq c_1\|d(x^k)\|^2$ for all $k = 1, 2, \ldots$. Then, every limit point of the sequence $\{x^k\}$ generated by the Algorithm 7 is critical for G under the assumptions of Lemma 3.8.*

A proposition is also given in [38], which is an alternative to the Lemma 3.2.

Proposition 3.2. ([38]). *Let $\{x^k\}$ be a bounded sequence generated by Algorithm 7. Suppose $a > 0$ is a scalar such that for all k and $j \in I$, we have*

$$z^\top B_j(x^k) z \geq a\|z\|^2 \text{ for all } z \in \mathbb{R}^n.$$

Then, $\lim_{k\to\infty} \tau(x^k) = 0$ *and* $\lim_{k\to\infty} \|d(x^k)\| = 0$.

The local convergence in [38] has been proved under some assumptions. We have written these assumptions before. Therefore, we are just mentioning these assumptions below.

A2.1 Under Algorithm 7, (3.9) and (3.10) hold.

Theorem 3.10. (Superlinear convergence [38]). *Let the assumptions in Proposition 3.2 and assumption A2.1 hold. Then, $\alpha^k = 1$, for sufficiently large k, and the sequence $\{x^k\}$ converges to a critical point $\bar{x} \in \mathbb{R}^n$ superlinearly.*

The q-quasi-Newton method is also given for MOP (2.1) in [37] using q-gradient of the objective function of MOP. However, for the second-order approximation of Hessian, the BFGS update formula (3.8) is used in [37]. Therefore, most of the analysis of the q-quasi-Newton method is the same as in previously mentioned methods. We start with the definition of q-gradient.

Definition 3.1. (q-gradient [49]). The q-gradient for a scalar-valued function $f_j, j \in I$ at $\bar{x} \in \mathbb{R}^n$, denoted by $\nabla_q f_j(\bar{x})$, is defined by

$$\nabla_q f_j(\bar{x}) = (D_{q,x_1} f_j(\bar{x}), D_{q,x_2} f_j(\bar{x}), \ldots, D_{q,x_n} f_j(\bar{x}))^\top,$$

where $D_{q,x_i} f_j(\bar{x})$ is the q-partial derivative of f_j at $\bar{x}$ defined by

$$D_{q,x_i} = \begin{cases} \frac{f_x - (\varepsilon_{q,i} f_j)(x)}{(1-q)x_i}, & x_i \neq 0, q \neq 1 \\ \frac{\partial f_j}{\partial x_i}, & x_i = 0 \end{cases}$$

with $(\varepsilon_{q,i})f_j(\bar{x}) = f_j(\bar{x}_1, \bar{x}_2, \ldots, q\bar{x}_i, \bar{x}_{i+1}, \ldots, \bar{x}_n)$.

The procedure to find the q-quasi-Newton direction in [37] is the same as in [34]. Analogous to subproblem (3.1), a subproblem is also mentioned in [37] using the q-gradient instead of the conventional gradient. The rest of the analysis remains the same because there is no change in the approximation of the Hessian matrix and in the calculation of step length in [37]. Therefore, we present the algorithm proposed in [37] without extra explanation below.

Algorithm 8 Quasi-Newton Algorithm for Multiobjective Optimization Proposed in [37]

Step 1 (Generation of search direction $\tilde{d}(x^k)$ and $\tilde{\tau}(x^k)$)

$$\tilde{d}(x^k) = \arg\min_{d\in\mathbb{R}^n} \max_{j\in I} \nabla_q f_j(x^k)^\top d + \tfrac{1}{2} d^\top \tilde{B}_j(x^k) d,$$
$$\tilde{\tau}(x^k) = \min_{d\in\mathbb{R}^n} \max_{j\in I} \nabla_q f_j(x^k)^\top d + \tfrac{1}{2} d^\top \tilde{B}_j(x^k) d.$$

Step 3 (Update of iteration point and Hessian approximations)
Set $x^{k+1} = x^k + \alpha^k \tilde{d}(x^k)$ and update the Hessian approximation matrices $\tilde{B}_j(x^k)$ by

$$\tilde{B}_j(x^{k+1}) = \tilde{B}_j(x^k) - \frac{\tilde{B}_j(x^k) s_k s_k^\top \tilde{B}_j(x^k)}{s_k^\top \tilde{B}_j(x^k) s_k} + \frac{\tilde{y}_j^k \tilde{y}_j^{k\top}}{s_k^\top \tilde{y}_j^k},$$

where $s_k = x^{k+1} - x^k$ and $\tilde{y}_j^k = \nabla_q f_j(x^{k+1}) - \nabla_q f_j(x^k)$ for each $j \in I$.

The convergence of the Algorithm 8 has been shown in [37] by a similar result to Theorem 3.4 with the change that conventional gradient is replaced by q-gradient.

All the work in the previous methods has been done using Armijo line search or without using any line search. However, authors in [41] have proposed a quasi-Newton method using the Wolfe line search technique for MOP (2.1) recently. There is no change in the approximated matrix and descent direction for the method in [41]. Therefore, the basic analysis for this method is the same as in the previous for this method. We present the algorithm given in [41] in Algorithm 9.

Although there is not much change in the basic analysis for Algorithm 9, the convergence of Algorithm 9 is not straightforward. First, the well-definedness of the algorithm is given in [41] for nonconvex MOPs. After that, the global and R-linear convergence have been proved for strongly convex multiobjective functions.

Theorem 3.11. (Well-definedness [41]). *Algorithm 9 is well defined under the assumption that F is bounded below in S_o.*

Before starting the global convergence for strongly convex multiobjective function, the following assumption is taken in [41] as well.

A3.1 The level set is convex and there exist constants $a, b > 0$ such that

$$a\|z\|^2 \le z^\top \nabla^2 g_j(x) z \le b\|z\|^2, j \in I,$$

for all $z \in \mathbb{R}^n$ and $x \in S_o$.

