Communications in Combinatorics and Optimization
Vol. xx, No. x (xxxx), pp. 1-34
https://doi.org/10.22049/cco.2025.30421.2459

CCO
Commun. Comb. Optim.

Research Article

A new quasi-Newton algorithm for constructing the Pareto front of multiobjective optimization problems by implementing warm-start strategies

Fereshteh Akbari[1,†], Esmaile Khorram[1,*], Mehrdad Ghaznavi[2]

[1]Faculty of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran
[†]f.akbari94@aut.ac.ir
[*]eskhor@aut.ac.ir

[2]Faculty of Mathematical Sciences, Shahrood University of Technology, Shahrood, Iran
ghaznavi@shahroodut.ac.ir

Received: 14 March 2025; Accepted: 17 may 2025
Published Online: 10 September 2025

Abstract: Many numerical procedures for finding efficient solutions of multiobjective optimization problems are variants of Newton method, that utilize the Hessian matrix of second derivatives. Quasi-Newton methods are used for situations in which the calculation of the Hessian matrix or its inverse is difficult or expensive. In the quasi-Newton methods, only first derivatives are utilized to build an approximation of the actual Hessian matrix over a number of iterations. One of the weaknesses of Newton and quasi-Newton methods is choosing the proper starting points. In fact, the starting points should be close enough to the nondominated solutions to have at least quadratic convergence. Therefore, in this study, by applying the convex hull of the individual minimums (CHIMs), we present a procedure for selecting an appropriate starting point for the quasi-Newton method with the BFGS (Broyden, Fletcher, Goldfarb and Shanno) approximation. Moreover, a new algorithm for constructing a uniform approximation of the Pareto front is presented, which can produce more than one efficient point located on the Pareto front in each iteration. To comprehensively compare the proposed algorithm with existing algorithms, three indices are considered: purity metric, measures of coverage, and spacing metric. Extensive numerical experiments show the significant advantage of the proposed algorithm. Moreover, the obtained boundary approximation follows an almost uniform distribution.

Keywords: multiobjective optimization, quasi-Newton method, stationary point, convergence.

AMS Subject classification: 90C29, 90C30, 90C53

* *Corresponding Author*

1. Introduction

One of the crucial areas in optimization is the discussion of multiobjective optimization problems (MOPs), where there may not be a solution that optimizes all objectives. For this reason, the Pareto optimal concept was presented. The applications of multiobjective programs can be found in various fields, including engineering, economics, management, medicine, machine learning, and others [6, 14, 21, 28, 52, 53, 55]. Therefore, different solution approaches such as stochastic methods [48], evolutionary algorithms [10, 12], or interactive methods [10, 12, 37] are of particular importance to solve these type of problems. Another category of solution methods for MOPs is the scalarization approach. In this approach, the MOP is replaced by a suitable scalarization problem that contains multiple parameters and/or additional constraints. Recently, this theory has been widely developed [1, 2, 11, 17, 24, 25, 43]. Note that solving the single-objective problem (SOP) is much easier than solving the corresponding MOP. However, the main obstacle that arises in this type of solution approach for the decision-maker is to determine the appropriate parameters before solving the SOPs. Because choosing the proper parameters is a complicated or sometimes impossible process for some problems, see the example provided in [18], an efficient strategy to solve these problems is to use non-parametric methods based on gradients, which do not require the selection of parameters by the decision-maker.

Non-parametric methods or non-scalarization methods, are generalizations of directional single-objective optimization methods such as steepest descent, Barzilai and Borwein, Newton, and quasi-Newton methods. For example, Mukai [41] for the first time, generalized the descent method for MOPs. Then, Fliege and Svaiter [19] presented a steepest descent method for unconstrained multicriteria optimization and a feasible descent direction method for the constrained case. In recent years, this work has spurred motivation for research in this field. Nevertheless, empirical evidence indicates that the Armijo line search frequently yields a very small step size along the steepest direction, resulting in a significant deceleration of convergence. The problem arises from the imbalance among the objective functions. To overcome this drawback, Barzilai-Borwein descent method for multiobjective optimization (BBDMO) was proposed by Chen et al [8]. BBDMO, as a first-order method, effectively manages imbalances among objective functions, making it a potential candidate for multitask learning within the realm of multiobjective optimization. Drummond and Iusem [16] provided a projected gradient method for vector optimization problems. In MOPs, the possibility of obtaining a good approximation has become a stimulus for developing these types of methods (see e.g. [3, 7–9, 18, 20, 22, 31–33, 35, 39, 40, 44, 46, 50]).

One of the well-known methods in single-objective optimization is Newton method, in which a Hessian matrix with a positive definite condition is used for the decreasing direction. In 2009, this method was extended to MOPs [18]. In practice, Newton method is rarely used for large-scale problems because it requires calculation, storage, and calculation of the inverse of the Hessian matrix, which imposes an enormous computational cost. To solve these problems, Qu et al., [44] and Povalej [46] extended quasi-Newton method for MOPs using the BFGS approximation. On the other hand, Newton method starts from a point close enough to the solution that has a convergence rate of at least quadratic. However, for points far from the solution, the direction produced by this method is not necessarily decreasing because the Hessian matrix is not necessarily positive definite. Quasi-Newton method is one of the most common approaches for solving unconstrained single-objective optimization problems, which has been extended to multiobjective optimization problems. In the quasi-Newton method, the search direction is computed based on a second-order convex model of the objective function, where the actual Hessian is replaced by approximations at each iteration. Well-known multiobjective versions of quasi-Newton methods, including BFGS, self-scaling BFGS (SS-BFGS), and Huang BFGS (H-BFGS), have been considered [30]. A weakness of some algorithms presented for Newton and quasi-Newton methods is the selection of a set of starting points randomly or uniformly at the beginning of the algorithm. Moreover, the approximation obtained from these methods does not completely cover the Pareto frontier.

Warm-start is a widely used strategy in optimization that involves initializing the algorithm with a solution derived from a related or previously solved problem. This technique was initially introduced in the context of the simplex method for linear programming. In modern optimization, warm-start plays a crucial role in accelerating convergence, especially in iterative methods such as alternating direction method of multipliers (ADMM), gradient-based approaches, and interior-point algorithms. By starting from an informed initial guess rather than a random point, warm-start reduces computational effort and enhances stability, particularly in scenarios involving sequential or large-scale problem instances [15, 42, 54].

The use of warm-start in nonlinear optimization, particularly in Newton and quasi-Newton methods, is an important technique for improving efficiency and convergence speed [5, 42, 54]. The advantages of using warm-start in Newton and quasi-Newton methods are faster convergence and reduced matrix computations.

Here, we present a new quasi-Newton optimization algorithm based on the BFGS method with a warm-start for estimating the Pareto frontier of unconstrained MOPs, which covers the weaknesses mentioned earlier. The new algorithm has some advantages, including:

(i) There is no need to determine the starting points before beginning the algorithm. This is because, during the algorithm's execution, a suitable starting point is generated at each iteration.

(ii) In each iteration of the proposed algorithm, the feasible region of MOP for selecting the starting point is restricted. This reduces the influence of the convexity condition of MOP.

(iii) In each iteration of the algorithm, we can produce more than one efficient point located on the Pareto frontier. In the proposed algorithm for approximating the Pareto frontier, we first consider an unrealistic approximation of the frontier, to generate starting points, which is updated at each iteration. In fact, in subsequent iterations, the unrealistic frontier consists of several unrealistic subregions, each of which is called a CHIM, and with each of CHIMs, an efficient point is generated.

(iv) In the proposed algorithm, initiating optimization from a warm-start near the optimal solution decreases the number of iterations needed for convergence.

(v) Quasi-Newton methods involve computing or approximating the Hessian or gradient. In the proposed algorithm, choosing a suitable warm-start point lessens the computational load and speeds up the calculations.

The proposed algorithm was tested on several benchmark problems, and its performance was compared with that of the BBDMO algorithm with max-type nonmonotone line search in [8], Newton algorithm in [18], and quasi-Newton algorithm using BFGS method in [44]. In the numerical comparison section, we have used three indicators, namely, purity metric, measures of coverage, and spacing metric, for comparison to show the better performance of the proposed algorithm. Also, the relationship of the stationary point with (weakly) Pareto optimal solutions of the MOPs is presented. The next sections of the paper are organized as follows: In Sect. 2, some preliminaries and basic definitions are provided. In Sect. 3, the principal results, including the proposed algorithm, are presented. In Sect. 4, the performance of the suggested algorithm is shown by some test problems. Finally, the obtained results are summarized in Sect. 5.

2. Preliminaries

In this section, we present several definitions that will be used throughout the remainder of the paper. Multiobjective optimization problems involve multiple conflicting objective functions and typically do not yield a single optimal solution, but rather a set of possible solutions. An unconstrained multiobjective optimization problem is formulated as follows:

$$MOP: \ \min_{x\in U} F(x) = (F_1(x), F_2(x), ..., F_p(x)), \tag{2.1}$$

where $U \subseteq \mathbb{R}^n$ is a nonempty and open feasible set and $F : U \longrightarrow \mathbb{R}^p, (p \geq 2)$ includes the objective functions that are continuously differentiable ($F \in C^1(U, \mathbb{R}^p)$). We denote the Jacobian matrix of F at $x \in U$ by $JF(x) \in \mathbb{R}^{p\times n}$. For all $j \in \{1, 2, \ldots, p\}$, the jth row of the Jacobian matrix is $\nabla F_j(x)^T$. The image of U under F is denoted by Y. We define the natural ordering cone as $\mathbb{R}^p_{\geqq} = \{x \in \mathbb{R}^p : x_i \geqslant 0, i = 1, 2, \ldots, p\}$. For $y, \hat{y} \in \mathbb{R}^p$,

$$\hat{y} - y \in int(\mathbb{R}^p_{\geqq}) \Leftrightarrow y_k < \hat{y}_k, \forall k = 1, 2, \ldots, p,$$

$$\hat{y} - y \in \mathbb{R}^p_{\geqq} \Leftrightarrow y_k \leq \hat{y}_k, \forall k = 1, 2, \ldots, p,$$

$$\hat{y} - y \in \mathbb{R}^p_{\geqq}\backslash\{0\} \Leftrightarrow y_k \leq \hat{y}_k \text{ but } y \neq \hat{y}.$$

Definition 1. Let $U \subset \mathbb{R}^n$ be a convex set. Then,

(i) $F : U \to \mathbb{R}^p$ is $\mathbb{R}^p$-convex, if F is componentwise convex [26].

(ii) $F : U \to \mathbb{R}^p$ is $\mathbb{R}^p$-strictly convex, if F is componentwise strictly convex.

Now, we present the concept of optimality in multiobjective optimization problems under the title of efficient and weakly efficient solutions.

Definition 2. A feasible solution $\hat{x} \in U$ is called

(i) an efficient (a Pareto optimal) solution for MOP (2.1), if there is no other $x \in U$ such that $F(x) \leq F(\hat{x})$,

(ii) a weakly efficient (a weakly Pareto optimal) solution for MOP (2.1), if there is no other $x \in U$ such that $F(x) < F(\hat{x})$.

(iii) a local (weakly) efficient solution for MOP (2.1), if there exists a neighborhood $N[\hat{x}, r] \subset U$ of $\hat{x}$ with radius r, so that $\hat{x}$ is an (a weakly) efficient solution on $N[\hat{x}, r]$.

