4OR-Q J Oper Res (2018) 16:261–294
https://doi.org/10.1007/s10288-017-0363-1

RESEARCH PAPER

Quasi-Newton methods for multiobjective optimization problems

Vahid Morovati[1] · **Hadi Basirzadeh**[1] · **Latif Pourkarimi**[2]

Received: 16 August 2016 / Revised: 9 September 2017 / Published online: 14 November 2017

Abstract This work is an attempt to develop multiobjective versions of some well-known single objective quasi-Newton methods, including BFGS, self-scaling BFGS (SS-BFGS), and the Huang BFGS (H-BFGS). A comprehensive and comparative study of these methods is presented in this paper. The Armijo line search is used for the implementation of these methods. The numerical results show that the Armijo rule does not work the same way for the multiobjective case as for the single objective case, because, in this case, it imposes a large computational effort and significantly decreases the speed of convergence in contrast to the single objective case. Hence, we consider two cases of all multi-objective versions of quasi-Newton methods: in the presence of the Armijo line search and in the absence of any line search. Moreover, the convergence of these methods without using any line search under some mild conditions is shown. Also, by introducing a multiobjective subproblem for finding the quasi-Newton multiobjective search direction, a simple representation of the Karush–Kuhn–Tucker conditions is derived. The H-BFGS quasi-Newton multiobjective optimization method provides a higher-order accuracy in approximating the second order curvature of the problem functions than the BFGS and SS-BFGS methods. Thus, this method has some benefits compared to the other methods as shown in the numerical results. All mentioned methods proposed in this paper are evaluated and compared with each other in different aspects. To do so, some well-known test problems and performance assessment criteria are employed. Moreover, these methods are compared with each other

✉ Vahid Morovati
vahidmath89@gmail.com

[1] Department of Mathematics, Faculty of Mathematical Sciences and Computer, Shahid Chamran University of Ahvaz, Ahvaz, Iran

[2] Department of Mathematics, Razi University, Kermanshah, Iran

with regard to the expended CPU time, the number of iterations, and the number of function evaluations.

Keywords Quasi-Newton methods · Multiobjective optimization · Nonparametric methods · Nondominated points · Performance profiles

Mathematics Subject Classification 90C29 · 90C30 · 90C53

1 Introduction

As an important area of optimization, multiobjective optimization is dealing with problems in which some objective functions are required to be simultaneously optimized. It has attracted attention in many applied sciences in the presence of trade-off analysis, such as: medicine, economics, engineering, transportation, and logistics optimization. In these fields, the multiobjective optimization is the best strategy to make a desired decision. For example, the problem of buying a car can be considered as a multiobjective optimization problem (MOP) where minimizing cost and maximizing comfort are two conflicting objectives. In practical optimization problems, the number of objective functions is larger than one objective. Due to the inherent conflict among objective functions, there exists no solution that could simultaneously optimize all objectives. There are a large number of solutions for these problems called efficient (Pareto optimal) solutions. An efficient or Pareto optimal solution is a solution in which the improvement of any objective function leads to a decline in at least one objective function. The value of the objective functions at an efficient solution is called a nondominated point. The set of all nondominated points and the set of all efficient solutions are called the nondominated and efficient sets, respectively. As shown in proposition 2.4 in Ehrgott (2005), each nondominated point belongs to the boundary of the feasible set in the objective space. Thus, the set of all nondominated points is usually called nondominated frontier. In contrast to single objective optimization, there are many concepts of solution in MOP. Researchers can find a representation of efficient solutions (nondominated points) or employ a trade-off analysis to find the most preferred solution in the decision maker point of view (Sayadi-Bander et al. 2015; Sayadi-bander et al. 2017a, b; Basirzadeh et al. 2014; Eskelinen and Miettinen 2012; Sakawa and Yano 1990; Kuk et al. 1997).

One of the interesting fields in multiobjective optimization is the approximation of the entire nondominated frontier. Computing an approximation of the whole nondominated frontier enables decision makers to have a deep insight into the problem structure and information about trade-off. A trade-off denotes the amount of giving up on one of the objective functions, which leads to the improvement of other objective functions. In multiobjective optimization problems, having a good approximation of the entire nondominated frontier enables us to employ some secondary preferences over the nondominated frontier so as to choose the most preferred solution. Motivated by this important application, nowadays there is an increasing interest in the methods of approximating the whole nondominated frontier. This fact is demonstrated by many published papers related to this topic (Benson and Sayin 1997; Das and Dennis 1998;

Fliege 2004, 2006; Basseur 2006; Fliege and Heseler 2002; Segura et al. 2013; Da Silva 2010; Schandl et al. 2001).

One of the most important classes of algorithms for solving an MOP is the class of scalarization methods. Some parameters are usually needed to be determined for scalarizing an MOP, but obtaining a suitable scalarization function with parameters needs a great insight into the problem structure, which is a very difficult task and in some cases almost impossible. The weaknesses of the scalarization method are illustrated explicitly in detail in Sect. 7 of the work of Fliege et al. (2009). Hence, developing some methods which do not need this prior information can be of special interest. In this regard, recently a new class of multiobjective optimization algorithms has been developed which does not depend on any parameter (Fliege and Svaiter 2000; Drummond and Iusem 2004; Drummond and Svaiter 2005; Fliege et al. 2009; Qu et al. 2011; Povalej 2014; Morovati et al. 2016; Qu et al. 2013; Villacorta et al. 2014). Indeed, these methods are generalizations of optimization algorithms for single objective problems.

Fliege et al. (2009) have proposed an a priori parameter-free Newton-type optimization method for obtaining a point that satisfies the first order necessary condition for multiobjective optimization. Similar to the classical Newton method in each iteration, they considered some quadratic approximations related to the original objective functions. They used a quadratic subproblem to find a search direction.

Newton's method is a well-known method for solving single objective optimization problems, because it uses the Hessian which offers useful curvature information. However, for various practical problems, the computing efforts of the Hessian matrices are very expensive or the Hessian is not available analytically.

These difficulties lead to some methods that only use the function values and the gradients of the objective functions in ways closely related to Newton's method. Some of these methods are quasi-Newton methods which do not need to compute the Hessian, but generate a series of Hessian approximations and at the same time maintain a fast rate of convergence (Sun and Yuan 2006). Recently, the BFGS quasi-Newton multiobjective optimization (QNMO) method has been proposed by some authors Qu et al. (2011), Povalej (2014).