Theorem 3.12. ([41]). *Let $\{x^k\}$ be a sequence generated by Algorithm 9. Then, sequence $\{x^k\}$ converges to a Pareto optimal point $\bar{x}$ of G under the assumption A3.1.*

Algorithm 9 Quasi-Newton Algorithm for Multiobjective Optimization Proposed in [41]

Step 0 (Initial step)

Initially choose x^0, constants $\alpha^0 \in (0,1)$, $c_1 \in (0,\frac{1}{2})$, $c_2 \in (c_1,1)$ and a positive definite initial matrix $B_j(x^0)$ for all $j \in I$.

Set iteration counter $k = 0$.

Step 2 (Line search)

Compute the step length α^k satisfies

$$\begin{aligned} f_j(x^k + \alpha^k d(x^k)) \leq & f_j(x^k) + c_1 \alpha^k \max_{j\in I} \nabla f_j(x^k)^\top d(x^k),\ j \in I, \\ \nabla f_j(x^k + \alpha^k d(x^k))^\top d(x^k) \geq & c_2 \nabla f_j(x^k)^\top d(x^k). \end{aligned}$$

Step 3 (Update of iteration point and Hessian approximations)

Set $x^{k+1} = x^k + \alpha^k d(x^k)$ and update the Hessian approximation matrices $B_j(x^k)$ by

$$\begin{aligned} B_j(x^{k+1}) = & B_j(x^k) - \frac{(\rho_j^k)^{-1} B_j(x^k) s_k s_k^\top B_j(x^k)}{\left((\rho_j^k)^{-1} - s_k^\top y_j^k\right)^2 + (\rho_j^k)^{-1} s_k^\top B_j^k s_k} \\ & + \frac{(s_k^\top B_j(x^k) s_k) y_j^k (y_j^k)^\top}{\left((\rho_j^k)^{-1} - s_k^\top y_j^k\right)^2 + (\rho_j^k)^{-1} s_k^\top B_j^k s_k} \\ & + \left((\rho_j^k)^{-1} - s_k^\top y_j^k\right) \frac{y_j^k s_k^\top B_j^k + B_j^k s_k (y_j^k)^\top}{\left((\rho_j^k)^{-1} - s_k^\top y_j^k\right)^2 + (\rho_j^k)^{-1} s_k^\top B_j^k s_k}, \end{aligned} \tag{3.28}$$

where $y_j^k = \nabla f_j(x^{k+1}) - \nabla f_j(x^k)$, $s_k = x^{k+1} - x^k$ and

$$\rho_j^k = \begin{cases} \frac{1}{s_k^\top y_j^k}, & \text{if } s_k^\top y_j^k > 0 \\ \frac{1}{\left(\max_{j\in I}(\nabla f_j(x^k+\alpha^k d(x^k))^\top s_k) - \nabla f_j(x^k)^\top s_k\right)}, & \text{otherwise.} \end{cases}$$

A different approach to convergence is also given in [41]. The authors in [41] have shown that sequence $\{x^k\}$ converges to $\bar{x}$ rapidly enough that

$$\sum_k \|x^k - \bar{x}\| < \infty, \tag{3.29}$$

which plays an important role in the superlinear convergence of Algorithm 9.

Theorem 3.13. (R-linear convergence [41]). *Let $\{x^k\}$ be a sequence generated by the Algorithm 9. Let $\bar{x}$ be as in Theorem 3.12. Then, $\{x^k\}$ converges R-linearly to $\bar{x}$ and (3.29) holds.*

The local convergence of Algorithm 9 has been established for nonconvex multiobjective functions under suitable assumptions in [41]. First, we present the assumption and then the local convergence theorem.

A4.1 The sequence $\{x^k\}$ generated by Algorithm 9 converges to a Pareto optimal point $\bar{x}$.

A4.2 For Hessian matrices $\nabla^2 f_j(\bar{x})$, for all $j \in I$, there exist a neighborhood U of $\bar{x}$ and positive constants c_1, c_2 and K such that

$$c_1\|z\| \leq z^\top \nabla^2 f_j(x) z \leq b\|z\|^2 \text{ for all } j \in I$$

and

$$\|\nabla^2 f_j(x) - \nabla^2 f_j(\bar{x})\| \leq K\|x - \bar{x}\| \text{ for all } j \in I,$$

for all $z \in \mathbb{R}^n$ and $x \in U$. Now we give the theorem regarding the Q-superlinear convergence of Algorithm 9.

Theorem 3.14. (Q-superlinear convergence [41]). *Suppose $\{x^k\}$ is the sequence generated by the Algorithm 9. Then, $\{x^k\}$ converges to $\bar{x}$, Q-superlinearly, under the assumptions A4.1 and A4.2.*

An obvious comparison between standard BFGS-Armijo and standard BFGS-Wolfe algorithms is given in [41]. This comparison is based on various factors such as the number of iterations, the number of derivative evaluations, CPU time, etc. The numerical performance of the algorithm has been assessed in [41] as well. In [41], the Purity metric and Spread metrics have been used to compare the numerical results. The performance profiles have been considered in [41] for each problem as the performance measurements using (i) number of iterations; (ii) CPU time; (iii) number of functions evaluations; and (iv) number of derivative evaluations.

The newest addition in the series of quasi-Newton methods is the proximal quasi-Newton method for MOPs [40]. In [40], authors have proposed proximal methods with Armijo and without any line search for a special type of MOP (3.30). Also, the conventional BFGS, SS-BFGS, and H-BFGS update formulae are used to approximate the Hessian matrix of component functions of the objective function of MOP (3.30). We present an MOP, which is considered in [40], below

$$\min_{x \in \mathbb{R}^n} \mathscr{F}(x), \tag{3.30}$$

where $\mathscr{F} = (\mathscr{F}_1, \mathscr{F}_2, \ldots, \mathscr{F}_m)^\top : \mathbb{R}^n \to (\mathbb{R} \cup \{\infty\})^m$ is a vector-valued function. The components $\mathscr{F}_j$ of objective function $\mathscr{F}$ for all $j \in I$, are taken in [40], as

$$\mathscr{F}_j(x) = \Phi_j(x) + \Psi_j(x) \text{ for all } j \in I,$$

where $\Phi_j : \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable strongly convex function and $\Psi_j : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is a proper convex and lower semicontinuous but not necessarily differentiable.