The set of all (local) efficient and (local) weakly efficient solutions are denoted, respectively, by $U_{(L)E}$ and $U_{(L)WE}$. Their images in the space $\mathbb{R}^p$ are called, respectively, (local) nondominated and (local) weakly nondominated solutions and are denoted by Y_N and Y_{WN}, respectively.
Suppose that $x_j^* = \arg\min_{x \in U} F_j(x),\ \forall j = 1, 2, ..., p$. Similar to [2, 11], let $\Phi = (F(x_1^*), \ldots, F(x_p^*))_{p \times p}$ be a pay-off matrix, where $F(x_j^*)$ is the jth column. Then $\{\Phi\beta | \beta \in \mathbb{R}^p, \sum_{j=1}^{p} \beta_j = 1, \beta_j \geq 0\}$ is the set of all points in $\mathbb{R}^p$ that are convex combinations of $F(x_j^*)$ for each $j = 1, 2, \ldots, p$, and is called the convex hull of the individual minima.
Assume that $f : U \longrightarrow \mathbb{R}$. A single-objective optimization problem is demonstrated as follows:

$$SOP: \min_{x \in U.} f(x) \tag{2.2}$$

Definition 3. A feasible solution $\hat{x} \in U$ is said to be

(i) an optimal solution for SOP (2.2), if $f(\hat{x}) \leqslant f(x), \forall x \in U$,

(ii) a strictly optimal solution for SOP (2.2), if $f(\hat{x}) < f(x), \forall x \in U \setminus \{\hat{x}\}$.

Definition 4. (i) [46] A point $\hat{x} \in U$ is said to be a stationary (critical) point of the vector function F, if $R(JF(\hat{x})) \subset \mathbb{R}^p_{\geqq}$ or $R(JF(\hat{x})) \cap -int(\mathbb{R}^p_{\geqq}) = \emptyset$, where $R(JF(\hat{x}))$ defines the range (image) space of the Jacobian of the continuously differentiable function F at $\hat{x}$.

(ii) [44] If $\hat{x}$ is a nonstationary point, then, there exists $s \in \mathbb{R}^n$ such that $\nabla F_j(\hat{x})^T s < 0, \forall j \in \{1, 2, \ldots, p\}$.

(iii) Suppose that F is a continuously differentiable function. Then, the directional derivative of F at $\hat{x}$ in the direction s is defined as

$$\lim_{t\to 0} \frac{F_j(\hat{x}+ts) - F_j(\hat{x})}{t} = \nabla F_j(\hat{x})^T s, \qquad \forall j \in \{1, 2, \ldots, p\}.$$

Therefore, if s is a descent direction of F at $\hat{x}$, then there exists $t_0 > 0$ with

$$F(\hat{x}+ts) < F(\hat{x}), \qquad \forall 0 < t \leq t_0.$$

3. Main results

3.1. The quasi-Newton methods

In quasi-Newton methods, instead of calculating the Hessian matrix or its inverse, a positive definite approximation is calculated from them. The Broyden class and the self-scaling Broyden class, are two efficient classes that just use gradient information to approximate the Hessian matrices, and the obtained matrices from these classes are symmetric. These approximations from one iteration to the next one must be updated in such a way that they satisfy the following condition called the secant or quasi-Newton equation:

$$B_k v_k = y_k,$$

where $v_k = x_{k+1} - x_k$, $g_k = \nabla f(x_k)$, with $f : U \longrightarrow \mathbb{R}$ as twice continuously differentiable function on an open set U, $y_k = g_{k+1} - g_k$ and B_k is an approximation of the Hessian matrix at x_k, and $\{x_k\}$ is supposed to be a sequence created by the quasi-Newton method. BFGS formula of the Broyden class can be mentioned among the most efficient quasi-Newton update formulas for single-objective optimization problems [42], which is expressed as follows:

$$B_{k+1} = B_k - \frac{B_k v_k v_k^T B_k}{v_k^T B_k v_k} + \frac{y_k y_k^T}{v_k^T y_k}.$$

In the BFGS update formula, if B_k is a positive definite matrix, then the BFGS update is also positive definite if the following inequality, called the curvature condition, holds:

$$v_k^T y_k > 0.$$

This inequality holds under the Wolfe's linear search conditions:

$$f(x_k + t_k s_k) - f(x_k) \leq \delta t_k g_k^T s_k,$$

$$\nabla f(x_k + t_k s_k)^T s_k \geq \sigma g_k^T s_k,$$

where $0 < \delta < \sigma < 1$ and t_k is the step length in iteration k. Therefore, the directions produced by the BFGS method are descent directions [42, 49]. For single-objective optimization, according to Newton method, the direction of searching for quasi-Newton methods in each iteration is calculated from the following relationship:

$$s_k = -B_k^{-1}\nabla f(x_k).$$

Similar to the development of Newton method for multiobjective problems [18], the quasi-Newton method has been extended by updating BFGS for single-objective optimization to multiobjective optimization problems, with the difference that instead of using the exact Hessian matrix, its approximation was used [44, 46].
To find the quasi-Newton direction in $x \in U$, it is necessary to solve the following subproblem and consider its optimal solution as a search direction for the quasi-Newton method at $x \in U$.

$$\min_{s\in\mathbb{R}^n} \max_{j\in\{1,2,\dots,p\}} \nabla F_j(x)^T s + \frac{1}{2}s^T B_j(x)s, \tag{3.1}$$

where $B_j(x)$ is an approximation of the Hessian matrix of $F_j(x)$. The optimal objective function value for Subproblem (3.1) is denoted by $\theta(x_k)$. We can rewrite Subproblem (3.1) in the following form:

$$\begin{aligned} &\min\ t \\ &s.t.\ \nabla F_j(x)^T s + \frac{1}{2}s^T B_j(x)s \le t, \\ &(t,s) \in \mathbb{R}\times\mathbb{R}^n. \end{aligned} \tag{3.2}$$

The purpose of the following theorem is to express the relationship between (weakly) efficient solutions and stationary points of MOPs. To study the proof of this theorem, we refer to [18].

Theorem 1. *Assume that $F \in C^1(U,\mathbb{R}^p)$.*

(i) If $\hat{x}$ is a locally weak Pareto optimal solution, then it is a stationary point of F.

(ii) If U is convex, F is $\mathbb{R}^p$-convex and $\hat{x} \in U$ is critical for F, then $\hat{x}$ is a weakly Pareto optimal solution.

(iii) If U is convex, $F \in C^2(U,\mathbb{R}^p)$, $\nabla^2 F_j(x) > 0$ for all $j \in \{1,2,\dots,p\}$ and all $x \in U$, and if $\hat{x} \in U$ is critical for F, then $\hat{x}$ is Pareto optimal.

3.2. Determining the step length

In general, the step length t_k in iterative methods for solving single-objective problems is chosen so that the objective function $f : U \longrightarrow \mathbb{R}$ has an acceptable reduction along the descent direction $s(x_k) \in \mathbb{R}^n$, and in other words, $f(x_k) - f(x_k + t_k s(x_k)) > 0$ is a reasonable value. Such a process for determining step length is called approximate line search [42]. Armijo's condition is a suitable criterion for creating an acceptable reduction in the line search process, which is expressed as follows:

$$f(x_k + t_k s(x_k)) - f(x_k) \leq \delta t_k \nabla f(x_k)^T s(x_k).$$

where $0 < \delta < 1$, is given. This condition clarifies that the decrease of the function f must be consistent with the step length t_k and the directional derivative $\nabla f(x_k)^T s(x_k)$.
Likewise, the above inequality can be used for multiobjective optimization problems as follows:

$$F_j(x_k + t_k s(x_k)) - F_j(x_k) \leq \delta t_k \theta(x_k), \qquad \forall j = 1, \dots, p,$$

where $\theta(x_k)$ is the optimal objective function value of Subproblem (3.1). In papers [46] and [44], the above condition is used in the line search process of the BFGS quasi-Newton method. The above inequality will be a scale for accepting the step length in the quasi-Newton direction for multiobjective problems, and the following theorem guarantees the existence of such a step length. The proof of the following theorem is similar to Corollary 3.3 in [18], by the difference that, instead of the Hessian matrix, an approximation of it will be used here.

Theorem 2. *Let $\hat{x}$ be a nonstationary point of F. Then, for any $0 < c < 1$, there is $0 < t_0 \leqslant 1$ and $r > 0$ such that $\forall x \in N[\hat{x}, r] \subset U$,*

$$x + ts(x) \in U \text{ and } F_j(x + ts(x)) \leq F_j(x) + ct\theta(x), \tag{3.3}$$

for all $j = 1, 2, \dots, p$ and $\forall\ 0 \leqslant t \leqslant t_0$.

Proof. Utilizing item c of Lemma 2 in [44] (continuity of θ), there is $r > 0$ such that

$$\theta(x) < \theta(\hat{x})/2 \leq 0, \qquad \forall x \in N[\hat{x}, r] \subset U.$$

Then, each $x \in N[\hat{x}, r] \subset U$ is a nonstationary point and since U is an open set, there exists $0 < t_0 \leqslant 1$ such that

$$x + ts(x) \in U, \qquad \forall 0 \leqslant t \leqslant t_0.$$

According to the Taylor series of function F_j, $\forall j \in \{1, 2, \dots, p\}$, we can write

$$F_j(x + ts(x)) = F_j(x) + t\nabla F_j(x)^T s(x) + o_j(\|ts(x)\|), \qquad \forall x \in N[\hat{x}, r], \tag{3.4}$$

in which, $\frac{o_j(\|ts(x)\|)}{t\|s(x)\|} \underset{t\to 0}{\longrightarrow} 0$. Utilizing item 3 of Theorem 5 in [38], $s(x)$ is bounded on $N[\hat{x}, r]$. Then, there is $M > 0$ such that $\|s(x)\| < M$. Therefor, $\frac{o_j(\|ts(x)\|)}{t} \underset{t\to 0}{\longrightarrow} 0$. Since, the approximation of the Hessian matrix is positive definite, we have

$$\nabla F_j(x)^T s(x) \leq \nabla F_j(x)^T s(x) + \frac{1}{2} s(x)^T B_j(x) s(x) \leq \theta(x), \qquad \forall j = 1, 2, \ldots, p. \tag{3.5}$$

From relations (3.4) and (3.5), we have

$$F_j(x + ts(x)) \leq F_j(x) + ct\theta(x) + t((1-c)\theta(x) + \frac{o_j(\|ts(x)\|)}{t}), \qquad \forall t \in [0, t_0].$$

Since $\theta(x) \leq 0$, for $t \in [0, t_0]$ sufficiently small, we have

$$F_j(x + ts(x)) \leq F_j(x) + ct\theta(x).$$

Therefore, the proof is completed. □

In the following lemma, we present two inequalities in order to prove the convergence of the BFGS quasi-Newton method concerning any nonstationary point of F.

Lemma 1. *Let $\hat{x}$ be a nonstationary point of F. Then,*

(i) there is $0 < t_0 \leqslant 1$ such that

$$t\nabla F_j(\hat{x})^T s(\hat{x}) + \frac{1}{2} t^2 s(\hat{x})^T B_j(\hat{x}) s(\hat{x}) < 0, \tag{3.6}$$

for all $j = 1, 2, \ldots, p$ and $\forall 0 \leqslant t \leqslant t_0$.