As mentioned, the approximation of the Hessian matrix is used in the quasi-Newton methods instead of the Hessian matrix. There are many methods for approximating this matrix. In the QNMO method proposed by Povalej (2014), the BFGS method is used to approximate the Hessian matrix. It is well-known that the BFGS method belongs to the Broyden class (Nocedal and Wright 2006). In the quasi-Newton methods proposed in this paper for the MOP, besides the BFGS method which is the most well-known and important among the methods of the Broyden class, the self-scaling BFGS (SS-BFGS) method from the self-scaling Broyden class and the Huang BFGS (H-BFGS) method from the Huang class are used. We tested the methods on several well-known test problems in two points of view. The first point of view contains CPU times and the computational effort and the second one, which deals with the quality of the nondominated frontiers, contains such criteria as the number of nondominated points, the value of the purity metric, spread metric, and the epsilon indicator. In order to have a good comparison, an approximated nondominated frontier for each test problem, namely a reference frontier, was determined by using the numerical results of all proposed methods. Then, the obtained approximated nondominated frontier of any algorithm was compared with the reference frontier. All of the numerical results are explicitly represented in some appropriate figures. Each of these figures contains some curves called performance profiles which correspond to some methods. These profiles can be used to compare the computational performances of the methods (Dolan and More 2002). Also, the number of iterations and function evaluations corresponding to each of the methods is reported.

Unlike the methods of Broyden and self-scaling Broyden classes, the methods of the Huang class not only use the gradient information, but also employ function values to approximate the Hessian matrix. For this reason, the methods of the Huang class provide a higher-order accuracy in approximating the second-order curvature of the problem functions than the methods of the Broyden and self-scaling Broyden classes (See Theorem 2). The numerical results show that the H-BFGS and the SS-BFGS methods have a better performance than the BFGS method in different points of view.

As given in Fliege et al. (2009) and Povalej (2014), in the QNMO methods a scalar quadratic subproblem is required to find the search direction in each iteration. In this paper, first, a quadratic multiobjective subproblem is introduced. Then, comparing the solution sets of this subproblem and the scalar subproblem, some theoretical results are derived. Particularly, using these theoretical results and the known results from the theory of multiobjective optimization, a simple representation of the Karush–Kuhn–Tucker (KKT) conditions is presented.

In the BFGS QNMO method developed by Qu et al. (2011) and Povalej (2014), similar to the single objective case, an Armijo-like rule is used for line search. In this paper, we use the Armijo-like rule to find a step length in a similar manner. When employing the Armijo-like rule in multiobjective problems, more than one inequality should be simultaneously satisfied. Thus, this rule imposes a large computational effort and yields a small step length. Consequently, using the Armijo-like rule for finding a step length decreases the speed of convergence. Based on this fact, in this paper all methods are considered in two cases: in the presence of the Armijo line search and in the absence of any line search. Similar to Fliege et al. (2009), Povalej proved the convergence of the QNMO methods in the presence of the Armijo line search (Povalej 2014). In this paper, we extend this proof to the case where no line search was used.

This paper is organized as follows. Section 2 contains some notations, definitions, and preliminaries that are used in the remainder of the paper. Section 3 is devoted to the quasi-Newton optimization methods. First, a brief description of the quasi-Newton directions for single objective optimization problems is presented. Then, the quasi-Newton subproblem for the search direction in the MOP is investigated and some propositions are provided. Section 4 proceeds with the convergence analysis. Section 5 deals with the performance assessment. In this section, we review some well-known standard criteria for comparing the quality of the obtained nondominated frontiers. In Sect. 6, the three mentioned quasi-Newton optimization methods for the MOP are implemented for several well-known test problems. Moreover, in this section the obtained numerical results are explicitly evaluated by some standard criteria. Finally, the paper concludes with some discussions on comparing the QNMO methods.

2 Preliminaries

In this section, some notations that are used in the remainder of the paper are introduced. Throughout the paper, $\mathbb{R}$, $\mathbb{R}_{\geq}$, and $\mathbb{R}_{>}$ stand for the sets of real numbers, non-negative real numbers, and strictly positive real numbers, respectively. $V \subset \mathbb{R}^n$ denotes an open set and $F : V \to \mathbb{R}^m$ is an arbitrary vector function. For this vector function, $\mathbb{R}^n$ and $\mathbb{R}^m$ are called the decision and objective spaces, respectively. We assume that $F \in C^2(V, \mathbb{R}^m)$; i.e. $F = (F_1, \ldots, F_m)$ is twice continuously differentiable on V. For $x \in V$, we will use symbols $JF(x) \in \mathbb{R}^{m\times n}$, $\nabla F_j(x) \in \mathbb{R}^m$, $HF_j(x) \in \mathbb{R}^{n\times n}$, and $B^j(x) \in \mathbb{R}^{n\times n}$ to denote the Jacobian matrix of F, the gradient of F_j, the Hessian matrix of F_j, and the approximation of the Hessian matrix of F_j at x, respectively. We denote the range or image space of matrix A by $S(A)$. Also, the notation $B \preccurlyeq C$ ($B \prec C$) means that $C - B$ is a positive semidefinite (definite) matrix.

In this paper, the following unconstrained MOP is considered:

$$\min_{x\in V} \ F(x). \tag{1}$$

V and $Y = F(V)$ are called the feasible set in the decision and objective spaces of Problem (1), respectively. In the following, related to Problem (1), some preliminary definitions, a lemma, and a theorem are presented.

Definition 1 (Ehrgott 2005) A feasible solution $x^* \in V$ is called an (a weakly) efficient solution of Problem (1), if there is no other $x \in V$ such that $F(x) \leq F(x^*)$ and $F(x) \neq F(x^*)$ ($F(x) < F(x^*)$). If x^* is an (a weakly) efficient solution, $F(x^*)$ is called a (weakly) nondominated point.

The set of all nondominated points and efficient solutions are called the nondominated and efficient sets, respectively. As shown in proposition 2.4 in Ehrgott (2005), each nondominated point belongs to the boundary of the feasible set in the objective space (Y). Thus, the set of all nondominated points is usually called nondominated frontier. We denote the nondominated frontier or nondominated set by Y_N.

Numerical algorithms usually find a local solution. Definition 2 deals with the locally (weakly) efficient solutions.

Definition 2 (Fliege et al. 2009) A feasible solution $x^* \in V$ is called a locally (weakly) efficient solution, if there is a neighbourhood $W \subseteq V$ of x^* such that x^* is an (weakly) efficient solution on W. If x^* is a locally (weak) efficient solution, $F(x^*)$ is called a locally (weakly) nondominated point.

Definition 3 (Ehrgott 2005) Let $V \subset \mathbb{R}^n$ be a convex set. $F : V \to \mathbb{R}^m$ is called $\mathbb{R}^m$-convex if F is componentwise-convex.

In convex problems, globally optimal treatment can be derived from a local treatment as given in Lemma 1.

Lemma 1 (Fliege et al. 2009) *Let $V \subset \mathbb{R}^n$ be a convex set and $F : V \to \mathbb{R}^m$ be an $\mathbb{R}^m$-convex function. Then, any locally efficient solution is an efficient solution.*

In the presence of some assumptions and $\mathbb{R}^p_{\geq}$-compactness, the objective functions satisfy some nice properties which are useful for the theoretical and computational aspects. In the following $\mathbb{R}^p_{\geq}$-compactness is defined.