Remark 3.7. If we take Ψ_j a zero function for all $j \in I$, i.e., $\Psi_j(x) = 0$ for all $x \in \mathbb{R}^n$ and $j \in I$, then MOP (3.30) reduces to the convex MOP evaluated in [34], i.e., MOP (3.2).

Since the objective function taken in [40] is different, therefore the previous theories and results cannot be applied directly to find the descent direction for this proximal quasi-Newton method. However, similar to the previous ones in [40], a subproblem is defined using the first and second-order information of objective function, and the solution of this subproblem is called the descent direction. That subproblem is

$$\min_{d \in \mathbb{R}^n} \varphi_{w,x}(d) = \theta_x(d) + \tfrac{w}{2}\|d\|^2, \tag{3.31}$$

where $\theta_x(d) = \max_{j \in I}\{\nabla \Phi_j(x)^\top d + \frac{1}{2} d^\top B_j(x) d + \Psi_j(x + d) - \Psi_j(x)\}$. The proximal quasi-Newton direction (basically descent direction for proximal quasi-Newton method) is defined

[40] by

$$d_w(x) = \operatorname*{argmin}_{d \in \mathbb{R}^n} \varphi_{w,x}(d). \tag{3.32}$$

After giving the idea of proximal quasi-Newton direction, an important lemma analogous to Lemma 3.2 is given in [40] replacing $\tau(x)$ and $d(x)$ by $\varphi_{w,x}(d)$ and $d_w(x)$, respectively. This lemma establishes a relation between stationarity of a point x, $\varphi_{w,x}(d)$ and $d_w(x)$. There is no change in BFGS update in [40]. In fact, authors in [40] have used not only conventional BFGS but SS-BFGS and H-BFGS also as in [39]. Similar to [39], two proximal quasi-Newton methods (with or without line searches) have been given in [40]. Now we are able to study the algorithms proposed in [40]. Below we are presenting the proximal quasi-Newton method with line searches first followed by the proximal quasi-Newton method in which any line search is not used.

Algorithm 10 Quasi-Newton Algorithm with Armijo line search for Multiobjective Optimization Proposed in [40]

Step 1 (Generation of search direction $d_w(x^k)$ and $\tau_w(x^k)$)
Solve subproblem (3.31) at x^k. The solution $d_w(x^k)$ of subproblem (3.31) at x^k is a search direction. The optimal objective function value of subproblem (3.31) at x^k is $\tau_w(x^k)$. If $d_w(x^k) = 0$, then stop.

Step 3 (Update iteration point and Hessian approximation)
Set $x^{k+1} = x^k + \alpha^k d_w(x^k)$ and update Hessian approximation matrix $B_j(x^k)$ by (3.22), (3.25), and (3.26).

Now we describe the algorithm without any line search proposed in [40]. In this algorithm, there is no update in step length in the line search step. The authors have assumed that $\alpha^k = 1$ at every iteration. So, the updated point at kth iteration is $x^{k+1} = x^k + d_w(x^k)$.

Algorithm 11 Quasi-Newton Algorithm without any Line Search for Multiobjective Optimization Proposed in [40]

Step 1 (Generation of search direction $d_w(x^k)$ and $\tau_w(x^k)$)
Solve subproblem (3.31) at x^k. The solution $d_w(x^k)$ of subproblem (3.31) at x^k is a search direction. The optimal objective function value of subproblem (3.31) at x^k is $\tau_w(x^k)$. If $d_w(x^k) = 0$, then stop.

Step 2 (Line search)
Take $\alpha^k = 1$.

Step 3 (Update of iteration point and Hessian approximations)
Set $x^{k+1} = x^k + d_w(x^k)$ and update the Hessian approximation matrices $B_j(x^k)$ by (3.8), (3.25), and (3.26).

Remark 3.8. One thing is to note that the stopping criteria given in Algorithms 10 and 11 is $d_w(x^k) = 0$, which the solution of the subproblems considered in [40]. Instead of taking $d_w(x^k) = 0$, it should be $\tau_w(x^k) = 0$ as taken in previous algorithms, which is more suitable for the algorithm.

The authors in [40] have provided the convergence analysis for both algorithms under the assumptions. The convergence of Algorithm 10 is presented below.

Theorem 3.15. ([40]). *Let $\mathscr{F}_j$ be bounded from below for all $j \in I$. Then, every accumulating point of the sequence $\{x^k\}$ generated by Algorithms 10 and 11, if it exists, is a Pareto stationary point. Moreover, if level set $\{x \in \mathbb{R}^n : \mathscr{F}(x) \leq \mathscr{F}(x^0)\}$ is bounded, then $\{x^k\}$ has accumulation points and they are all Pareto stationary.*

To our best knowledge, the paper [40] is the latest one to have a quasi-Newton algorithm for MOPs. This paper may be last till now but not least. This paper has two applications, which is a new things and totally different from the others. The first application is in constrained multiobjective optimization, and the second is in robust multiobjective optimization.

In [40], many numerical experiments are performed to verify the effectiveness of the proposed algorithms. The authors in [40] have compared their methods with proximal gradient method given in [50]. Based upon the numerical data, it is also mentioned in [40] that the introduced method is more effective than the proximal gradient method given in [50]. A handy comparison between the algorithms using (3.22), (3.25), and (3.26) formulae with line search is also given in [40]. The algorithm with the H-BFGS update formula is more effective than the others. A similar comparison between the algorithms without line searches is also given in [40]. In this comparison also, the introduced quasi-Newton method in [40] without line search is more effective than the proximal gradient method without line search given in [50]. And the algorithm without line search with the H-BFGS update formula is more effective than the algorithms without line search with BFGS and SS-BFGS updates formulae. However, the convergence rate of Algorithms 10 and 11 is not discussed in [40].