(ii) for any $0 < c < 1$, there is $0 < t_0 \leqslant 1$ and $r > 0$ such that $\forall x \in N[\hat{x}, r] \subset U$

$$F_j(x) - F_j(x + ts(x)) \geq -c(t\nabla F_j(x)^T s(x) + \frac{1}{2} t^2 s(x)^T B_j(x) s(x)), \tag{3.7}$$

for all $j = 1, 2, \ldots, p$ and $\forall 0 \leqslant t \leqslant t_0$.

Proof. (i) By contradiction, assume that relation (3.6) is not satisfied. Then, there exist $k_i \in \mathbb{R}^p_{\geqq}$ and $0 < t_0 \leqslant 1$ such that

$$t\nabla F_j(\hat{x})^T s(\hat{x}) + \frac{1}{2} t^2 s(\hat{x})^T B_j(\hat{x}) s(\hat{x}) = k_j, \qquad \exists 0 < t \leqslant t_0, \ \exists j \in \{1, 2, \ldots, p\}.$$

Since $\hat{x}$ is a nonstationary point of F, we have

$$\theta(\hat{x}) \geq \frac{1}{t}(k_j + \frac{t - t^2}{2} s(\hat{x})^T B_j(\hat{x}) s(\hat{x})) \geq 0, \qquad \forall 0 < t \leqslant t_0, \ \forall j = 1, 2, \ldots, p$$

a contradiction to nonstationarity of $\hat{x}$ for F.

(ii) According to the Taylor series of function $F_j, \forall j \in \{1, 2, \ldots, p\}$, we have

$$F_j(x + ts(x)) = F_j(x) + t\nabla F_j(x)^T s(x) + o_j(\|ts(x)\|, \qquad \forall x \in N[\hat{x}, r] \quad (3.8)$$

in which $\frac{o_j(\|ts(x)\|)}{t\|s(x)\|} \underset{t \to 0}{\longrightarrow} 0$. Once again, employing the boundedness of $s(x)$ on $N[\hat{x}, r]$, we deduce that $\frac{o_j(\|ts(x)\|)}{t} \underset{t \to 0}{\longrightarrow} 0$ uniformly for $x \in N[\hat{x}, r]$. Since, the approximation of the Hessian matrix is positive definite, we have

$$\nabla F_j(x)^T s(x) \leq \nabla F_j(x)^T s(x) + \frac{1}{2} ts(x)^T B_j(x) s(x) = \gamma_j(x, t), \qquad \forall x \in N[\hat{x}, r]. (3.9)$$

From relations (3.8) and (3.9), for all $t \in [0, t_0]$, we have

$$F_j(x) - F_j(x + ts(x)) \geq -(ct\gamma_j(x, t) + t((1 - c)\gamma_j(x, t) + \frac{o_j(\|ts(x)\|)}{t})).$$

Since $t\gamma_j(x, t) < 0$, for $t \in [0, t_0]$ sufficiently small and $\forall x \in N[\hat{x}, r]$, we obtain

$$F_j(x) - F_j(x + ts(x)) \geq -c(t\nabla F_j(x)^T s(x) + \frac{1}{2} t^2 s(x)^T B_j(x) s(x)),$$

for all $j = 1, 2, \ldots, p$ and $\forall 0 \leqslant t \leqslant t_0$.

□

From Part 1 of Definition 4, the following lemma can be presented [46].

Lemma 2. *$\hat{x}$ is stationary point of F if* $\psi(\hat{x}) = \sup_{\|s\| \leq 1} \max_{j \in \{1, 2, \ldots, p\}} \{-\nabla F_j(x)^T s\} = 0.$

The following theorem has been proved in [46] (Theorem 5) by the following assumptions A_1 and A_2.
A_1**:** Assume that the level set $L_0 = \{x \in \mathbb{R}^n \ : \ F(x) \leq F(x_0)\}$ is bounded.
A_2**:** Assume that for sufficient large k, the step length $t_k = 1$ is accepted.
Now, we can replace the assumption A_2 with the following assumption $\hat{A}_2$ and provide a sufficient condition for global convergence.
$\hat{A}_2$**:** Assume that for sufficiently large k, the step length $t_k = \frac{1}{2^k}$ with $0 < t_k \leq 1$, is accepted and satisfies conditions (3.3).

Theorem 3. *Suppose that there exists $\beta > 0$ such that $\|B_j(x)\| \leq \beta$, for all $j \in \{1, 2, \ldots, p\}$ and all $x \in \{z \in \mathbb{R}^n : F(z) \leq F(x_0)\}$. Then the sequence $\{x^k\}$ converges to an stationary point of F.*

Proof. It suffices to prove that the limit point of the sequence $\{x^k\}$ satisfies in the necessary condition of Lemma 2. Let s^k be an optimal solution of $\psi(x^k)$. Then, by Lemma 1 and relation (3.9), for any $0 < c < 1$ there exists $0 < t_0 \leqslant 1$ such that equation (3.7) holds:

$$F_j(x^k) - F_j(x^k + t^k s(x^k)) \geq -ct^k\gamma(x^k, t^k) =$$

$$-ct^k(\nabla F_j(x^k)^T s(x^k) + \frac{1}{2}t^k s(x^k)^T B_j(x^k)s(x^k)),\ \forall 0 \leqslant t^k \leqslant t_0,\ \forall j.$$

Moreover, we can write

$$\begin{aligned} \max_{j\in\{1,2,\ldots,p\}}\{F_j(x^k) - F_j(x^k + t^k s(x^k))\} &\geq \max_{0\leqslant t^k\leqslant t_0}\max_{j\in\{1,2,\ldots,p\}}\{-ct^k\gamma(x^t, t^k)\}, \\ &\geq \max_{0\leqslant t^k\leqslant t_0}\{c(t^k\psi(x^k) - \frac{1}{2}(t^k)^2\beta)\} \geq \frac{c\psi(x^k)}{2}\min\{t_0, \frac{\psi(x^k)}{\beta}\}. \end{aligned} \tag{3.10}$$

According to Assumption A_1, we assume that $\{x^k\}_{k\in\mathbb{N}}$ is a subsequence that converges to $\hat{x}$. Then, from Assumption $\hat{A}_2$, there exists a $k_0 \in \mathbb{N}$ that for every $k > k_0$ we have an enough large step length $t_k = \frac{1}{2^k}$ such that

$$\begin{aligned} \sum_{k>k_0}\frac{c\psi(x^k)}{2}\min\{t_0, \frac{\psi(x^k)}{\beta}\} &\leq \sum_{k>k_0}\max_{j\in\{1,2,\ldots,p\}}\{F_j(x^k) - F_j(x^{k+1})\}, \\ &\leq \max_{j\in\{1,2,\ldots,p\}}\{F_j(x^0) - F_j(\hat{x})\} < \infty, \end{aligned} \tag{3.11}$$

where the last relation follows from Theorem 5 in [46]. Now, to complete the proof, it suffices to prove $\psi(\hat{x}) = 0$. Suppose that $\psi(\hat{x}) > 0$, so $\psi(x^k) \geq \alpha > 0$ holds for some $\alpha > 0$ and $\varepsilon_0 > 0$ with $\left\|x^k - \hat{x}\right\| \leq \varepsilon,\ \forall\varepsilon \geq \varepsilon_0,\ k > k_0$. Therefore, we can write

$$\sum_{k>k_0}\psi(x^k)\min\{t_0, \frac{\psi(x^k)}{\beta}\} \geq \sum_{k\in\{k:\ \|x^k-\hat{x}\|\leq\varepsilon,\varepsilon\geq\varepsilon_0,k>k_0\}}\alpha\min\{t_0, \frac{\alpha}{c}\} = \infty.$$

This contradicts the inequality (3.11). □

3.3. Determining warm-starting point

In almost all of the presented nonlinear methods in the literature, the obtained optimal solutions are dependent on the starting point and generally for different starting points, distinct optimal solutions are generated. Therefore, in each nonlinear problem, the starting points are chosen uniformly between the upper and the lower bounds [8, 22, 23, 38]. By this selection, the initial point may be far away from the feasible region, and eventually, it may not converge to the critical point. Moreover, the generated efficient solution set may not cover all of the optimal feasible region. To

overcome the mentioned weaknesses, we present the following method for generating starting points.
The suggested method is based on changing the CHIM to generate warm-starting points according to a particular procedure at each step. At first, the initial CHIM is defined according to the NBI method [11], and in each iteration of the algorithm, the current CHIM will be updated [2].
Consider the $CHIM = \{\Phi\beta|\beta \in \mathbb{R}^p,\ \beta_k = \frac{1}{p}\}$, where $\Phi = (F(x_1^*), ..., F(x_p^*))_{p\times p}$ is the pay-off matrix and x_k^* for each $k = 1, \dots, p$ is the individual minimum of the function $F_k(x)$. Now, suppose that $a_{k,i} = \Phi_{\mathrm{k,i}}\beta^T$ on the $CIHM_i^k$ in step k of iteration $i = 1, \dots, p^{k-1}$, then we select the starting point $x_{k,i}^0$ as follows:

$$x_{k,i}^0 := \arg\min_{x\in U} \|F(x) - a_{k,i}\|_{2or\infty}. \tag{3.12}$$

In the next subsection, applying the presented warm initial point, we present a modified version of the quasi-Newton algorithm for multiobjective optimization problems.

3.4. Quasi-Newton algorithm for multi-criteria optimization

In this subsection, we extend the quasi-Newton algorithm by using a new algorithm. Although by starting from a point close enough to the optimal solution (warm-start), Newton method has convergence with order of least quadratic, but for starting points far from the optimal solution, the direction produced by this method is not necessarily decreasing, because the Hessian matrix far from the solution is not necessarily positive definite. On the other hand, Newton method is rarely used for large-scale problems, because it requires calculating, storing and inverting the Hessian matrix, which imposes a high computational cost [49]. To overcome these difficulties, we propose a new algorithm, called Algorithm 1, to produce an approximation of the Pareto front. The new algorithm is based on CHIMs in the objective space Y. At first, we choose a point on the current CHIM and obtain its image in the feasible set U, using Subproblem (3.12). Then, we consider the obtained point as the warm starting point for quasi-Newton process. After obtaining a Pareto optimal point on the true Pareto frontier using the quasi-Newton method, we update the current CHIM. We repeat this systematic process until Algorithm 1 reaches an approximation of the true Pareto frontier. In this method, approximation of the Hessian matrix is considered by using the values and gradients of the objective functions. In this method, we use the BFGS formula of the Broyden class to approximate the Hessian matrix B_k and generate a sequence of $\{B_k\}$ and a sequence of decreasing directions $\{s_k\}$.
Using the proposed algorithm for determining the starting point in quasi-Newton methods, increases the convergence speed and reduces the number of iterations, as it provides a point close to the Pareto frontier. Unlike scalarization methods based on CHIMs [2, 24] that require the selection of appropriate weights, this approach effectively guides the optimization process without the need for explicit scalarization.