Definition 4 (Ehrgott 2005) A set $Y \subset \mathbb{R}^p$ is called $\mathbb{R}^p_{\geq}$-compact if for all $y \in Y$ the section $(y - \mathbb{R}^p_{\geq}) \cap Y$ is compact.

Definition 5 (Ehrgott 2005) The nondominated set Y_N is said to be externally stable if for each $y \in Y \backslash Y_N$ there is $\hat{y} \in Y_N$ such that $y \in \hat{y} + \mathbb{R}^p_{\geq}$.

A necessary condition for a given point $x \in \mathbb{R}^n$ to be a locally weakly efficient solution for function $F : V \to \mathbb{R}^m$ is $S(JF(x)) \cap (-\mathbb{R}^m_{>}) = \emptyset$ (Gopfert and Nehse 1990; Luc 1988). Motivated by this fact, a critical point is defined as follows.

Definition 6 (Fliege and Svaiter 2000) Point $x \in \mathbb{R}^n$ is said to be a locally critical point of vector function $F : V \to \mathbb{R}^m$ if $S(JF(x)) \cap (-\mathbb{R}^m_{>}) = \emptyset$.

From Definition 6, it is clear that if x is not critical, then there exists $d \in \mathbb{R}^n$ such that $\nabla F_j(x)d < 0$ for all $j \in \{1, \ldots, m\}$.

In Theorem 1, the relation between a critical point and a (weakly) efficient solution is explained (Fliege et al. 2009).

Theorem 1 (Fliege et al. 2009) *Let $F \in C^1(V, \mathbb{R}^m)$,*

1. *If $\bar{x}$ is a locally weakly efficient solution, then $\bar{x}$ is a critical point for F.*
2. *If V is convex, F is $\mathbb{R}^m$-convex and $\bar{x} \in V$ is critical for F, then $\bar{x}$ is a weakly efficient solution.*
3. *If V is convex, $\bar{x} \in V$ is a critical point of F, $F \in C^2(V, \mathbb{R}^m)$ and $HF_j(x) \succ 0$ for all $x \in V$ and $j \in \{1, 2, \ldots, m\}$, then $\bar{x}$ is an efficient solution.*

We refer interested readers to Fliege et al. (2009) for the proof of Theorem 1.

In Sect. 3, first, a brief exposition of the quasi-Newton direction for single objective optimization is presented and then it will be extended for multiobjective optimization (Qu et al. 2011; Povalej 2014). Finally, we will give some algorithms of quasi-Newton methods for multiobjective optimization, both with line search (Qu et al. 2011; Povalej 2014) and without any line search.

3 Quasi-Newton optimization methods

Newton's method is a well-known method that uses the Hessian matrix and considers curvature for solving single objective nonlinear optimization problems. However, the analytical computing of the Hessian matrix is a difficult or impossible task in many problems. In order to overcome this difficulty, the quasi-Newton methods are introduced, which use the gradients of objective functions and function values. In these methods, approximations of the Hessian matrices are considered and a fast rate of convergence is provided at the same time. The quasi-Newton methods generate a sequence of approximated Hessian matrices $\{B_k\}$ and a sequence of related directions

$\{d_k\}$ where $d_k = -B_k^{-1}\nabla f(x_k)$. We hope that $\{B_k\}$ possesses positive definiteness and the algorithm behaves like Newton's method.

In this section, first, a brief description of the quasi-Newton direction for single objective optimization problems is presented and some of the well-known update formulas are introduced. Then, in the subsequent subsection, these update formulas and the quasi-Newton direction are extended to the multiobjective case.

3.1 Quasi-Newton methods for single objective optimization

Let $f : V \subset \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable function on an open set $V \subset \mathbb{R}^n$. Suppose, we want to solve the problem $\min_{x \in V}\ f(x)$ by a quasi-Newton method. The quasi-Newton search direction at a point x_k ($d_k := d(x_k)$) can be obtained by solving the following quadratic subproblem:

$$\min_{d_k \in \mathbb{R}^n}\ \nabla f(x_k)^T d_k + \frac{1}{2} d_k^T B_k d_k, \tag{2}$$

where $B_k := B(x_k)$ is an approximation of the Hessian matrix of f at x_k. An easy computation shows that the unique solution of Subproblem (2) is $d_k = B_k^{-1}\nabla f(x_k)$. Thus, the new iteration derives from $x_{k+1} = x_k + t_k d_k$ with the step length $t_k \in \mathbb{R}_{\geq}$.

Two important classes for the approximation of the Hessian matrices are the Broyden class and the self-scaling Broyden class. These two classes just use gradient information. All approximate matrices obtained from these classes are symmetric and satisfy the following equation, namely, the secant equation or the quasi-Newton equation:

$$B_{k+1} s_k = y_k, \tag{3}$$

where $s_k := x_{k+1} - x_k = t_k d_k$, $y_k := \nabla f(x_{k+1}) - \nabla f(x_k)$, $B_{k+1} := B(x_{k+1})$, and $\{x_k\}$ is supposed to be the sequence generated by a quasi-Newton method. Indeed, the secant equation is derived from the gradient information of the objective function at points x_k and x_{k+1}. In what follows, the updating formulas related to the BFGS method of the Broyden class and the self-scaling BFGS (SS-BFGS) method of the self-scaling Broyden class are explained with their required conditions for preserving the positive definiteness property (Nocedal and Wright 2006; Sun and Yuan 2006). It should be noted that the initial matrix B_0 for all updating formulas presented in this section is assumed to be the identity matrix.

The updating formula for BFGS is (Nocedal and Wright 2006)

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}. \tag{4}$$

The updating formula for SS-BFGS is (Sun and Yuan 2006)

$$B_{k+1} = \frac{s_k^T y_k}{s_k^T B_k s_k}\left(B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}\right) + \frac{y_k y_k^T}{s_k^T y_k}. \tag{5}$$

As mentioned before, the matrix B_{k+1} obtained from the above updating formula satisfies the secant equation. Thus, for preserving the positive definiteness property of B_{k+1}, the following curvature condition is a sufficient condition:

$$s_k^T y_k > 0. \tag{6}$$

Unlike the Broyden and the self-scaling Broyden classes that employ only the gradients and ignore the function value information, the Huang class uses both gradient information and function evaluations for approximating the Hessian matrix. All approximate matrices obtained from the Huang class should satisfy the following equation, namely, the Huang quasi-Newton equation instead of the secant equation (3) (Zhang and Xu 2001; Sun and Yuan 2006):

$$B_{k+1}s_k = \hat{y}_k, \quad \hat{y}_k = y_k + \frac{\theta_k}{s_k^T y_k} y_k, \tag{7}$$

where $\theta_k := 6(f_k - f_{k+1}) + 3(\nabla f_k + \nabla f_{k+1})^T s_k$. In this case, the Huang BFGS (H-BFGS) updating formula is as follows:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{\hat{y}_k \hat{y}_k^T}{s_k^T \hat{y}_k}. \tag{8}$$