4. Quasi-Newton method for unconstrained nonsmooth MOPs

In Section 3, the unconstrained smooth MOP is considered. All the variants of quasi-Newton methods are to solve the smooth MOP. However, there may happen that the objective function in MOP is nondifferentiable, i.e., MOP may be nonsmooth. To solve nonsmooth MOP, a quasi-Newton method is also proposed by Qu et al. [42]. In particular, authors of [42] have extended the quasi-Newton methods to MOP, whose objective function is nondifferentiable, but a directional derivative of each component of the objective function exists.

In [42], the following unconstrained nonsmooth MOP has been considered

$$\min_{x \in \mathbb{R}^n} L(x), \tag{4.1}$$

where $L = (l_1, l_2, \ldots, l_m)^\top : \mathbb{R}^n \to \mathbb{R}^m$ is a locally Lipschitz continuous nondifferentiable nonconvex multiobjective function.

From now onwards, we consider L and l_j for all $j \in I$ as defined in (4.1) throughout the paper.

Before proposing any method to solve MOPs, it is necessary to define the critical point for the objective function of MOP (see Remark 2.1). Since Definition 2.3 of critical point demands the Jacobian of multiobjective function, therefore a similar definition of critical point for a nondifferentiable multiobjective function is introduced in terms of Clarke derivative or generalized Jacobian.

Definition 4.1. (Clarke derivative [51]). The Clarke derivative of L at $\bar{x}$ is defined by

$$\partial L(\bar{x}) = \text{conv}\{A : A = \lim_{k\to\infty} \text{J}(L(x_k)) \text{ for some } \{x_k\} \subset \{\theta_L : x_k \to \bar{x}\}\},$$

where θ_L is the set of all points for which L is Fréchet differentiable.

Definition 4.2. (Critical point for nondifferentiable multiobjective functions [42]). A point $\bar{x} \in \mathbb{R}^n$ is critical for L if

$$\mathscr{R}(\partial L(\bar{x})) \cap (-\mathbb{R}^m_{++}) = \emptyset.$$

Definition 4.2 explains that if $\bar{x} \in \mathbb{R}^n$ is a critical point for L, then there does not exist a descent direction at $\bar{x}$. To give the algorithm for MOP (4.1), an assumption is taken in [42] to guarantee the existence of the standard directional derivative.

A5.1 Assume $\displaystyle\lim_{\substack{U_j \in \partial L_j(x+\alpha d)\\ \alpha\to 0}} (U_j^\top d)$ exists for all $j \in I$.

Under this assumption, an analogous result to Proposition 3.1 has been given below.

Lemma 4.1. ([42]). *Under the assumption A5.1, a point $\bar{x} \in \mathbb{R}^n$ is critical for L if and only if one of the following two conditions is satisfied:*

(i) *there does not exist a descent direction at $\bar{x}$, i.e., $l'_{j_o}(\bar{x}; d) \geq 0$ for at least one $j_o \in I$ and for all $d \in \mathbb{R}^n$;*

(ii) *in a special case, there also exists at least one $j_o \in I$ such that $0 \in \partial l_{j_o}(\bar{x})$.*

The objective function in MOP (4.1) is locally Lipschitz continuous, not necessarily convex. A theorem is given below that connects the efficiency and criticality of the Lipschitz multiobjective function.

Theorem 4.1. ([42]). *Let the assumption A5.1 holds.*

(i) *If $\bar{x} \in \mathbb{R}^n$ is a locally weak Pareto optimum, then $\bar{x}$ is a critical point for L.*

(ii) *If L is $\mathbb{R}^m$-convex and $\bar{x} \in \mathbb{R}^n$ is critical for L, then $\bar{x}$ is a weak Pareto optimum.*

(iii) *If L is strictly $\mathbb{R}^m$-convex and $\bar{x} \in \mathbb{R}^n$ is critical for L, then $\bar{x}$ is Pareto optimal.*

Next, a nonsmooth subproblem and related results are provided to find the descent direction for Algorithm 12. This nonsmooth subproblem is analogous to subproblem (3.1), which is expressed as follows (see [42]):

$$\begin{aligned} &\text{minimize } \ t \\ &\text{subject to } \phi_j(x,d) + \tfrac{1}{2} d^\top B_j(x) d - t \leq 0, \ j \in I \\ &\qquad\qquad \|d\| \leq 1, t \leq -\varepsilon, d \in \mathbb{R}^n, \end{aligned} \tag{4.2}$$

where $\phi_j(x,d)$ and $\frac{1}{2} d^\top B_j(x) d$ carry certain first-order and second-order informations of $l_j(x)$, respectively. In [42], $-\phi(\cdot,d)$ is lower semicontinuous for any $d \in \mathbb{R}^n$ and $\phi(x,\alpha d) \leq \alpha\phi(x,d)$ for all $x \in L_o, 0 \leq \alpha \leq 1$, where $S'_o = \{x \in \mathbb{R}^n : L(x) \leq L(x_o)\}$ is the level set.

Remark 4.1. Note that if L is twice continuously differentiable, then $\phi_j(x,d) = \nabla l_j(x)^\top d$. Thus, subproblem (4.2) reduces to subproblem (3.1) for twice continuously differentiable multiobjective function.

Under suitable assumptions, if the feasible set of subproblem (4.2) is empty for some sufficiently small $\varepsilon > 0$, then there does not exist a descent direction at x, and a good estimate to a critical point is obtained. The reason for choosing some small enough positive ε in subproblem (4.2) is that with $t \le 0$ replaced by $t \leqslant -\varepsilon$, the trivial solution $\left(\bar{d},\bar{t}\right) = (0,0)$ can be excluded. If ε is nonzero, subproblem (4.2) may be infeasible. In this case, the current point x is a good estimate of a critical point for L. The following theorem explains this under the following assumptions in [42].