Algorithm 1 includes three steps. In the first step, the necessary inputs are appropriately selected and N_2 is chosen such that the algorithm runs long enough for the stopping condition to be satisfied. The second step in each iteration determines the parameter a and the warm-starting point x^0 from solving Subproblem (3.12) utilizing the idea given in subsection 3.3, where $a_{k,i} = \Phi_{k,i}\beta^T$ on the $CHIM_i^k$ and $x_{k,i}^0 := \arg\min_{x\in U} \|F(x) - a_{k,i}\|_{2or\infty}$, in step k of iteration i. Then, to generate the quasi-Newton descent direction $s_{k,i}(x)$ in step k of iteration i, Subproblem (3.2) is solved. Next, according to Theorem 2, the step length is determined by the modified Armijo condition. At the end of the third step, we update the point x_m and approximate the Hessian $B_{j,k}(x_m)$ by using the BFGS method. The general scheme of the procedure is given in Algorithm 1. We note that in this algorithm, with k iterations, the expected number of efficient points at the Pareto frontier is $\dfrac{p^k - 1}{p-1}$.
The key idea behind Algorithm 1 is generating starting points using CHIM and then applying the proposed quasi-Newton algorithm to refine the solution. While the quasi-Newton method is a popular choice due to its efficiency in many cases, other optimization methods can also be used in the local optimization step. This flexibility makes the algorithm versatile and applicable to a broader range of optimization problems.
Finally, by choosing each starting point selected in CHIM, the quasi-Newton process converges to a Pareto point according to Theorem 3 in a finite number of iterations. Therefore, the method is well-defined.

Algorithm 1 Approximating the Pareto front by the BFGS quasi-Newton method.

(Initialization)
Let N_1 and N_2 be the number of runs.
Select a sufficient small positive scalar $\varepsilon > 0$ and $\sigma \in (0,1)$. For each index $j \in \{1, 2, \ldots, p\}$, we consider matrix $B_{j,0}$ equal to identity matrix I.
(Main loop)
Determine the parameter $a_{k,i}$ on the $CHIM_i^k$ to generate the initial decision
vector $x_{k,i}^0$ chosen from U.
for $k := 1, 2, \cdots, N_1$ **do**
    Suppose that we would like the $mCHIM$ to be refined k times,
    therefore $mCHIMs := \{CHIM_1^k, \cdots, CHIM_{p^{k-1}}^k\}$.
    for $i := 1, 2, \cdots, p^{k-1}$ **do**
        Let $a_{k,i} := \Phi_{k,i}\beta^T$.
        Solve Subproblem (3.12) to obtain start point.
        for $m := 1, 2, \cdots, N_2$ **do**
            Find $s_{k,i}(x_m)$ and $\theta(x_m)$ as the optimal solution of Subproblem (3.2).
            if then $|\theta(x_m)| \leq \varepsilon$
                Stop and set $y_{k,i} := F(x_m)$.
            else
                Find the largest $t_m = \frac{1}{2^n}, n \in \mathbb{N}$ which satisfies in the following

conditions:

$$x_m + t_m s(x_m) \in U,$$

$$F_j(x_m + t_m s(x_m)) \leq F_j(x_m) + \sigma t_m \theta(x_m).$$

So, set $x_{m+1} := x_m + t_m s(x_m)$ and update the positive definite matrix
$B_j(x_{m+1}),\ \forall j \in \{1, 2, \ldots, p\}$ with BFGS method.
Set $m := m + 1$.
end if
end for
Let $Y_k := \{y_{k,1}, y_{k,2}, \ldots, y_{k,p^{k-1}}\}$. Therefore, update current mCHIM.
end for
end for

Output: Set $Y := \bigcup_{k=1,\ldots,N_1} Y_k$. Hence, Y is an approximation of the Pareto front.

3.5. Evaluation of the algorithms

In general, the Pareto front contains infinite nondominated points. Therefore, the researchers try to construct a reasonable discrete approximation of the Pareto front by producing a minimal number of Pareto points. It is fundamental that the suggested algorithm be capable to construct an even approximation of the whole Pareto front. There are many indicators in the literature that evaluate the efficiency of the algorithms. Among them, we utilize indicators such as purity metric, measures of coverage, and spacing metric [4, 34, 36], to compare the algorithm proposed in this paper with some existing algorithms.

Measures based on the position of the nondominated front [4]:

The purity index is used to compare and evaluate the approximate Pareto frontier obtained using different methods. Let $P_1, P_2, \ldots, P_N$ be N approximation sets of the Pareto optimal set for N different algorithms to the same problem. Then, after removing the dominant points compared to other points from set $\mathbb{P} := \bigcup_{i \in \{1,2,\ldots,N\}} P_i$, we consider the obtained set as an approximation of the Pareto frontier and call it the reference set. So, the measure of purity is expressed as:

$$PM_i = \frac{|\mathbb{P} \cap P_i|}{|\mathbb{P}|}, \qquad \forall i \in \{1, 2, \ldots, N\}.$$

The value of PM_i is between zero and one, and according to the above relationship, the higher the value of PM_i, the more Pareto points are obtained by method $i \in \{1, 2, \ldots, N\}$.

Measures of coverage of the Pareto front [34]:

Assume that $P_1, P_2, \dots, P_N$ are N approximation sets of the Pareto optimal set for N different algorithms to the same problem. To evaluate the performance of an algorithm for approximating the Pareto front, coverage measurement is used to show that to what extent it covers different parts of the Pareto frontier. First, the set of reference points as

$$\mathcal{P} = \{(U_1, L_2, \cdots, L_p), \cdots, (L_1, \cdots, L_{p-1}, U_p)\}$$

is considered, in which $U_j = \max\limits_{x \in \bigcup\limits_{i=1,\dots,N} P_i} (f_j(x))$ and $L_j = \min\limits_{x \in \bigcup\limits_{i=1,\dots,N} P_i} (f_j(x))$.
Now, $d_r^{P_i} = \min\{d(s_r, s) | s \in P_i\}$ is defined as the distance in the objective space between a reference solution $s_r \in \mathcal{P}$ to the set P_i. So, the spread of P_i is defined as follows:

$$EX_i = \sqrt{\sum_{r=1,\dots,p} (d_r^{P_i})^2 / p}.$$

A smaller EX_i means P_i has a better well extended Pareto frontier.
Spacing metric [36]:
One of the most desirable features of the optimal points obtained from an algorithm is the uniform distribution, which indicates that no region of the optimal boundary has the lowest or highest density of generated points. Here, we utilize a criterion to measure the uniformity of the produced optimal points. For this purpose, for each point d_i produced on the optimal boundary, the smallest and the largest spheres (circles for two dimensions) with diameters d_i^l and d_i^u, respectively, are constructed that can be formed between d_i and another point of the set of points produced on the optimal boundary. So, the diameter of each sphere is equal to the distance between the two points. Moreover, there is no other point in these two spheres. The measure of uniformity is expressed as:

$$EV = \sigma_d / \hat{d},$$

in which $\hat{d}$ and σ_d are the mean and standard deviation of $d = \{d_1^l, d_1^u, \cdots, d_n^l, d_n^u\}$. A set of points is precisely evenly distributed when $EV = 0$.

4. Numerical Results

In this section, the numerical performance of Algorithm 1 is compared with the BB-DMO algorithm with max-type nonmonotone line search in [8], Newton algorithm in [18], and Quasi-Newton algorithm using BFGS method in [44] based on the number of known test problems, explained in Table 1. All test problems, described in Table 1, are solved by Algorithm 1, Newton algorithm and quasi-Newton algorithm equipped with Armijo's condition. Similar to [18], as test problems have box constraints of the form $x_L \leq x \leq x_U$, we solve the subproblem (3.1) with an additional box constraint in the form $x_L - x \leq s \leq x_U - x$. The inclusion of box constraints is considered in the numerical experiments to reflect practical scenarios where such constraints are often present [18, 38, 44, 46]. While the theoretical analysis remains general. For the BBDMO algorithm, we select $\alpha_{min} = 10^{-3}$, $\alpha_{max} = 10^3$, and $M = 10$ in max-typ nonmonotone line search in [8]. We set the maximum number of iterations to $N_2 = 500$. The stopping criterion is $|\theta(x_m)| \leq \varepsilon$, where $\varepsilon = 10^{-4}$.

In the presented research, the algorithms have been implemented in MATLAB (R2018a). CVX package solver has been used for solving the direction search Subproblem (3.2). The test problems have been performed on a core i7-9700H processor CPU with 3 GHz and 32 GB RAM.

Algorithms considered for comparison with each other, are implemented on several test problems listed in the first column of Table 1. The dimensions of the variables and box constraints, which represent the lower bound x_L and the upper bound x_U of the variables, are presented in the third to fifth columns of Table 1, respectively. The proposed algorithm is primarily designed for convex multi-objective optimization problems, but because of its special structure, this algorithm can also be used for solving non-convex problems.

Table 1. Introducing all test problems to compare the iterative methods of Newton, quasi-Newton, BBDMO and the new Algorithm. Note: $e = (1, .., 1)$ is a $1 \times n$ vector.

Problems	Sources	n	x_L	x_U
JOS1a	[27]	600	$0e$	e
JOS1b	[27]	600	$-2e$	$2e$
JOS1c	[27]	600	$-10e$	$10e$
JOS1d	[27]	600	$-50e$	$50e$
JOS1e	[27]	600	$-100e$	$100e$
JOS2a	[27]	2	$0e$	e
JOS2b	[27]	2	$-2e$	$2e$
JOS2c	[27]	2	$-10e$	$10e$
JOS2d	[27]	2	$-50e$	$50e$
JOS2e	[27]	2	$-100e$	$100e$
KW2	[29]	2	$-[3,3]$	$[3,3]$
PNR	[45]	2	$-[2,2]$	$[2,2]$
WIT1	[51]	2	$-[2,2]$	$[2,2]$
WIT2	[51]	2	$-[2,2]$	$[2,2]$
WIT3	[51]	2	$-[2,2]$	$[2,2]$
WIT4	[51]	2	$-[2,2]$	$[2,2]$
WIT5	[51]	2	$-[2,2]$	$[2,2]$
WIT6	[51]	2	$-[2,2]$	$[2,2]$
DTLZ2	[13, 47]	3	$[0,0,0]$	$[1,1,1]$
COMET	[13]	3	$[1,-2,0]$	$[3.5,2,1]$

Note that among the test problems listed in Table 1, Problems JOS1a-e, JOS2a-e and WIT1-6 are convex.

Example 1. This test problem has been studied in [27] and has a convex Pareto front and is formulated as follows:

$$\mathbf{JOS}: \min_{x\in\mathbb{R}^n} \left(\frac{1}{n}\sum_{i=1}^{n} x_i^2, \frac{1}{n}\sum_{i=1}^{n}(x_i-2)^2\right). \tag{4.1}$$

First, in the second step of Algorithm 1, we obtain the starting point for each $CHIM_i^k$, and then we solve Subproblem (3.2) to obtain the search direction vector to determine the starting point for the next iteration on the current $CHIM_i^k$. The starting points for the BBDMO algorithm in [8], Newton algorithm in [18], and quasi-Newton algorithm for solving the BFGS method in [44] are chosen uniformly between the upper and lower bounds.