Theorem 2 (Zhang et al. 1999) *Assume that $f(.)$ is a sufficiently smooth function. If $\|s_k\|$ is small enough then we have:*

$$\begin{aligned} s_k^T \left[Hf(x_{k+1})s_k - \hat{y}_k\right] &= O(\|s_k\|^4), \\ s_k^T \left[Hf(x_{k+1})s_k - y_k\right] &= O(\|s_k\|^3). \end{aligned} \tag{9}$$

Assume that B_{k+1} and $\hat{B}_{k+1}$ satisfy Eqs. (3) and (7), respectively. In this case, by the above theorem, the following relations hold:

$$\begin{aligned} s_k^T \hat{B}_{k+1} s_k &= s_k^T Hf(x_{k+1})s_k + O(\|s_k\|^4), \\ s_k^T B_{k+1} s_k &= s_k^T Hf(x_{k+1})s_k + O(\|s_k\|^3). \end{aligned} \tag{10}$$

The two relations shown in (10) show that in the presence of the assumptions of Theorem 2, $s_k^T \hat{B}_{k+1} s_k$ approximates the second-order curvature $s_k^T Hf(x_{k+1})s_k$ with a higher precision than $s_k^T B_{k+1} s_k$.

The following result can be obtained from Relations (10) and Theorem 2.

Corollary 1 (Zhang and Xu 2001) *If f is a cubic function then $s_k^T \hat{B}_{k+1} s_k = s_k^T Hf(x_{k+1})s_k$.*

In the class of Huang, for preserving the positive definiteness instead of the curvature condition, the following condition should be satisfied:

$$s_k^T \hat{y}_k = s_k^T y_k + \theta_k > 0. \tag{11}$$

In order for condition (11) to hold, two methods are suggested in Zhang and Xu (2001). In the first method, using the Wolfe line search and, in the second one, imposing the following condition on θ_k are offered.

$$\theta_k := (\varepsilon - 1)s_k^T y_k, \quad if \quad \theta_k < (\varepsilon - 1)s_k^T y_k, \ \ \varepsilon \in (0, 1). \tag{12}$$

Clearly, condition (12) implies relation $s_k^T \hat{y}_k \geqslant \varepsilon s_k^T y_k$. Thus, by considering condition (12) and the curvature condition, the updating formula of the Huang class preserves positive definiteness. Since the first method (using the Wolfe line search) imposes a large computational effort, in this paper, we use the second method to preserve the positive definiteness property of the matrix obtained from the updating formula of the Huang class.

Based on the above statements, if $f(.)$ is a strongly convex function, then the matrix B_{k+1} obtained from each of the mentioned updating formulas for approximating the Hessian matrix always preserves positive definiteness. Thus, in the rest of this paper, we assume that all components of the vector function F in Problem (1) are strongly convex.

3.2 Quasi-Newton methods for multiobjective optimization

Recently, the BFGS quasi-Newton method for single objective optimization problems has been extended to multiobjective optimization by Qu et al. (2011) and Povalej (2014). In a similar manner, all of the quasi-Newton methods can be generalized to multiobjective problems. In this paper, besides the BFGS QNMO method, we further consider the SS-BFGS and H-BFGS QNMO methods. In this subsection, we briefly explain an extension of the quasi-Newton methods to multiobjective problems.

As seen in the previous subsection, in order to find the search direction at x, we have to solve Subproblem (2). The direct extension of this idea to a vector valued function $F : V \to \mathbb{R}^m$ leads to the following multiobjective subproblem for finding a search direction at x:

$$\begin{aligned} &\min_{d\in\mathbb{R}^n} \ \nabla F_1(x)^T d + \tfrac{1}{2}d^T B^1(x)d \\ &\min_{d\in\mathbb{R}^n} \ \nabla F_2(x)^T d + \tfrac{1}{2}d^T B^2(x)d \\ &\ \vdots \\ &\min_{d\in\mathbb{R}^n} \ \nabla F_m(x)^T d + \tfrac{1}{2}d^T B^m(x)d, \end{aligned} \tag{13}$$

where $B^j(x)$ is an approximation of the Hessian matrix of jth objective function F_j at x.

Subproblem (13) is a nice quadratic multiobjective optimization subproblem because it has some properties as given in Theorem 3. Any objective function of (13) is a bounded-below quadratic strictly convex function. The following theorem holds for Subproblem (13).

Theorem 3 *Let Y and Y_N denote the feasible set in the objective space and the set of all nondominated points of Subproblem (13), respectively. Suppose that $B^j(x) \succ 0$*

for all $j \in \{1, 2, \dots, \}$ *and* $x \in V$. *Then Subproblem* (13) *satisfies the following properties:*

1. *Y is an $\mathbb{R}^m_{\geq}$-compact set.*
2. *The efficient set of* (13) *is nonempty.*
3. *Y_N is externally stable.*
4. *Y_N is connected.*

Proof 1. $B^j(x)$ is a positive definite matrix, thus the jth objective function of (13) is bounded below. Hence, Y is $\mathbb{R}^m_{\geq}$-compact.
2. Since Y is $\mathbb{R}^m_{\geq}$-compact, by Corollary 2.15 in Ehrgott (2005), $Y_N \neq \emptyset$.
3. External stability of Y_N is concluded from Part (1).
4. This part is a direct consequence of Theorem 3.35 in Ehrgott (2005).

□

In what follows, the relation between Subproblem (13) and the criticality of a given point $x \in V$ is investigated.

Theorem 4 *Let $B^j(x) \succ 0$ for all $x \in V$ and $j \in \{1, \dots, m\}$. Then for $F : V \to \mathbb{R}^m$ the following propositions are equivalent:*

1. *x is a critical point for F;*
2. *$\hat{d} = 0$ is a weakly efficient solution of Subroblem* (13)*;*
3. *There is a vector $\hat{\lambda} \in \mathbb{R}^m_{\geq}$ with $\sum_{j=1}^m \hat{\lambda}_j = 1$ such that*

$$\forall d \in \mathbb{R}^n \setminus \{0\}, \sum_{j=1}^{m} \hat{\lambda}_j \left(\nabla F_j(x)^T d + \frac{1}{2} d^T B^j(x) d \right) > 0,$$

4. *There is a vector $\hat{\lambda} \in \mathbb{R}^m_{\geq}$ with $\sum_{j=1}^m \hat{\lambda}_j = 1$ such that*

$$\sum_{j=1}^{m} \hat{\lambda}_j \nabla F_j(x) = 0.$$

Proof 1. $(1 \Longrightarrow 2)$
By contradiction, assume that $\hat{d} = 0$ is not a weakly efficient solution of Subproblem (13). Hence there exists $d \in \mathbb{R}^n$ such that

$$\nabla F_j(x)^T d \leq \nabla F_j(x)^T d + \frac{1}{2} d^T B^j(x) d < 0, \quad \forall j \in \{1, \dots, m\}.$$

Thus, x is not critical, which contradicts (1).
2. $(2 \Longrightarrow 3)$
Because $\hat{d} = 0$ is a weakly efficient solution and (13) is a strictly convex multiobjective subproblem, there is a vector $\hat{\lambda} \in \mathbb{R}^m_{\geq}$ with $\sum_{j=1}^m \hat{\lambda}_j = 1$ such that $\hat{d} = 0$ is the unique optimal solution of the single objective problem $\min_{d \in \mathbb{R}^n} f_{\hat{\lambda}}(d)$, where

$$f_{\hat{\lambda}}(d) := \sum_{j=1}^{m} \hat{\lambda}_j \left(\nabla F_j(x)^T d + \frac{1}{2} d^T B^j(x) d \right).$$

This completes the proof.