A6.1 For all $x \in \mathbb{R}^n, d \in \mathbb{R}^n$, $\liminf\limits_{t\to 0} \frac{\phi(x,td)}{t} \le l'(x;d)$, where $\phi = (\phi_1,\ldots,\phi_m)^\top$.

A6.2 For all $x \in \mathbb{R}^n, \phi(x,0) = 0$, and $\phi(x,\cdot)$ is lower semicontinuous.

Theorem 4.2. ([42]). *Suppose Assumptions A5.1, A6.1, and A6.2 hold. For a sufficiently small positive scalar ε, if the feasible set of subproblem (4.2) is nonempty, then x is noncritical and any feasible solution $d_\varepsilon(x)$ is a descent direction for L; otherwise, x is a good estimate of the critical point for L.*

Take $S(x,\varepsilon)$

$$= \left\{(d,t) \in R^{n+1} : \phi_j(x,d) + \tfrac{1}{2} d^\top B_j(x) d \leqslant t, \text{ for all } j \in I, \|d\| \le 1, t \leqslant -\varepsilon\right\}.$$

Then, the optimal solution $\bar{d}_\varepsilon(x) = (d_\varepsilon(x), t_\varepsilon(x))$ of subproblem (4.2) and its corresponding optimal value $v_\varepsilon(x)$ can be expressed as $v_\varepsilon(x) = \min\limits_{(d,t)\in S(x,\varepsilon)} t$ and $\bar{d}_\varepsilon(x) = (d_\varepsilon(x), t_\varepsilon(x)) = \operatorname*{argmin}\limits_{(d,t)\in S(x,\varepsilon)} t$, where $d_\varepsilon(x)$ is the search direction. Note that $v_\varepsilon(x) = +\infty$ when the feasible set $S(x,\varepsilon)$ is empty. If $\varepsilon = 0$, then define

$$v(x) = v_0(x) = \inf_{\|d\|\le 1} \max_{j\in I} \phi_j(x,d) + \tfrac{1}{2} d^\top B_j(x) d,$$

$$d(x) = d_0(x) = \operatorname*{argmin}_{\|d\|\le 1} \max_{j\in I} \phi_j(x,d) + \tfrac{1}{2} d^\top B_j(x) d.$$

Using this observation, a lemma has been given below. This lemma is analogous to the very important result Lemma 3.2 to discuss the stability analysis of function $v_\varepsilon(x)$ and a connection with $\bar{d}_\varepsilon(x)$ and the stationarity of x. After that, we give the algorithm proposed in [42].

Lemma 4.2. ([42]). *Let Assumptions A5.1, A6.1 and A6.2 hold. Then,*

(i) *the following conditions are equivalent:*
   (a) *$x \in \mathbb{R}^n$ is noncritical for L;*
   (b) *for sufficiently small $\varepsilon > 0$, the solution set $S(x,\varepsilon)$ is nonempty and $(0,0) \notin S(x,\varepsilon)$;*
   (c) $v_\varepsilon(x) < 0$.

(ii) *the feasible set $S(x,\varepsilon)$ is nonempty for $x \in \mathbb{R}^n$ if and only if $v(x) = v_\varepsilon(x)$.*

(iii) *If $\bar{x}$ is critical, then the value function $v_\varepsilon(\cdot)$ defined by $v_\varepsilon(x) = \min\limits_{(d,t)\in S(x,\varepsilon)} t$ is noncontinuous at $\bar{x}$.*

One of the key points of the work in [42] is that the authors have proved global and local convergence separately under separate assumptions.

To prove global convergence, the following basic assumptions have been taken.

A7.1 The level set S_o is bounded.

A7.2 For any convergent subsequence $\left\{x^k\right\}_{k\in K}$, if $d(x^k) \to 0$, then $\max\limits_{j\in I}(l_j(x^k + d(x^k)) - l_j(x^k)) \le \max\limits_{j\in I} \phi_j(x^k, d(x^k)) + o(\|d(x^k)\|)$.

Algorithm 12 Quasi-Newton Algorithm for Nonsmooth Multiobjective Optimization Proposed in [42]

Step 1 (Generation of search direction $d_\varepsilon(x^k)$ and $v_\varepsilon(x^k)$)
Solve subproblem (4.2) at x^k. The solution $d_\varepsilon(x^k)$ of subproblem (4.2) at x^k is a search direction. The optimal objective function value of subproblem (4.2) at x^k is $v_\varepsilon(x^k)$. If subproblem (4.2) is infeasible, then terminate.

Step 3 (Update iteration point and Hessian approximation)
Set $x^{k+1} = x^k + \alpha^k d_\varepsilon(x^k)$ and update Hessian approximation matrix $B(x^k)$ by

$$B_j(x^{k+1}) = \begin{cases} B_j(x^k) - \frac{B_j(x^k)s^k {s^k}^\top B_j(x^k)}{{s^k}^\top B_j(x^k)s^k} + \frac{{y'}_j^k {{y'}_j^k}^\top}{{s^k}^\top {y'}_j^k}, & \text{if } k \in {K'}_j \\ B_j(x^k), & \text{otherwise,} \end{cases}$$

where ${K'}_j = \{k : {{y'}_j^k}^\top s^k \geq \varepsilon \min\{-v_\varepsilon(x^k), 1\}\}$, $s^k = x^{k+1} - x^k$ and ${y'}_j^k = \phi_j(x^{k+1}, d(x^{k+1})) - \phi_j(x^k, d(x^k))$, for each $j \in I$.

A7.3 Assume that for sufficiently large k, the step length $\alpha^k = 1$ is accepted.