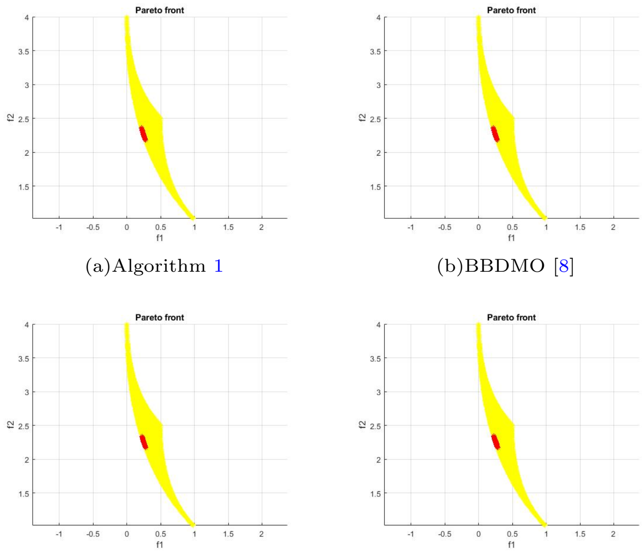
Figures 1-10 show numerical results in value space obtained by Algorithm 1 with $k = 6$, and $N = 2^6 - 1$ starting points with uniform distribution for the Newton algorithm, the quasi-Newton algorithm, and the BBDMO with max-type nonmonotone line search for problem JOS. As can be seen in part (a) of these figures, an almost uniform approximation is obtained against Parts b-d. By comparing Figures 1 and 2 for problems JOS1a and JOS2a with $0 \le x_i \le 1, \forall i = 1, \ldots, n$, we find that as the value of n increases for BBDMO, Newton and quasi-Newton algorithms, the coverage of the Pareto frontier decreases. Similarly, in Figures 3 to 10 for problems JOS1b to JOS1e and JOS2b to JOS2e, for some of the algorithms mentioned above, the coverage of the Pareto frontier decreases by increasing the parameter n.

It should be mentioned that the Pareto frontier of JOS2b to JOS2e is obtained by Algorithm 1 with a good approximation, and the CPU time of Algorithm 1 is close to the CPU time of other algorithms.

Table 2 shows the results obtained from Algorithms 1, BBDMO, Newton, and quasi-Newton methods, respectively. According to the obtained values of EV, EX and PM indices, we find that Algorithm 1 performs better for JOS problem.

Table 2. Numerical results including average CPU time, EX, EV and PM of Algorithm 1, Newton algorithm, quasi-Newton algorithm using BFGS method, and BBDMO with max-type nonmonotone line search.

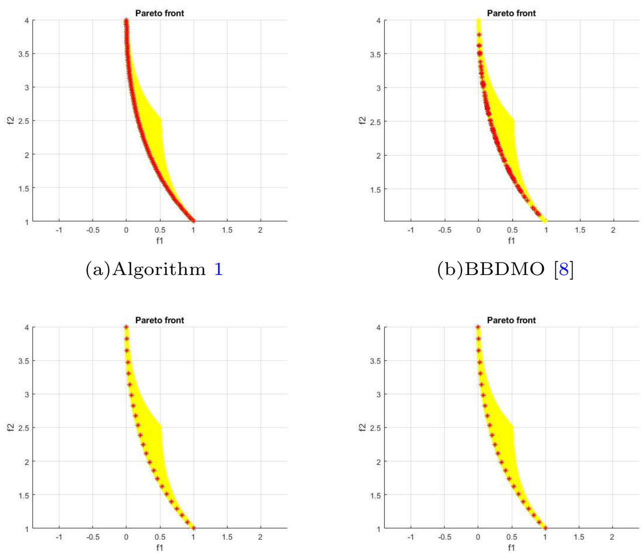
Problem	Algorithm	CPU time (s)	EV	EX	PM
JOS1a	Algorithm 1	1.33	0.09	0.5	1
	Newton	1.1	1.1315	1.42	1
	Quasi-Newton	0.41	0.63	1.42	0.008
	BBDMO	0.1	1.83	1.4	1
JOS1b	Algorithm 1	0.68	0.56	6.0298e-12	1
	Newton	1.4	1.94	2.7	1
	Quasi-Newton	0.88	1.9	2.302	0.008
	BBDMO	0.1	1.85	2.4	0.5
JOS1c	Algorithm 1	0.55	0.5	7.9930e-10	1
	Newton	1.4	2.6	2.31	1
	Quasi-Newton	0.87	0.7	2.311	0.008
	BBDMO	0.1	2.11	2.3	1
JOS1d	Algorithm 1	0.84	0.5	1.5693e-08	1
	Newton	1.35	2.4	2.31	1
	Quasi-Newton	0.87	1.1	2.31	0.008
	BBDMO	0.1	2	2.3	0.7
JOS1e	Algorithm 1	1.6	0.5	9.7479e-17	1
	Newton	1.32	2.4	2.31	1
	Quasi-Newton	0.88	0.8	2.31	0.008
	BBDMO	0.1	2.4	2.3	0.5



(a)Algorithm 1 (b)BBDMO [8]

(c)Newton algorithm [18] (d)Quasi-Newton algorithm [44]

Figure 1. **Example 1: Comparison of the existing algorithms for the JOS1a test problem**

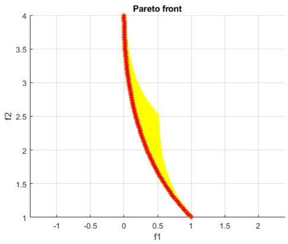


(a)Algorithm 1 (b)BBDMO [8]

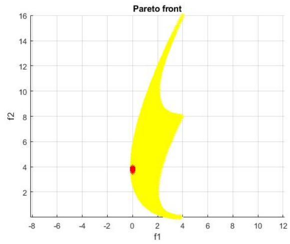
(c)Newton algorithm [18] (d)Quasi-Newton algorithm [44]

Figure 2. **Example 1: Comparison of the existing algorithms for the JOS2a test problem**

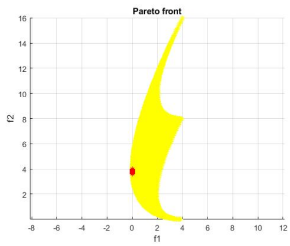
Example 2. Consider a bi-objective problem from [29] with a non-convex Pareto front,



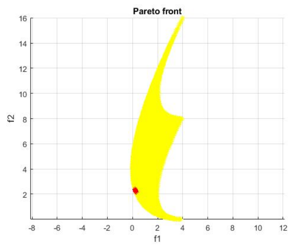
(a)Algorithm 1



(b)BBDMO [8]

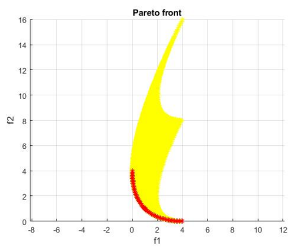


(c)Newton algorithm [18]

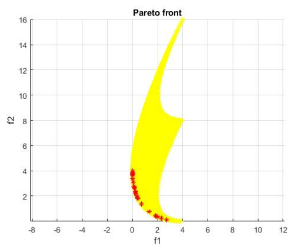


(d)Quasi-Newton algorithm [44]

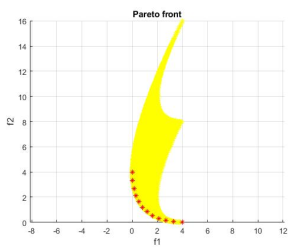
Figure 3. **Example 1: Comparison of the existing algorithms for the JOS1b test problem**



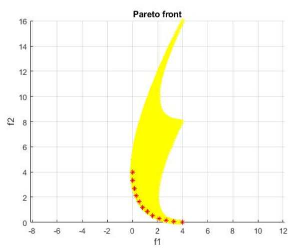
(a)Algorithm 1



(b)BBDMO [8]

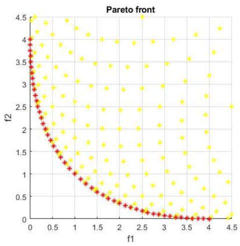


(c)Newton algorithm [18]

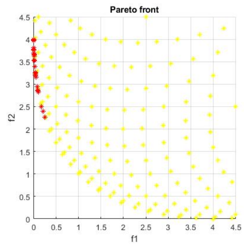


(d)Quasi-Newton algorithm [44]

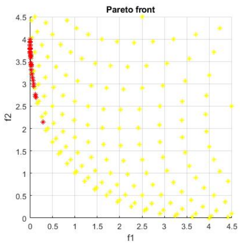
Figure 4. **Example 1: Comparison of the existing algorithms for the JOS2b test problem**



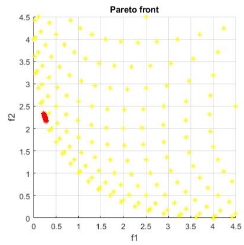
(a)Algorithm 1



(b)BBDMO [8]

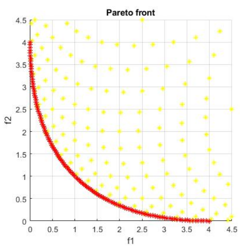


(c)Newton algorithm [18]

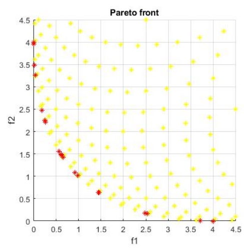


(d)Quasi-Newton algorithm [44]

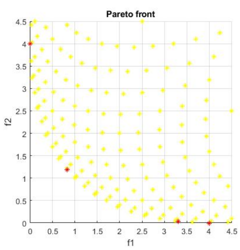
Figure 5. Example 1: Comparison of the existing algorithms for the JOS1c test problem



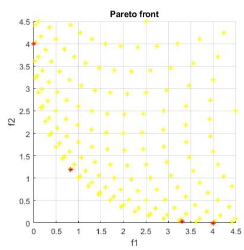
(a)Algorithm 1



(b)BBDMO [8]

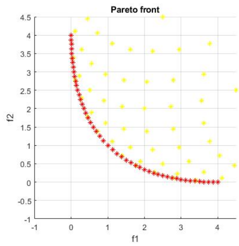


(c)Newton algorithm [18]

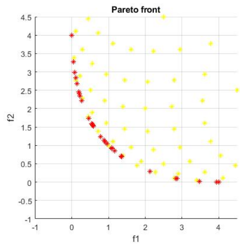


(d)Quasi-Newton algorithm [44]

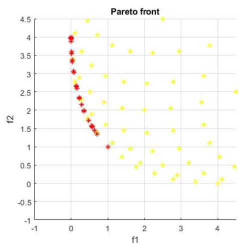
Figure 6. Example 1: Comparison of the existing algorithms for the JOS2c test problem



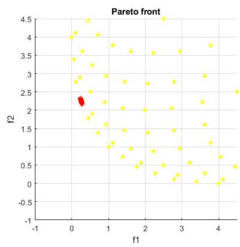
(a)Algorithm 1



(b)BBDMO [8]

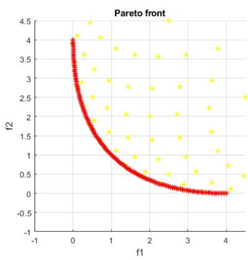


(c)Newton algorithm [18]

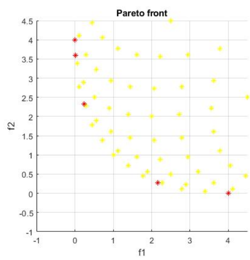


(d)Quasi-Newton algorithm [44]

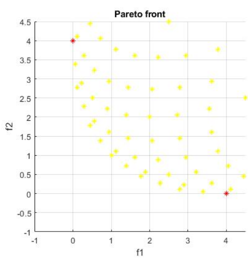
Figure 7. **Example 1: Comparison of the existing algorithms for the JOS1d test problem**



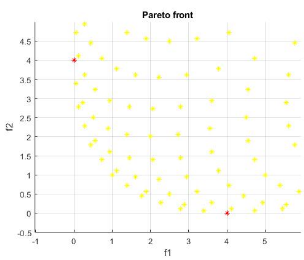
(a)Algorithm 1



(b)BBDMO [8]