3. (3 $\Longrightarrow$ 4)
Since $f_{\hat{\lambda}}(d)$ is a strictly convex function, $\hat{d} = 0$ is the minimizer of $f_{\hat{\lambda}}(d)$ if and only if

$$\nabla_d f_{\hat{\lambda}}(0) = \sum_{j=1}^{m} \hat{\lambda}_j \nabla F_j(x) = 0.$$

4. (4 $\Longrightarrow$ 1)
By contradiction, assume that Part (4) holds, but x is not critical. Then there exists $d \in \mathbb{R}^n$ such that $\nabla F_j(x)^T d < 0$ for all $j \in \{1, \ldots, m\}$. Therefore, for any $\lambda \in \mathbb{R}^n_{\geq}$ with $\sum_{j=1}^{m} \lambda_j = 1$, we have $\sum_{j=1}^{m} \lambda_j \nabla F_j(x) d < 0$ which is a contradiction to the assumption.

□

As mentioned in Povalej (2014), the quasi-Newton search direction at $x \in V$ is the unique minimizer of the following minimax subproblem which is a weakly efficient solution of the multiobjective optimization Subproblem (13).

$$\min_{d \in \mathbb{R}^n} \max_{j \in \{1,..,m\}} \nabla F_j(x)^T d + \frac{1}{2} d^T B^j(x) d. \tag{14}$$

This suproblem is evidently equivalent to the following subproblem:

$$\begin{aligned} &\min \ h(t,d) = t \\ &s.t \ \begin{cases} \nabla F_j(x)^T d + \frac{1}{2} d^T B^j(x) d \leq t, \quad (1 \leq j \leq m), \\ (t,d) \in \mathbb{R} \times \mathbb{R}^n, \end{cases} \end{aligned} \tag{15}$$

Subproblem (15) is a convex subproblem with quadratic constraints. From now on, for each $x \in V$, we assume that $d(x)$ and $\alpha(x)$ are the optimal solution and the optimal value of Subproblem (14), respectively; that is

$$\alpha(x) := \inf_{d \in \mathbb{R}^n} \max_{j=1,\ldots,m} (\nabla F_j(x)^T d + \frac{1}{2} d^T B^j(x) d), \tag{16}$$

and

$$d(x) := \arg \min_{d \in \mathbb{R}^n} \max_{j=1,\ldots,m} (\nabla F_j(x)^T d + \frac{1}{2} d^T B^j(x) d). \tag{17}$$

For completeness, Theorem 5 from Fliege et al. (2009), Povalej (2014) is given, which states some properties of $\alpha(x)$ and $d(x)$.

Theorem 5 *Consider $\alpha(x)$ and $d(x)$ as defined by (16) and (17), respectively. Let $B^j(x) \succ 0$ for all $x \in V$. Then*

1. *For each $x \in V$, $\alpha(x) \leq 0$*
2. *The following conditions are equivalent:*
   (a) *Point x is not critical,*
   (b) *$\alpha(x) < 0$,*
   (c) *$d(x) \neq 0$.*
3. *The function $d : V \to \mathbb{R}^n$ is bounded on compact sets and $\alpha : V \to \mathbb{R}$ is continuous.*

The interested readers are referred to Fliege et al. (2009), Povalej (2014) for the proof of Theorem 5.

Here, at first, a brief review of the Karush–Kuhn–Tucker (KKT) conditions is given (Fliege et al. 2009; Povalej 2014). At the end of this section, a simple representation of these conditions is proposed, which is based on the multiobjective Subproblem (13). Using the Lagrangian function of Subproblem (15), an explicit representation of $d(x)$ in terms of $\nabla F_j(x)$ and $B^j(x)$ is stated. To this aim, consider the Lagrangian function of Subproblem (15) as

$$L((t,d),\lambda) = t + \sum_{j=1}^{m} \lambda_j \left(\nabla F_j(x)^T d + \frac{1}{2} d^T B^j(x) d - t \right).$$

Subproblem (15) has the unique optimal solution $(t^*, d^*) := (\alpha(x), d(x))$ and a Slater point $(t, d) = (1, 0)$. Hence, there exists a KKT multiplier $\lambda := \lambda(x)$, that, for $d := d(x)$ and $\alpha := \alpha(x)$ satisfies the following conditions:

$$\sum_{j=1}^{m} \lambda_j = 1, \sum_{j=1}^{m} \lambda_j \left(\nabla F_j(x) + B^j(x) d \right) = 0, \tag{18}$$

$$\lambda_j \left(\nabla F_j(x)^T d + \frac{1}{2} d^T B^j(x) d - t \right) = 0, \qquad (1 \leq j \leq m), \tag{19}$$

$$\lambda_j \geq 0, \qquad \nabla F_j(x)^T d + \frac{1}{2} d^T B^j(x) d \leq t, \quad (1 \leq j \leq m). \tag{20}$$

In particular, from (18) we have

$$d(x) = -\left(\sum_{j=1}^{m} \lambda_j(x) B^j(x) \right)^{-1} \left(\sum_{j=1}^{m} \lambda_j(x) \nabla F_j(x) \right). \tag{21}$$

The existence of the KKT multipliers for the convex Subproblem (15) implies that there is no duality gap, and so

$$\begin{aligned} \alpha(x) &= \sup_{\substack{\lambda \geq 0 \\ \sum \lambda_j = 1}} \inf_{d \in \mathbb{R}^n} L((t,d),\lambda) \\ &= \sup_{\substack{\lambda \geq 0 \\ \sum \lambda_j = 1}} \inf_{d \in \mathbb{R}^n} \sum_{j=1}^{m} \lambda_j \left(\nabla F_j(x)^T d + \frac{1}{2} d^T B^j(x) d - t \right). \end{aligned} \tag{22}$$

Theorem 23 shows that the KKT conditions (18) can be obtained easily using the multiobjective Subproblem (13).