Theorem 4.3. ([42]). *Let there exists a constant c such that $\|B_j(x)\| \leq c$ for any $x \in S_o$ and $j \in I$. Then, under assumptions A5.1, A6.1, A6.2, and A7.1-A7.3, every accumulation point of the sequence x^k generated by Algorithm (4), is critical for L.*

To prove local convergence, the following basic assumptions have been taken in [42].

A8.1 Given $0 < c_0 \leq c_1 < \infty$ and $0 < b_0 \leq b_1 < \infty$ for all $x \in S_o$, assume that $c_0 \leq \|B_j(x)\| \leq c_1$ for all $j \in I; b_0 \leq \|U_j\| \leq b_1$ for all $U_j \in \partial l_j(x)$ for all $j \in I$;

A8.2 Let $\bar{x}$ be an accumulation point of the sequence $\{x_k\}$ generated from Algorithm (4);

A8.3 Suppose the following conditions hold:
$\bar{U}_j - U_j^k - B_j(x^k)(\bar{x} - x^k) = o(\|\bar{x} - x^k\|)$ for all $j \in I$;
$\lambda_j^k - \bar{\lambda}_j = o(\|\bar{x} - x^k\|)$ for all $j \in I$;
$$\left(\sum_{j\in I} \lambda_j^k B_j\left(x^k\right)\right)^{-1} - \left(\sum_{j\in I} \bar{\lambda}_j B_j(\bar{x})\right)^{-1} = o\left(\left\|\bar{x} - x^k\right\|\right), \text{ for all } j \in I.$$

Remark 4.2. There are some reasons and explanations for taking assumptions A8.1-A8.3. Assumption A8.1 implies that $\left\|\left(\sum_{j\in I} \lambda_j^k B_j\left(x^k\right)\right)^{-1}\right\|$ and $\left\|\sum_{j\in I} \bar{\lambda}_j \bar{U}_j\right\|$ are bounded from above. Assumption A8.3 implies that the matrix $B_j\left(x^k\right)$ and parameter λ^k generated in every step converge to $B_j(\bar{x})$ and $\bar{\lambda}$, respectively when $\{x^k\}$ converges to $\bar{x}$.

Theorem 4.4. (Superlinear convergence [42]). *Suppose that the assumptions* A5.1, A7.3, and A8.1-A8.3 *hold. If for any $j \in I$, l_j is locally Lipschitzian at $\bar{x}$, then $\{x^k\}$ is superlinearly convergent to $\bar{x}$.*

In [42], authors have considered an example and have performed the proposed method. Also, they have compared the method with the subgradient method given in [52].

5. DISCUSSION

After a significant review of quasi-Newton methods, one question arises. Which method is best? Unfortunately, there is no specific answer. However, the strengths and weaknesses of the methods can be discussed so that readers can prefer the method based on the problem. In this section, we discuss the pros and cons of the methods. The first method proposed in [35] is for both nonconvex and convex MOPs. In [35], there are no numerical results and numerical comparisons given. Therefore, the accuracy of this method can not be discussed. The method proposed in [34] is compared with Newton's method for MOPs given in [32]. In this comparison, it has also been shown that the approximation of Hessian is faster than its evaluation (which is a big advantage of this method). This method is for strongly convex MOPs. This method does not apply to nonconvex MOPs in general. However, the method given in [34] can solve some nonconvex MOPs (PNR [45] is solved in [34]). Some nonconvex problems cannot be solved by this method; for instance, consider JOS4 and ZDT1-ZDT4 from [53] and [54], respectively. The first component function of these problems is $g_1(x) = x_1$. Since $y_1^k = \nabla g_1(x^{k+1}) - \nabla g_1(x^k) = 0$, BFGS update formula (3.8) cannot be applied. There are several advantages of the method given in [36]. This version of quasi-Newton methods has one common Hessian approximation matrix instead of having one for each of the component functions of the objective functions. The algorithm in [36] does not depend upon the choice of the initial point. It is also mentioned in [36] that the method converges not only to Pareto critical but also to local weak efficient points, even for nonconvex MOPs with detailed analysis. However, the authors did not prove the rate of convergence in [36] but mentioned that the method has at least superlinear convergence under several reasonable assumptions. Sometimes the condition $\cos^2(\theta_j^k) > \delta$ mentioned in Theorem 3.7 is difficult to handle. A comprehensive and comparative study of quasi-Newton methods is given in [39]. The plus point of [39] is the detailed numerical analysis of the methods proposed in [39]. Not only comparison but performance assessment (performance profiles; purity metric, spread metric, and epsilon indicator) of the methods is also given in [39]. By considering the numerical data provided in [39], the quasi-Newton method (with and without line search) with the H-BFGS update formula is better than the other variants. However, the authors in [39] have also mentioned that the H-BFGS method without line search has a faster speed of convergence. The reason is that when we employ the Armijo line search rule in MOPs, more than one inequality (equal to the number of component functions of the objective function) should be satisfied simultaneously. Thus, this rule imposes large computational efforts and reduces the speed of convergence. For numerical purposes, this paper is far ahead of the other one. The non-monotone quasi-Newton method presented in [38] is for strongly convex functions (although there are some nonconvex problems also solved by this method, for instance, H1 [55], KW2 [56], PNR [45], DTLZ2 [45], etc.). This method is better than the monotone quasi-Newton method. However, the comparison with H-BFGS update quasi-Newton is not given in [38]. Therefore, which one is better between them is still a question. One of the salient features of the method given in [38] is that this method is superlinear convergent, which is proved with detailed theoretical analysis. Overall this method is good, but the choice of η_k is a matter of concern sometimes. In [38], the value of η_k is taken 0.85, which works pretty well. However, who knows what will be the suitable value of η_k for other test problems (which are not considered in [38]). The q-quasi-Newton method introduced in [37] using q-derivative is also superlinear. This method is parameters-free. This method is compared with the method given in [34]. Two points can be highlighted. First, the performance profile of the method is not given in [37]. Second, the value of q may concern with finding the q-derivative. The authors in [37] did not mention the value of q while solving the test problems neither. The Wolfe quasi-Newton [41] converges to a Pareto optimal point globally and R-linearly for strongly convex problems. In [41], the given method has been compared with BFGS Armijo and standard BFGS Wolfe methods. The numerical experiments in [41] show that the formula (3.28) provides a nonnegligible acceleration of the BFGS method which is a big plus point of this method. The method given in [40] has three variants (BFGS, SS-BFGS, H-BFGS) as in [39] with and without line search. In [40] also, H-BFGS performs better than the others. The proximal quasi-Newton method is compared with the proximal gradient method [50]. However, the authors in [40] did not compare the BFGS, SS-BFGS, and H-BFGS proximal quasi-Newton methods with the BFGS, SS-BFGS and H-BFGS methods proposed in [39]. Therefore, one can not choose one of them. This can be a future direction for the researchers.