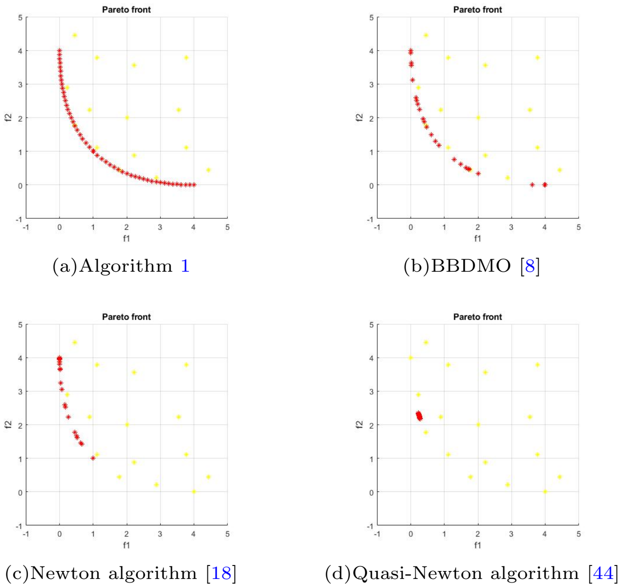


(c)Newton algorithm [18]



(d)Quasi-Newton algorithm [44]

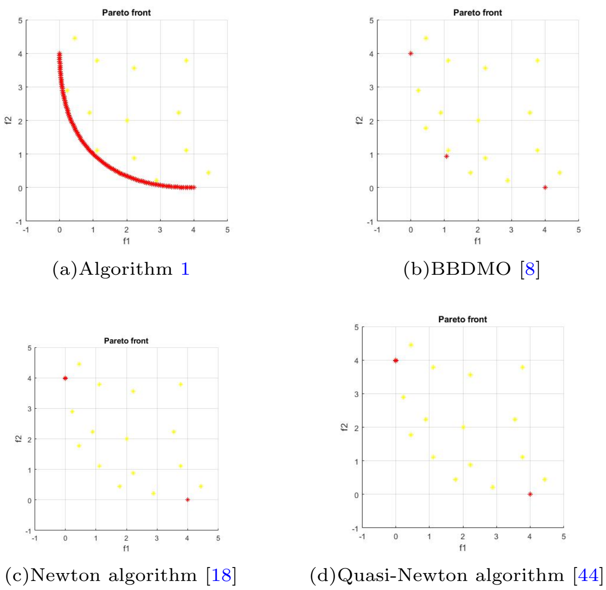
Figure 8. **Example 1: Comparison of the existing algorithms for the JOS2d test problem**



(a)Algorithm 1 (b)BBDMO [8]

(c)Newton algorithm [18] (d)Quasi-Newton algorithm [44]

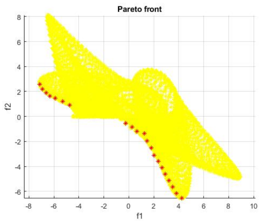
Figure 9. Example 1: Comparison of the existing algorithms for the JOS1e test problem



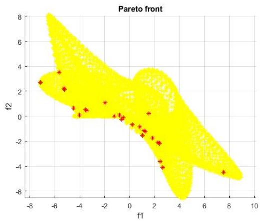
(a)Algorithm 1 (b)BBDMO [8]

(c)Newton algorithm [18] (d)Quasi-Newton algorithm [44]

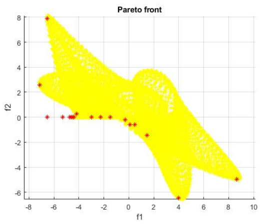
Figure 10. Example 1: Comparison of the existing algorithms for the JOS2e test problem



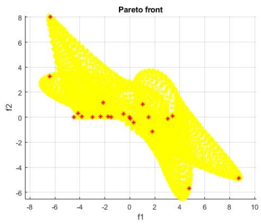
(a)Algorithm 1



(b)BBDMO [8]



(c)Newton algorithm [18]



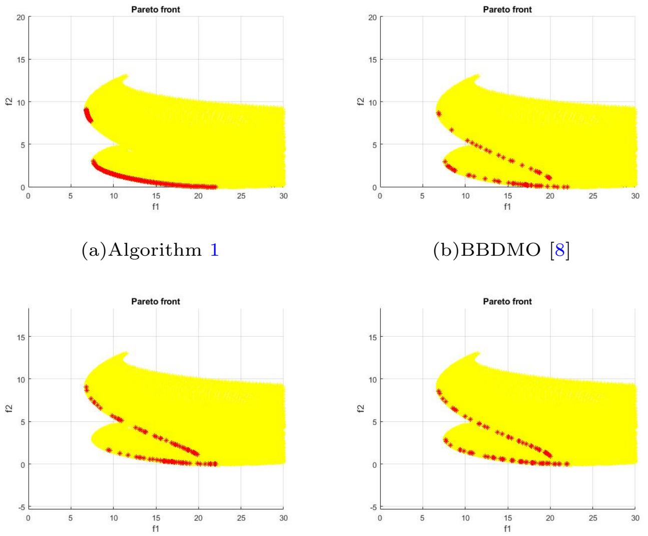
(d)Quasi-Newton algorithm [44]

Figure 11. **Example 2: Comparison of the existing algorithms**

which is expressed as follows:

$$\begin{aligned} \mathbf{KW2} : \min & \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} \\ s.t. \; -3 \le x_i \le 3, \; & i = 1,2 \\ f_1(x) = 3(1-x_1)^2 e^{-x_1^2-(x_2+1)^2} - 10(\frac{x_1}{5} - x_1^3 - x_2^5) e^{-x_1^2-x_2^2} & \\ -3e^{-(x_1+2)^2-x_2^2} + 0.5(2x_1+x_2), & \\ f_2(x) = 3(1-x_2)^2 e^{-x_2^2-(-x_1+1)^2} - 10(-\frac{x_2}{5} + x_2^3 - x_1^5) e^{-x_1^2-x_2^2} & \\ -3e^{-(-x_2+2)^2-x_1^2}. & \end{aligned} \tag{4.2}$$

In this example, we examine a bi-objective problem that has a non-convex Pareto front using Algorithm 1, BBDMO in [8] with max-type nonmonotone line search, Newton algorithm in [18], and quasi-Newton algorithm for solving BFGS method in [44].
In Figure 11, the Pareto front approximated by the algorithms mentioned above is shown. By comparing Part (a) of Figure 11 with other parts of this figure, it can be seen that Algorithm 1 has approximated the Pareto frontier very well, while other algorithms have not obtained good results. Using criteria PM, EV, and EX, we compare these algorithms with $k = 5$ for Algorithm 1, and $N = 2^5 - 1$ stating points with uniform distribution for other algorithms and show the results in Table 3. According to the data of this table, it can be concluded that Algorithm 1 performs better than the other ones.



(a)Algorithm 1 (b)BBDMO [8]

(c)Newton algorithm [18] (d)Quasi-Newton algorithm [44]

Figure 12. **Example 3: Comparison of the existing algorithms**

Table 3. **Numerical results of Example 2**

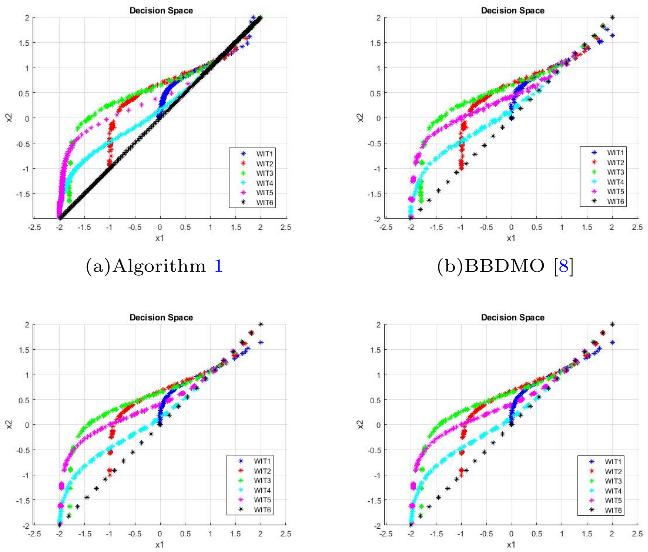
Problem	Algorithm	CPU time (s)	EX	EV	PM
KW2	Algorithm 1	0.34	0.4	0	1
	Newton	0.1	0.9	0.2	0.563
	Quasi-Newton	2.3	0.6	0.72	0.563
	BBDMO	0.21	1.45	1.23	0.32

Example 3. In this example, we examine a problem with a disconnected Pareto front that consists of two objective functions, in which f_1 is a polynomial of degree four and f_2 is the equation of a circle with center $(0, 1)$.

$$\begin{aligned} \textbf{PNR} : \min \ & \left(x_1^4 + x_2^4 - x_1^2 + x_2^2 - 10x_1x_2 + 0.25x_1 + 20, \ x_1^2 + (x_2 - 1)^2\right) \\ s.t. \quad & -2 \leqslant x_1, x_2 \leqslant 2. \end{aligned} \tag{4.3}$$

In this problem, we set the number of repetitions for Algorithm 1 equal to $k = 7$. Also, we consider the number of starting points with uniform distribution for Newton algorithm, quasi-Newton algorithm using BFGS method and BBDMO with max-type nonmonotone line search to be equal to $2^7 - 1$.

As seen in Figure 12, the Pareto frontier is discrete. According to Parts (a)-(d) of Figure 12, the Pareto frontier is completely covered by Algorithm 1, but it is not so in other algorithms. As demonstrated in Table 4, Algorithm 1 works well in this example, because EV and EX indices have lower values than other methods.



(a)Algorithm 1 (b)BBDMO [8]

(c)Newton algorithm [18] (d)Quasi-Newton algorithm [44]

Figure 13. **Example 4: Comparison of the existing algorithms**

Table 4. **Numerical results of Example 3**

Problem	Algorithm	CPU time (s)	EX	EV	PM
PNR	Algorithm 1	2.1	0.17	1.9342e-04	1
	Newton	0.02	1.14	6.8078e-04	0.5
	Quasi-Newton	0.01	1	0.243	0.46
	BBDMO	0.01	1	0.0643	0.8

Example 4. [51] This test problem can be stated as follows:

$$\mathbf{WIT} : \min \begin{bmatrix} \lambda((x_1-2)^2+(x_2-2)^2)+(1-\lambda)((x_1-2)^4+(x_2-2)^8) \\ (x_1+2\lambda)^2+(x_2+2\lambda)^2 \end{bmatrix}, \tag{4.4}$$

where $\lambda = 0, 0.5, 0.9, 0.99, 0.999, 1$ represents WIT1-6, respectively.
In this problem, we put $k = 7$ for Algorithm 1, and $N = 2^7 - 1$ starting points with uniform distribution for other algorithms. In Figure 13, we display the nondominated points obtained from Algorithm 1, Newton method, quasi-Newton algorithm using BFGS method, and BBDMO with max-type nonmonotone line search in the decision space. Furthermore, we have presented the results of comparing the algorithms in Table 5. According to the results given in Figure 13 and Table 5, it can be concluded that Algorithm 1 performs better than other algorithms.