Theorem 6 *Let $\hat{d}$ be the quasi-Newton direction of function F at x. Then there exists $\hat{\lambda} \in \mathbb{R}^m_{\geq}$ such that*

$$\sum_{j=1}^{m} \hat{\lambda}_j = 1, \quad \sum_{j=1}^{m} \hat{\lambda}_j \left(\nabla F_j(x) + B^j(x)\hat{d} \right) = 0. \tag{23}$$

Proof Assume that $\hat{d}$ is the quasi-Newton direction of F in x. Then, there is a scalar $\hat{t}$ such that $(\hat{t}, \hat{d})$ is an optimal solution of Subproblem (15). Thus, $\hat{d}$ is a weakly efficient solution of the convex multiobjective Subroblem (13). Therefore, there exists $\hat{\lambda} \in \mathbb{R}^m_{\geq}$ with $\sum_{j=1}^{m} \hat{\lambda}_j = 1$ such that $\hat{d}$ is the minimum of function $f_{\hat{\lambda}}(d) := \sum_{j=1}^{m} \lambda_j \left(\nabla F_j(x)^T d + \frac{1}{2} d^T B^j(x) d \right)$. Hence

$$\nabla_d f_{\hat{\lambda}}(\hat{d}) = \sum_{j=1}^{m} \hat{\lambda}_j \left(\nabla F_j(x) + B^j(x)\hat{d} \right) = 0,$$

and this is precisely the assertion of the theorem. □

Assume that $d(x) \in \mathbb{R}^n$ is a direction with $JF(x)d(x) < 0$. To compute a step length $t > 0$, an Armijo-like rule is used. Let $\sigma \in (0, 1)$ be a prescribed constant. The condition to accept t is

$$F_j(x + td(x)) \leq F_j(x) + \sigma t \alpha(x), \quad \text{for all } j \in \{1, \ldots, m\}.$$

Start with $t = 1$ and, while this condition is not satisfied, set $t := \frac{t}{2}$. Using this line search, we find the appropriate step length in Algorithm 1.

Povalej (2014) and Qu et al. (2011) used the Armijo line search in the implementation of the BFGS quasi-Newton method for multiobjective optimization. The convergence of this method was proven in Povalej (2014) and Qu et al. (2011). In this paper, besides the BFGS quasi-Newton method, some other well-known quasi-Newton methods are implemented for MOP in two cases: with line search (BFGSLS, SS-BFGSLS and H-BFGSLS) and without any line search (BFGS, SS-BFGS and H-BFGS). Related to these two cases, two algorithms are given below.

Algorithm 1. QNMO with Armijo line search

1. (Initialization)
   (a) Choose $\varepsilon > 0$, $x_0 \in V$, $\sigma \in (0, 1)$ and set $B_0^j = I$ for all $j \in \{1, 2, \ldots, m\}$.
   (b) Solve Subproblem (15) for x_0 to obtain $d_0 = d(x_0)$ and $\alpha_0 = \alpha(x_0)$. If $|\alpha_0| \leq \varepsilon$, then stop. Otherwise, set $x_1 = x_0 + d_0$, $k = 1$ and go to the Main loop.

2. (Main loop)
   (a) Solve Subproblem (15) for x_k to obtain $d_k = d(x_k)$ and $\alpha_k = \alpha(x_k)$.
   (b) If $|\alpha_k| \leq \varepsilon$ then stop. Otherwise, find the largest $t_k = \frac{1}{2^n}$ for $n = 1, 2, 3, \ldots$ that satisfies the following conditions

$$\begin{aligned} & x_k + t_k d_k \in V \\ & F_j(x_k + t_k d_k) \leq F_j(x_k) + \sigma t_k \alpha_k \end{aligned} \tag{24}$$

   and set $x_{k+1} := x_k + t_k d_k$.
   (c) Update B_k^j (for $j \in \{1, 2, \ldots, m\}$) using one of the update formulas (4), (5) or (8). Set $k = k + 1$ and go to Step 2(a).

Algorithm 2. QNMO without any line search

1. (Initialization)
   (a) Choose $\varepsilon > 0$, $x_0 \in V$, $\sigma \in (0, 1)$ and set $B_0^j = I$ for all $j \in \{1, 2, \ldots, m\}$.
   (b) Solve Subproblem (15) for x_0 to obtain $d_0 = d(x_0)$ and $\alpha_0 = \alpha(x_0)$. If $|\alpha_0| \leq \varepsilon$, then stop. Otherwise, set $x_1 = x_0 + d_0$, $k = 1$ and go to the Main loop.
2. (Main loop)
   (a) Solve Subproblem (15) for x_k to obtain $d_k = d(x_k)$ and $\alpha_k = \alpha(x_k)$.
   (b) If $|\alpha_k| \leq \varepsilon$ then stop. Otherwise, set $x_{k+1} = x_k + d_k$.
   (c) Update B_k^j (for $j \in \{1, 2, \ldots, m\}$) using one of the update formulas (4), (5) or (8). Set $k = k + 1$ and go to Step 2(a).

The convergence analysis for Algorithm 1 (with Armijo line search) has been presented in Povalej (2014) and the proof of Algorithm 2 is similar to it. However, to facilitate access and make our exposition self-contained, the proof of Algorithm 2 is rendered in the next section.

4 Convergence analysis

At the beginning of this section, we state some lemmas from Fliege et al. (2009) and Povalej (2014), which are needed to prove the convergence theorem.

Lemma 2 (Fliege et al. 2009) *Suppose that $U \subset V$ is a convex set. Consider $\varepsilon > 0$ and $\delta > 0$ such that for any $x,\ y \in U$ with $\|y - x\| < \delta$,*

$$\left\| HF_j(y) - HF_j(x) \right\| < \varepsilon \ \ for\ all\ \ j \in \{1, \ldots, m\}. \tag{25}$$

Under this assumption, for any $x,\ y \in U$ such that $\|y - x\| < \delta$, we have

$$\left\| \nabla F_j(y) - [\nabla F_j(x) + HF_j(x)(y - x)] \right\| < \varepsilon \|y - x\|, \tag{26}$$

for all $j \in \{1, \ldots, m\}$.

Lemma 3 (Povalej 2014) *Let $U \subset V$ be a convex subset and $\varepsilon, \delta \in \mathbb{R}_{\geq}$ be small constants such that for any $x, y \in U$, with $\|y-x\| < \delta$, (25) holds. Let $\{x_k\}$ be a sequence generated by a quasi-Newton method and $\{B_k^j\}$ $(j = 1, 2, \ldots, m)$ be the sequence generated by one of the update formulas (4), (5) or (8). Also, assume that for $\varepsilon > 0$ there exists $k_0 \in \mathbb{N}$ such that for all $k \geq k_0$*

$$\frac{\left\|\left(HF_j(x_k) - B_k^j\right)(y - x_k)\right\|}{\|y - x_k\|} < \frac{\varepsilon}{2} \quad for\ all\ j = 1, \ldots, m. \tag{27}$$

Under these assumptions, for any $k \geq k_0$, and any $y \in U$ such that $\|y - x_k\| < \delta$, we have

$$\left\|\nabla F_j(y) - \left(\nabla F_j(x_k) + B_k^j(y - x_k)\right)\right\| < \varepsilon \|y - x_k\|, \tag{28}$$

for all $j \in \{1, \ldots, m\}$.