To our best knowledge, we have tried to present key points of the variants of the quasi-Newton method briefly. Based on the discussion, the reader can choose the suitable version of the quasi-Newton method.

Acknowledgments

We put our sincere thanks to Dr. Balendu Bhooshan Upadhyay, Department of Mathematics, Indian Institute of Technology Patna, India, Prof. Shashi Kant Mishra, Department of Mathematics, Banaras Hindu University, India, and Prof. Pierre Marechal, Université Paul Sabatier, France, who requested D. Ghosh to write a review article on the topic of the paper. D. Ghosh acknowledges the financial support of the research grants MATRICS (MTR/2021/000696) and Core Research Grant (CRG/2022/001347) by the Science and Engineering Research Board, India. The research of X. Zhao was supported in part by the National Natural Science Foundation of China (Grant Number 11801411).

REFERENCES

[1] V. Bhaskar, S.K. Gupta, A.K. Ray, Applications of multiobjective optimization in chemical engineering, Rev. Chem. Eng. 16 (2000), 1–54.

[2] H. Eschenauer, J. Koski, A. Osyczka, Multicriteria Design Optimization: Procedures and Applications, Springer, Berlin, 1990.

[3] R.T. Marler, J.S. Arora, Survey of multi-objective optimization methods for engineering, Struct. Multidisc. Optim. 26 (2004), 369–395.

[4] M. Gravel, J.M. Martel, R. Nadeau, W. Price, R. Tremblay, A multicriterion view of optimal resource allocation in job-shop production, Eur. J. Oper. Res. 61 (1992), 230–244.

[5] D. White, Epsilon-dominating solutions in mean-variance portfolio analysis, Eur. J. Oper. Res. 105 (1998), 457–466.

[6] J. Fliege, Olaf–a general modeling system to evaluate and optimize the location of an air polluting facility, OR Spektrum 23 (2001), 117–136.

[7] Jauny, D. Ghosh, A. Upadhayay, Q.H. Ansari, A trust-region interior-point technique to solve multi- objective optimization problems and its application to a tuberculosis optimal control problem, J. Nonlinear Var. Anal. 6 (2022), 675–691.

[8] S. Peitz, M. Dellnitz, A survey of recent trends in multiobjective optimal control—surrogate models, feedback control and objective reduction, Math. Comput. Appl. 23 (2018) 30.

[9] K.K. Annamdas, S.S. Rao, Multi-objective optimization of engineering systems using game theory and particle swarm optimization, Eng. Optim. 41 (2009), 737–752.

[10] W.-Y. Liu, K. Yue, T.-Y. Wu, M.-J. Wei, An approach for multi-objective categorization based on the game theory and Markov process, Appl. Soft Comput. 11 (2011), 4087–4096.
[11] H. Yu, H. Liu, Robust multiple objective game theory, J. Optim. Theory Appl. 159 (2013), 272–280.
[12] M. Ehrgott, Multicriteria Optimization, Springer, Berlin, 2005.
[13] T. Erfani, S.V. Utyuzhnikov, Directed search domain: a method for even generation of the Pareto frontier in multiobjective optimization, Eng. Optim. 43 (2011), 467–484.
[14] I. Giagkiozis, P.J. Fleming, Methods for multi-objective optimization: An analysis, Inf. Sci. 293 (2015), 338–350.
[15] N. Gunantara, A review of multi-objective optimization: methods and its applications, Cogent Eng. 5 (2018), 1502242.
[16] R.T. Eckenrode, Weighting multiple criteria, Manag. Sci. 12 (1965), 180–192.
[17] J. Koski, Defectiveness of weighting method in multicriterion optimization of structures, Commun. Appl. Numer. Methods 1 (1985), 333–337.
[18] Y. Haimes, On a bicriterion formulation of the problems of integrated system identification and system optimization, IEEE Trans. Syst. Man Cybern. 1 (1971), 296–297.
[19] I. Das, J.E. Dennis, Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems, SIAM J. Optim. 8 (1998), 631–657.
[20] A. Messac, C.A. Mattson, Normal constraint method with guarantee of even representation of complete Pareto frontier, AIAA J. 42 (2004), 2101–2111.
[21] D. Ghosh, D. Chakraborty, A new Pareto set generating method for multi-criteria optimization problems, Oper. Res. Lett. 42 (2014), 514–521.
[22] A. Messac, Physical programming-effective optimization for computational design, AIAA J. 34 (1996), 149–158.
[23] D. Mueller-Gritschneder, H. Graeb, U. Schlichtmann, A successive approach to compute the bounded Pareto front of practical multiobjective optimization problems, SIAM J. Optim. 20 (2009), 915–934.
[24] A. Pascoletti, P. Serafini, Scalarizing vector optimization problems, J. Optim. Theory Appl. 42 (1984), 499–524.
[25] A. Upadhayay, D. Ghosh, Q.H. Ansari, Jauny, Augmented Lagrangian cone method for multiobjective optimization problems with an application to an optimal control problem, Optim. Eng. (2022), https://doi.org/10.1007/s11081-022-09747-y.
[26] J. Fliege, B.F. Svaiter, Steepest descent methods for multicriteria optimization, Math. Methods Oper. Res. 51 (2000), 479–494.
[27] L.M.G. Drummond, B.F. Svaiter, A steepest descent method for vector optimization, J. Comput. Appl. Math. 175 (2005), 395–414.
[28] L.M.G. Drummond, A.N. Iusem, A projected gradient method for vector optimization problems, Comput. Optim. Appl. 28 (2004), 5–29.
[29] H. Bonnel, A.N. Iusem, B.F. Svaiter, Proximal methods in vector optimization, SIAM J. Optim. 15 (2005), 953–970.
[30] E.H. Fukuda, L.M.G. Drummond, A survey on multiobjective descent methods, Pesqui. Oper. 34 (2014), 585–620.
[31] L.R.L. Pérez, L.F. Prudente, Nonlinear conjugate gradient methods for vector optimization, SIAM J. Optim. 28 (2018), 2690–2720.
[32] J. Fliege, L.M.G. Drummond, B.F. Svaiter, Newton's method for multiobjective optimization, SIAM J. Optim. 20 (2009), 602–626.
[33] J. Wang, Y. Hu, C.K.W. Yu, C. Li, X. Yang, Extended Newton methods for multiobjective optimization: majorizing function technique and convergence analysis, SIAM J. Optim. 29 (2019), 2388–2421.
[34] Ž. Povalej, Quasi-Newton's method for multiobjective optimization, J. Comput. Appl. Math. 255 (2014), 765–777.
[35] S. Qu, M. Goh, F.T.S. Chan, Quasi-Newton methods for solving multiobjective optimization, Oper. Res. Lett. 39 (2011), 397–399.
[36] M.A.T. Ansary, G. Panda, A modified quasi-Newton method for vector optimization problem, Optimization 64 (2015), 2289–2306.