Table 5. **Numerical results of Example 4**

Problem	Algorithm	CPU time (s)	EX	EV	PM
WIT1	Algorithm 1	0.5	0.6	1.3767e-06	1
	Newton	0.9	2.5	1.03	0.62
	Quasi-Newton	0.9	2.5	0.98	0.62
	BBDMO	0.75	4	0.0023	0.69
WIT2	Algorithm 1	0.6	0.9	4.88189e-05	1
	Newton	0.93	2.3	12.5	0.132
	Quasi-Newton	0.93	2.3	12.45	0.16
	BBDMO	1.8	3.1	0.043	1
WIT3	Algorithm 1	0.44	0.02	0.03	1
	Newton	0.85	1.8	16.7	0.24
	Quasi-Newton	0.84	1.8	16.6	0.25
	BBDMO	0.82	3.6	4.53	1
WIT4	Algorithm 1	0.44	0.14	1.5748e-05	1
	Newton	0.7	2.4	0.25	0.3
	Quasi-Newton	0.7	2.4	0.25	0.313
	BBDMO	0.7	1.5	0.004	1
WIT5	Algorithm 1	0.4	0.07	0.00063	1
	Newton	0.7	2.3	2.6	0.3333
	Quasi-Newton	0.7	2.3	2.56	0.3401
	BBDMO	0.73	5.7	0.011	1
WIT6	Algorithm 1	0.4	0.44	1.63236e-06	1
	Newton	0.5	2.4	0.09	0.72
	Quasi-Newton	0.5	2.4	0.09	0.73
	BBDMO	0.5	2.3	0.0012	0.65

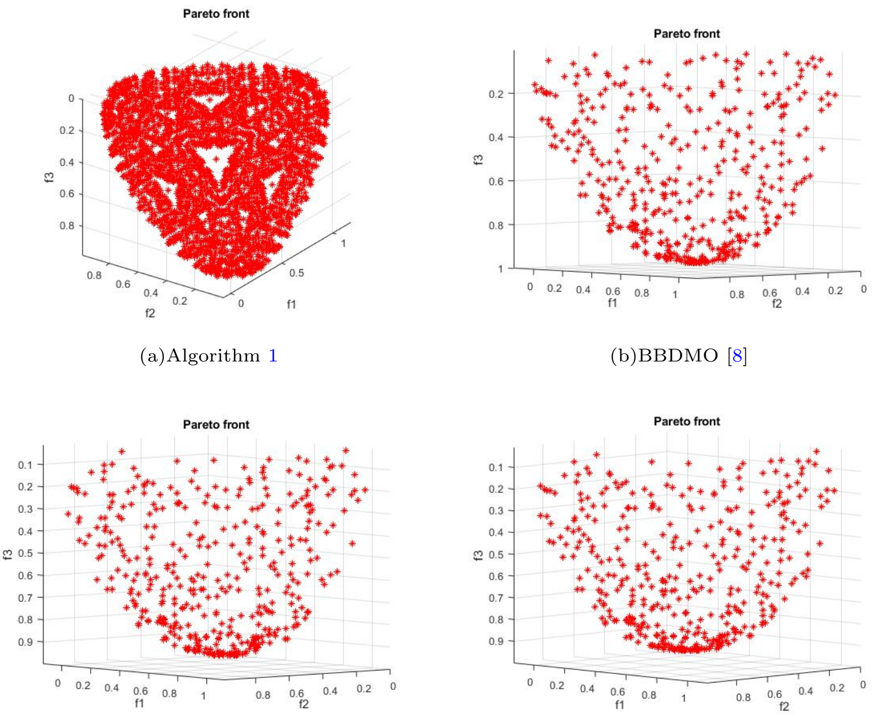
Example 5. This test problem has been studied in [13, 47] and is formulated as follows:

$$
\begin{aligned}
&\textbf{DTLZ2}: \min\ \ (f_1(x),\ f_2(x),\ f_3(x))\\
&\text{where } f_1(x) = (1+(x_3-0.5)^2)\cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2}),\\
&f_2(x) = (1+(x_3-0.5)^2)\cos(\frac{x_1\pi}{2})\sin(\frac{x_2\pi}{2}),\\
&f_3(x) = (1+(x_3-0.5)^2)\sin(\frac{x_1\pi}{2}),\\
&0 \le x_i \le 1,\ \ i = 1,2,3.
\end{aligned}
\tag{4.5}
$$

Here, we suppose that the number of repetitions equals $N = 8$ for Algorithms 1 and set the number of iterations to implement existing algorithm 400. In Figure 14, we display the non-dominated points obtained from Algorithms 1, Newton, quasi-Newton using BFGS method, and BBDMO with max-type nonmonotone line search in the decision space. Furthermore, we have presented the results of comparing the algorithms in Tables 6.

Table 6. **Numerical results of Example 5**

Problem	Algorithm	CPU time (s)	EX	EV	PM
DTLZ2	Algorithm 1	0.31	0.000304	1.82927e-05	1
	Newton	0.03	0.0025	0.0002	0.965
	Quasi-Newton	1.23	0.0025	0.00016	0.96
	BBDMO	1.11	0.0024	0.00015	0.9729



(a)Algorithm 1 (b)BBDMO [8]

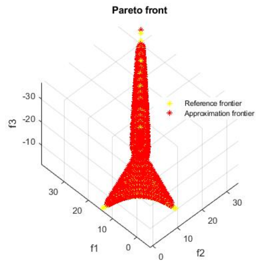
(c)Newton algorithm [18] (d)Quasi-Newton algorithm [44]

Figure 14. **Example 5: Comparison of the existing algorithms**

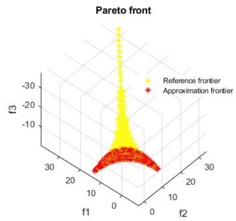
Example 6. This problem is called the comet problem. This implies that the Pareto front's shape begins from a broadly dispersed area and progressively narrows down to a thinner region [13]. This problem has a non-convex Pareto front and is formulated as follows:

$$\begin{aligned} \textbf{COMET} &: \min\ (f_1(x),\ f_2(x),\ f_3(x)) \\ \text{where } f_1(x) &= (1+x_3)(x_1^3x_2^2 - 10x_1 - 4x_2), \\ f_2(x) &= (1+x_3)(x_1^3x_2^2 - 10x_1 + 4x_2), \\ f_3(x) &= 3(1+x_3)x_1^2, \\ 1 &\le x_1 \le 3.5, \\ -2 &\le x_2 \le 2, \\ 0 &\le x_3 \le 1. \end{aligned} \tag{4.6}$$

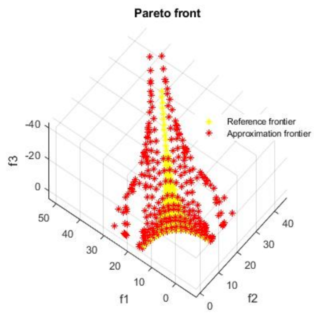
Here, we run Algorithm 1 with $k = 8$, and run the other algorithms with $N = 300$ starting points with uniform distribution. In Figure 15, we display the nondominated points obtained from algorithms 1, Newton, quasi-Newton using BFGS method, and BBDMO with max-type nonmonotone line search in the decision space. Furthermore, we have presented the results of comparing the algorithms in Table 7. According to Figure 15, Algorithm 1 approximates the Pareto frontier very well compared with the other algorithms.



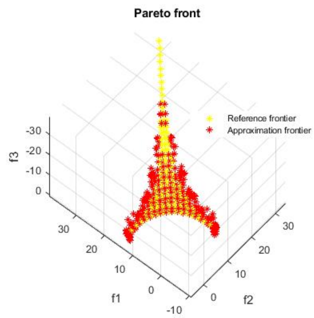
(a)Algorithm 1



(b)BBDMO [8]



(c)Newton algorithm [18]



(d)Quasi-Newton algorithm [44]

Figure 15. **Example 6: Comparison of the existing algorithms**

Table 7. **Numerical results of Example 6**

Problem	Algorithm	CPU time (s)	EX	EV	PM
COMET	Algorithm 1	0.28	0.0003	0.0015	1
	Newton	0.22	0.003	0.02	0.49
	Quasi-Newton	0.5	0.0033	0.016	0.737
	BBDMO	0.2	0.0033	0.017	1

The results of CPU time (s), EV, EX and PM of Algorithm 1, quasi-Newton algorithm using the BFGS method, Newton algorithm and BBDMO with max-type nonmonotone line search for each test problem mentioned in this section of the paper are listed in Tables 2-7. The numerical results obtained from the considered criteria for comparing the algorithms prove that Algorithm 1 outperforms BBDMO, Newton algorithm and quasi-Newton algorithm. As seen in Tables 2-7, the purity metric of Algorithm 1 in all of the examples is equal to 1, and this indicates that the mentioned algorithm includes a higher percentage of the nondominated boundary of each test problem. The results of the indices EV and EX of the mentioned algorithms show that the approximation of the nondominated boundary for each test problem with Algorithm 1 follows an almost uniform distribution. As it can be seen in the fourth row of Table 2, BBDMO with max-type nonmonotone line search does not work well in this example and we are not able to obtain the indices EX and EV. Especially, according to the results obtained for problem $KW2$ with the help of Algorithm 1, it is concluded that this algorithm performs better than other ones.

5. Conclusions

In this paper, we proposed a new algorithm to obtain a uniform approximation of convex or nonconvex Pareto frontier. In this algorithm, by applying the CHINs we provided a new method for generating the initial points for the quasi-Newton method suing BFGS formula. Then, by comparing the algorithms in [8, 18, 44] and the suggested algorithm, we concluded that the Pareto frontiers produced by the new algorithm are better constructed than those obtained by other ones. To achieve this comparison, we used several well-known test problems. In addition, to show the performance efficiency of the proposed algorithm, we used three commonly used indicators. The results of Tables 2-7 show that the algorithms are very encouraging. In view of the performance of Algorithm 1 on various test problems, we can mention the advantage of the new algorithm as follows:

(i) Generating initial points close to the Pareto frontier by applying CHIMs.

(ii) Exponential growth of Pareto points generated in a small number of iterations of the new process.

(iii) Approximation of the Pareto frontier of MOPs with uniform distribution.

(iv) Covering all parts of the Pareto frontier.

However, the suggested algorithm can be implemented for other gradient-based methods such as the Newton method and the quasi-Newton methods, including self-scaling BFGS (SS-BFGS), and the Huang BFGS (H-BFGS).

Acknowledgements: The authors express their sincere gratitude to the referees for the time and effort they dedicated to reviewing this work, as well as for their valuable comments and suggestions. Their constructive feedback has greatly contributed to improving the quality and clarity of this manuscript.

Statements and Declarations

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript. The authors have no relevant financial or non-financial interests to disclose. All authors contributed to the study conception and design and commented on previous versions of the manuscript. All authors read and approved the final manuscript. This research did not generate or analyze any datasets.

References

[1] F. Akbari, M. Ghaznavi, and E. Khorram, *A revised pascoletti–serafini scalarization method for multiobjective optimization problems*, J. Optim. Theory Appl. **178** (2018), no. 2, 560–590.
https://doi.org/10.1007/s10957-018-1289-2.

[2] F. Akbari, E. Khorram, and M. Ghaznavi, *A comparative study of different algorithms and scalarization techniques for constructing the pareto front of multiobjective optimization problems*, Optimization (2024), 1–38.
https://doi.org/10.1080/02331934.2024.2369605.

[3] M.A.T. Ansary, *A Newton-type proximal gradient method for nonlinear multiobjective optimization problems*, Optim. Methods Soft. **38** (2023), no. 3, 570–590.
https://doi.org/10.1080/10556788.2022.2157000.