Lemma 4 (Fliege et al. 2009; Povalej 2014) *Let $x \in V$ and $a, b \in \mathbb{R}_{>}$ such that $a \leq b$. Also, consider $B^j(x)$ as an approximation of $HF_j(x)$. If*

$$aI \preceq B^j(x) \preceq bI, \;\; for\ all\ \ j = 1, 2, \ \ldots, m, \tag{29}$$

then the following two inequalities hold,

$$\begin{cases} i)\ \frac{a}{2}\|d(x)\|^2 \leq |\alpha(x)| \leq \frac{b}{2}\|d(x)\|^2, \\ ii)\ |\alpha(x)| \leq \frac{1}{2a}\left\|\sum\limits_{j=1}^{m} \lambda_j \nabla F_j(x)\right\|^2 for\ all\ \lambda_j \geq 0, \\ \qquad j = 1, 2, \ldots, m,\ with\ \sum\limits_{j=1}^{m} \lambda_j = 1, \end{cases} \tag{30}$$

where $\alpha(x)$ and $d(x)$ are defined in (16) and (17), respectively.

During the following theorem and its corollary, using Lemmas 2-4 presented in this section, we will prove the convergence of Algorithm 2.

Theorem 7 *Suppose $\{x_k\}$ is an arbitrary sequence generated by Algorithm 2 and $x_0 \in V$ is the initial point of it. Take $B_k^j := B^j(x_k)$ as an approximation of the Hessian matrix at x_k corresponding to jth objective function of F. Let there be $r > 0$ such that $G[x_0, r] \subset V$ in which $G[x_0, r]$ is a closed ball with radius r centered at x_0. If we have $a, \delta, \varepsilon > 0$, with $a > \varepsilon$ such that*

1. *$aI \preceq B^j(x)$ and $aI \preceq HF_j(x)$ for all $j = 1, 2, \ldots, m$ and $x \in G[x_0, r]$,*
2. *$\left\|HF_j(y) - HF_j(x)\right\| < \varepsilon$, for all $j = 1, \ldots, m$ and $x, y \in G[x_0, r]$ with $\|y - x\| < \delta$,*

3. $\frac{\left\|\left(HF_j(x_k)-B_k^j\right)(y-x_k)\right\|}{\|y-x_k\|} < \frac{\varepsilon}{2}$*, for all* $j=1,\ldots,m$, $y \in G[x_0,r]$ *and* $k \geq k_0$*,*

4. $\|d_0\| < \min\{\delta, r(1-\frac{\varepsilon}{a})\}$,

then, for any $k \in \mathbb{N}$*,*

a. $x_k \in G[x_0,r]$,
b. $\|d_k\| < \left(\frac{\varepsilon}{a}\right)^k \|d_0\|$.

Proof The proof is based on induction. Clearly, the proposition holds for $k=0$. Now, assume that the proposition holds for $k \in \mathbb{N}$. We show that it also holds for $k+1$. $\{x_k\}$ is a sequence generated by Algorithm 2. Thus, $x_{k+1} = x_k + d_k$. Since the proposition holds for $k \in \mathbb{N}$, we have

$$\|d_k\| \leq \|d_0\| \left(\frac{\varepsilon}{a}\right)^k \ \ and \ \ x_k \in G[x_0,r]. \tag{31}$$

Consequently, by the triangular inequality, and with the sum of the first $k+1$ terms of a geometric series and the assumption $a > \varepsilon$, it is concluded that

$$\|x_{k+1}-x_0\| \leq \sum_{i=0}^{k} \|x_{i+1}-x_i\| \leq \|d_0\| \frac{1-\left(\frac{\varepsilon}{a}\right)^{k+1}}{1-\frac{\varepsilon}{a}} < r, \tag{32}$$

and this completes the proof of Part (a); that is $x_{k+1} \in G[x_0,r]$.

It is obvious that

$$\|x_{k+1}-x_k\| \leq \delta. \tag{33}$$

According to (33), the assumptions (2) and (3), and Lemma 3, it follows immediately that

$$\left\|\nabla F_j(x_{k+1}) - \left(\nabla F_j(x_k) + B_k^j d_k\right)\right\| < \varepsilon \|x_{k+1}-x_k\|, \qquad \forall j \in \{1,2,\ldots,m\}. \tag{34}$$

Let λ_k^j $(j=1,\ldots,m)$ be the KKT multipliers of Subproblem (15) at x_k. Hence, applying (34) yields

$$\sum_{j=1}^{m} \lambda_k^j \left\|\nabla F_j(x_{k+1}) - \left(\nabla F_j(x_k) + B_k^j d_k\right)\right\| < \sum_{j=1}^{m} \lambda_k^j \varepsilon \|d_k\|. \tag{35}$$

According to the properties of the KKT multipliers, including $\sum_{j=1}^{m} \lambda_k^j = 1$ and $\lambda_k^j \geq 0$ for all $j=1,2,\ldots,m$ and from (35), we have

$$\sum_{j=1}^{m} \left\|\lambda_k^j \nabla F_j(x_{k+1}) - \left(\lambda_k^j \nabla F_j(x_k) + \lambda_k^j B_k^j d_k\right)\right\| < \varepsilon \|d_k\|. \tag{36}$$

Using triangular inequality, we can rewrite (36) as

$$\left\| \sum_{j=1}^{m} \lambda_k^j \nabla F_j(x_{k+1}) - \left(\sum_{j=1}^{m} \lambda_k^j \nabla F_j(x_k) + \left(\sum_{j=1}^{m} \lambda_k^j B_k^j \right) d_k \right) \right\| < \varepsilon \left\| d_k \right\|. \tag{37}$$

(21) makes it obvious that

$$d_k = - \left(\sum_{j=1}^{m} \lambda_k^j B_k^j \right)^{-1} \left(\sum_{j=1}^{m} \lambda_k^j \nabla F_j(x_k) \right). \tag{38}$$

Substituting (38) into (37) yields

$$\left\| \sum_{j=1}^{m} \lambda_k^j \nabla F_j(x_{k+1}) \right\| < \varepsilon \left\| d_k \right\|. \tag{39}$$

According to Part (ii) of Lemma 4 and (39), we deduce that

$$|\alpha(x_{k+1})| < \frac{\varepsilon^2}{2a} \|d_k\|^2. \tag{40}$$

By (40) and Part (i) of Lemma 4, it is obvious that

$$\|d_{k+1}\| < \frac{\varepsilon}{a} \left\| d_k \right\|, \tag{41}$$

and, consequently, we have

$$\|d_{k+1}\| < \left(\frac{\varepsilon}{a} \right)^{k+1} \left\| d_0 \right\|, \tag{42}$$

which completes the proof. □

Corollary 2 *Under the assumptions of Theorem 7, the sequence generated by Algorithm 2 converges to a critical point.*