[37] K.K. Lai, S.K. Mishra, B. Ram, On q-quasi-Newton's method for unconstrained multiobjective optimization problems, Mathematics 8 (2020), 616.

[38] N. Mahdavi-Amiri, F.S. Sadaghiani, A superlinearly convergent nonmonotone quasi-Newton method for unconstrained multiobjective optimization, Optim. Meth. Softw. 35 (2020), 1223–1247.

[39] V. Morovati, H. Basirzadeh, L. Pourkarimi, Quasi-Newton methods for multiobjective optimization problems, 4OR-Q. J. Oper. Res. 16 (2018), 261–294.

[40] J.-W. Peng, J. Ren, Proximal quasi-Newton methods for multiobjective optimization problems, arXiv preprint, (2022), arXiv:2108.00125v2.

[41] L.F. Prudente and D. Souza, A quasi-Newton method with Wolfe line searches for multiobjective optimization, J. Optim. Theory Appl. 194 (2022), 1107–1140.

[42] S. Qu, C. Liu, M. Goh, Y. Li, Y. Ji, Nonsmooth multiobjective programming with quasi-Newton methods, Eur. J. Oper. Res. 235 (2014), 503–510.

[43] W. Zou, Y. Zhu, H. Chen, B. Zhang, Solving multiobjective optimization problems using artificial bee colony algorithm, Discrete Dyn. Nat. Soc. 2011 (2011), 569784.

[44] D.-H. Li, M. Fukushima, On the global convergence of the BFGS method for nonconvex unconstrained optimization problems, SIAM J. Optim. 11 (2001), 1054–1064.

[45] M. Preuss, B. Naujoks, G. Rudolph, Pareto set and EMOA behavior for simple multimodal multiobjective functions, In: T.P. Runarsson, H.-G. Beyer, E. Burke, J.J. Merelo-Guervós, L.D. Whitley, X. Yao, (eds.) Parallel Problem Solving from Nature-PPSN IX, Lecture Notes in Computer Science, vol 4193, pp. 513–522, Springer, Berlin, Heidelberg, 2006.

[46] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Trans. Evol. Comput. 10 (2006), 477–506.

[47] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, Math. Program. 91 (2002), 201–213.

[48] A.L. Custódio, J.F.A. Madeira, A.I.F. Vaz, L.N. Vicente, Direct multisearch for multiobjective optimization, SIAM J. Optim. 21 (2011), 1109–1140.

[49] P.M. Rajković, S.D. Marinković, M.S. Stanković, Fractional integrals and derivatives in q-calculus, Appl. Anal. Discret. Math. 1 (2007), 311–323.

[50] H. Tanabe, E.H. Fukuda, N. Yamashita, Proximal gradient methods for multiobjective optimization and their applications, Comput. Optim. Appl. 72 (2019), 339–361.

[51] R.T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, New Jersey, 1970.

[52] J.X. da Cruz Neto, G. Da Silva, O.P. Ferreira, J.O. Lopes, A subgradient method for multiobjective optimization, Comput. Optim. Appl. 54 (2013), 461–472.

[53] Y. Jin, M. Olhofer, B. Sendhoff, Dynamic weighted aggregation for evolutionary multi-objective optimization: why does it work and how? In: L. Spector, E.D. Goodman, A. Wu, W.B. Langdon, H.-M. Voigt, (eds.) Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation (GECCO'01), pp. 1042–1049, Morgan Kaufmann Publishers Inc., 2001, https://dl.acm.org/doi/abs/10.5555/2955239.2955427.

[54] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2000), 173–195.

[55] C. Hillermeier, Nonlinear multiobjective optimization: a generalized homotopy approach, Springer Science & Business Media, 2001.

[56] I.Y. Kim, O.L. De Weck, Adaptive weighted-sum method for bi-objective optimization: Pareto front generation, Struct. Multidisc. Optim. 29 (2005), 149–158.