[4] S. Bandyopadhyay, S.K. Pal, and B. Aruna, *Multiobjective GAs, quantitative indices, and pattern classification*, IEEE Trans. Syst. Man. Cybern., Part B **34** (2004), no. 5, 2088–2099.
https://doi.org/10.1109/TSMCB.2004.834438.

[5] S.P. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[6] Y. Cao, D. Acevedo, Z.K. Nagy, and C.D. Laird, *Real-time feasible multi-objective optimization based nonlinear model predictive control of particle size and shape in a batch crystallization process*, Control Eng. Pract. **69** (2017), 1–8.
https://doi.org/10.1016/j.conengprac.2017.08.008.

[7] G.A. Carrizo, P.A. Lotito, and M.C. Maciel, *Trust region globalization strategy for the nonconvex unconstrained multiobjective optimization problem*, Math. Program. **159** (2016), no. 1, 339–369.
https://doi.org/10.1007/s10107-015-0962-6.

[8] J. Chen, L. Tang, and X. Yang, *A Barzilai-Borwein descent method for multiobjective optimization problems*, Eur. J. Oper. Res. **311** (2023), no. 1, 196–209.
https://doi.org/10.1016/j.ejor.2023.04.022.

[9] G. Cocchi and M. Lapucci, *An augmented Lagrangian algorithm for multiobjective optimization*, Comput. Optim. Appl. **77** (2020), no. 1, 29–56.
https://doi.org/10.1007/s10589-020-00204-z.

[10] C.A. Coello Coello, G.B. Lamont, and D.A. VanVeldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, New York, 2007.

[11] I. Das and J.E. Dennis, *Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems*, SIAM J. Optim. **8** (1998), no. 3, 631–657.
https://doi.org/10.1137/S1052623496307510.

[12] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, Chichester, 2001.

[13] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable test problems for evo-*

lutionary multiobjective optimization, Tech. report, Kanpur Genetic Algorithms Lab. (KanGAL), Indian Inst. Technol., Kanpur, India, 2001.

[14] P. Dhal and C. Azad, *A multi-objective feature selection method using Newton's law based PSO with GWO*, Appl. Soft Comput. **107** (2021), 107394. https://doi.org/10.1016/j.asoc.2021.107394.

[15] A. Domahidi, E. Chu, and S. Boyd, *ECOS: An SOCP solver for embedded systems*, 2013 European Control Conference (ECC), IEEE, 2013, pp. 3071–3076. https://doi.org/10.23919/ECC.2013.6669541.

[16] L.M.G. Drummond and A.N. Iusem, *A projected gradient method for vector optimization problems*, Comput. Optim. Appl. **28** (2004), no. 1, 5–29. https://doi.org/10.1023/B:COAP.0000018877.86161.8b.

[17] M. Ehrgott, *Multicriteria Optimization*, Springer, Berlin, 2005.

[18] J. Fliege, L.M.G. Drummond, and B.F. Svaiter, *Newton's method for multiobjective optimization*, SIAM J. Optim. **20** (2009), no. 2, 602–626. https://doi.org/10.1137/08071692X.

[19] J. Fliege and B.F. Svaiter, *Steepest descent methods for multicriteria optimization*, Math. Oper. Res. **51** (2000), no. 3, 479–494. https://doi.org/10.1007/s001860000043.

[20] J. Fliege and A.I.F. Vaz, *A method for constrained multiobjective optimization based on SQP techniques*, SIAM J. Optim. **26** (2016), no. 4, 2091–2119. https://doi.org/10.1137/15M1016424.

[21] J. Fliege and R. Werner, *Robust multiobjective optimization & applications in portfolio optimization*, Eur. J. Oper. Res. **234** (2014), no. 2, 422–433. https://doi.org/10.1016/j.ejor.2013.10.028.

[22] N. Ghalavand, E. Khorram, and V. Morovati, *An adaptive nonmonotone line search for multiobjective optimization problems*, Comput. Oper. Res. **136** (2021), 105506.
https://doi.org/10.1016/j.cor.2021.105506.

[23] ______, *Two adaptive nonmonotone trust-region algorithms for solving multiobjective optimization problems*, Optimization **73** (2024), no. 9, 2953–2985. https://doi.org/10.1080/02331934.2023.2234920.

[24] A. Ghane-Kanafi and E. Khorram, *A new scalarization method for finding the efficient frontier in non-convex multi-objective problems*, Appl. Math. Model. **39** (2015), no. 23-24, 7483–7498.
https://doi.org/10.1016/j.apm.2015.03.022.

[25] M. Ghaznavi, F. Akbari, and E. Khorram, *Optimality conditions via a unified direction approach for (approximate) efficiency in multiobjective optimization*, Optim. Methods Softw. **36** (2021), no. 2-3, 627–652. https://doi.org/10.1080/10556788.2019.1571589.

[26] M. Ghaznavi and Z. Azizi, *An algorithm for approximating nondominated points of convex multiobjective optimization problems*, Bull. Iran. Math. Soc. **43** (2017), no. 5, 1399–1415.

[27] Y. Jin, M. Olhofer, and B. Sendhoff, *Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?*, Proceedings of

the Genetic and Evolutionary Computation Conference, 2001, pp. 1042–1049.

[28] V.D. Joshi, M. Sharma, and J. Singh, *Solving multi-choice solid stochastic multi objective transportation problem with supply, demand and conveyance capacity involving newton divided difference interpolations*, J. Comput. Anal. Appl. **33** (2024), no. 1, 372–395.

[29] I.Y. Kim and O.L. De Weck, *Adaptive weighted-sum method for bi-objective optimization: Pareto front generation*, Struct. Multidiscip. Optim. **29** (2005), no. 2, 149–158.
https://doi.org/10.1007/s00158-004-0465-1.

[30] K. Kumar, D. Ghosh, A. Upadhayay, J.C. Yao, and X. Zhao, *Quasi-newton methods for multiobjective optimization problems: A systematic review*, Appl. Set-Valued Anal. Optim. **5** (2023), no. 2, 291–321.
https://doi.org/10.23952/asvao.5.2023.2.12.

[31] M. Lapucci and P. Mansueto, *Improved front steepest descent for multi-objective optimization*, Oper. Res. Lett. **51** (2023), no. 3, 242–247.
https://doi.org/10.1016/j.orl.2023.03.001.

[32] ______, *A limited memory Quasi-Newton approach for multi-objective optimization*, Comput. Optim. Appl. **85** (2023), no. 1, 33–73.
https://doi.org/10.1007/s10589-023-00454-7.

[33] L.R. Lucambio Pérez and L.F. Prudente, *Nonlinear conjugate gradient methods for vector optimization*, SIAM J. Optim. **28** (2018), no. 3, 2690–2720.
https://doi.org/10.1137/17M1126588.

[34] H. Meng, X. Zhang, and S. Liu, *New quality measures for multiobjective programming*, Advances in Natural Computation (Berlin, Heidelberg) (L. Wang, K. Chen, and Y.S. Ong, eds.), Springer Berlin Heidelberg, 2005, pp. 1044–1048.

[35] Q. Mercier, F. Poirion, and J.A. Désidéri, *A stochastic multiple gradient descent algorithm*, Eur. J. Oper. Res. **271** (2018), no. 3, 808–817.
https://doi.org/10.1016/j.ejor.2018.05.064.

[36] A. Messac and C.A. Mattson, *Normal constraint method with guarantee of even representation of complete pareto frontier*, AIAA J. **42** (2004), no. 10, 2101–2111.
https://doi.org/10.2514/1.8977.

[37] K. Miettinen, *Nonlinear Multi-objective Optimization*, Kluwer Academic, Dordrecht, 1999.

[38] V. Morovati, H. Basirzadeh, and L. Pourkarimi, *Quasi-Newton methods for multiobjective optimization problems*, 4OR-Q J. Oper. Res. **16** (2018), no. 3, 261–294.
https://doi.org/10.1007/s10288-017-0363-1.

[39] V. Morovati and L. Pourkarimi, *Extension of Zoutendijk method for solving constrained multiobjective optimization problems*, Eur. J. Oper. Res. **273** (2019), no. 1, 44–57.
https://doi.org/10.1016/j.ejor.2018.08.018.

[40] V. Morovati, L. Pourkarimi, and H. Basirzadeh, *Barzilai and Borwein's method for multiobjective optimization problems*, Numer. Algorithms **72** (2016), no. 3, 539–604.
https://doi.org/10.1007/s11075-015-0058-7.

[41] H. Mukai, *Algorithms for multicriterion optimization*, IEEE Trans. Autom. **25** (1980), no. 2, 177–186.
https://doi.org/10.1109/TAC.1980.1102298.

[42] J. Nocedal and S.J. Wright, *Numerical Optimization*, 2nd ed., Springer Science+Business Media, LLC, New York, 2006.

[43] A. Pascoletti and P. Serafini, *Scalarizing vector optimization problems*, J. Optim. Theory Appl. **42** (1984), no. 4, 499–524.
https://doi.org/10.1007/BF00934564.

[44] Ž. Povalej, *Quasi-Newton's method for multiobjective optimization*, J. Comput. Appl. Math. **255** (2014), 765–777.
https://doi.org/10.1016/j.cam.2013.06.045.

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

[46] S. Qu, M. Goh, and F.T.S. Chan, *Quasi-Newton methods for solving multiobjective optimization*, Oper. Res. Lett. **39** (2011), no. 5, 397–399.
https://doi.org/10.1016/j.orl.2011.07.008.

[47] M.M. Rizvi, *New optimality conditions for non-linear multiobjective optimization problems and new scalarization techniques for constructing pathological pareto fronts*, Ph.D. thesis, University of South Australia, 2013.

[48] S. Schäffler, R. Schultz, and K. Weinzierl, *Stochastic method for the solution of unconstrained vector optimization problems*, J. Optim. Theory Appl. **114** (2002), no. 1, 209–222.
https://doi.org/10.1023/A:1015472306888.

[49] W. Sun and Y.X. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer, New York, 2006.

[50] D.A.G. Vieira, R.H.C. Takahashi, and R.R. Saldanha, *Multicriteria optimization with a multiobjective golden section line search*, Math. Program. **131** (2012), no. 1, 131–161.
https://doi.org/10.1007/s10107-010-0347-9.

[51] K. Witting, *Numerical algorithms for the treatment of parametric multiobjective optimization problem and applications*, Ph.D. thesis, Universität Paderborn, Paderborn, 2012.

[52] S. Yang, J. Wang, M. Li, and H. Yue, *Research on intellectualized location of coal gangue logistics nodes based on particle swarm optimization and quasi-newton algorithm*, Mathematics **10** (2022), no. 1, 162.
https://doi.org/10.3390/math10010162.

[53] F. YE, B. Lin, Z. Yue, P. Guo, Q. Xiao, and Y. Zhang, *Multi-objective meta learning*, Advances in Neural Information Processing Systems (M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J.W. Vaughan, eds.), vol. 34, Curran Associates, Inc., 2021, pp. 21338–21351.

[54] E.A. Yildirim and S.J. Wright, *Warm-start strategies in interior-point methods*

for linear programming, SIAM J. Optim. **12** (2002), no. 3, 782–810.
https://doi.org/10.1137/S1052623400369235.

[55] Y. Zhu, H. Wu, Z. Zhang, C. Zong, and D. Xu, *Optimal power flow research of AC–DC hybrid grid with multiple energy routers*, Electr. Power Syst. Res. **228** (2024), 110090.
https://doi.org/10.1016/j.epsr.2023.110090.