Proof Suppose $\{x_k\}$ is the obtained sequence from Algorithm 2. Then, from Theorem 7, for $\epsilon > 0$ there exists $m_0 \in \mathbb{N}$ such that for all n and m, where $n > m \geq m_0$ we have:

$$\|x_n - x_m\| \leq \sum_{i=m}^{n-m} \|x_{i+1} - x_i\| = \sum_{i=m}^{n-m} \|d_i\| \leq \|d_0\| \sum_{i=m}^{n-m} \left(\frac{\varepsilon}{a} \right)^i. \tag{43}$$

Since $a > \varepsilon$ (from the assumptions of Theorem 7), (43) shows that $\{x_k\}$ is a Cauchy sequence. Thus, it is clear that $\{x_k\}$ is a convergent sequence and there exists $\hat{x} \in \mathbb{R}^n$, such that

$$\lim_{k \to \infty} x_k = \hat{x}.$$

Besides, from Part (b) of Theorem 7 it is evident that

$$\lim_{k\to\infty} d_k = 0 \tag{44}$$

By Part (i) of Lemma 4, we get that $\lim_{k\to\infty} \alpha(x_k) = 0$. According to Part (3) of Theorem 5, $\alpha(x)$ is a continuous function. Hence, $\alpha(\hat{x}) = 0$. Thus, from Part (2) of Theorem 5 it is obvious that $\hat{x}$ is a critical point. □

In Corollary 2, we show that every initial point of Algorithm 2 converges to a critical point. To find the relations between critical point and (weakly) efficient solution, one can use Theorem 1.

5 Performance assessment

Unlike single objective methods, comparing the performance of different multiobjective optimization methods is a difficult task. This difficulty is due to the evaluation of the quality of the approximated nondominated frontier. There are several criteria that indicate the quality of the approximated nondominated frontier. In this paper, we use three criteria: the purity metric, the spread metric, Custodio et al. (2011), Bandyopadhyay et al. (2004) and the epsilon indicator (Knowles et al. 2006). Moreover, in order to have a good visual comparison, the numerical results are depicted using performance profiles (Dolan and More 2002). To this aim, in what follows we present a brief description of performance profiles and some performance assessment criteria.

5.1 Performance profiles

In order to compare different algorithms on a set of test problems, one can run these algorithms on given test problems and record such information as the number of iterations, the number of function evaluations, expended CPU time, and quality criteria. Considering these criteria, the performance of algorithms can be compared with each other using some tables. If the set of problems or the set of algorithms is suitably large, comparing these criteria using tables can be very difficult for readers. In 2002, Dolan and More introduced performance profiles as a means to evaluate and compare the performance of a set of algorithms S on a set of test problems P (Dolan and More 2002). In what follows, we briefly explain this method.

Let $t_{p,s}$ denote the performance of the algorithm $s \in S$ in solving the problem $p \in P$. This performance ($t_{p,s}$) must have the property in which better performances correspond to lower values of $t_{p,s}$. If the intended performance ($\bar{t}_{p,s}$) did not have this property, we would have to use $t_{p,s} = 1/\bar{t}_{p,s}$ instead of $\bar{t}_{p,s}$ as the performance. For example, one can consider $t_{p,s}$ as the expended CPU times in solving problem p by algorithm s. However, if $\bar{t}_{p,s}$ is the number of the obtained nondominated points in solving problem p by algorithm s, in order to plot the performance profiles, we have to consider $t_{p,s} = 1/\bar{t}_{p,s}$ instead of $\bar{t}_{p,s}$ as the performance. Accordingly, the performance ratio for the algorithm $s \in S$ in solving the problem $p \in P$ is defined as follows:

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}. \tag{45}$$

If the denominator in (45) is zero, we instead select the smallest positive value greater than zero in $\{t_{p,s} : s \in S\}$. Using the performance ratio, one can compare the performance of algorithm s on problem p with the best performance compared to the performance of the other algorithm for this problem. The performance of algorithm s on any given problem can be of interest, but we would like to obtain an overall assessment of the performance of the algorithm. Thus, $\rho_s(\tau)$ is defined as follows

$$\rho_s(\tau) = \frac{1}{|P|} \left|p \in P : r_{p,s} \leq \tau\right|. \tag{46}$$

$\rho_s(\tau)$ is the probability for algorithm $s \in S$ whose performance ratio is within the interval $[1, \tau]$. Indeed, ρ_s is the (cumulative) distribution function for the performance ratio. The performance profiles depict the function $\rho_s(\tau)$ for each algorithm $s \in S$.

The value of $\rho_s(1)$ is the probability that the algorithm s 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 $\rho_s(1)$ for all of the algorithms. For example, $\rho_{s_0}(1) = 0.6$ means that the algorithm s_0 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 $\rho_s(\tau)$ for large values of τ have a better performance than the other algorithms.

5.2 Purity metric

The purity metric is an important criterion for comparing the nondominated frontiers obtained by different algorithms (Custodio et al. 2011; Bandyopadhyay et al. 2004). The approximated nondominated frontier obtained by algorithm $s \in S$ for problem $p \in P$ is denoted by $F_{p,s}$. After removing the dominated elements from set $\cup_{s\in S} F_{p,s}$, a set, namely the reference nondominated frontier which is denoted by F_p is obtained. The purity metric for algorithm $s \in S$ and problem $p \in P$ is defined as $\bar{t}_{p,s} = c^{F_p}_{p,s}/c_{p,s}$ where $c^{F_p}_{p,s} = \left|F_{p,s} \cap F_p\right|$ and $c_{p,s} = \left|F_{p,s}\right|$. Higher values of $\bar{t}_{p,s}$ show that algorithm $s \in S$ yields a higher percentage of the nondominated points for problem $p \in P$. Thus, we have to consider $t_{p,s} = 1/\bar{t}_{p,s}$ instead of $\bar{t}_{p,s}$ as the performance. Clearly, $\bar{t}_{p,s} = 0$ or $t_{p,s} = \infty$ shows that the set of points obtained by algorithm s in solving problem p does not have any intersection with the reference nondominated frontier (F_p).

5.3 Spread metric

The spread metric is another metric to measure the spread of the obtained nondominated frontiers (Custodio et al. 2011; Bandyopadhyay et al. 2004). Naturally, we are interested in computing a set of points which spans the entire exact nondominated frontier. The spread metric is an attempt to measure the largest gap in the obtained nondominated frontiers. In the following, we explain how one can compute the value of the spread metric for each algorithm s in solving the problem p. First, we have to compute the extreme points of the reference nondominated frontier (F_p) in the objective space of the problem p. There is an extreme point corresponding to each component of the objective function. In other words, we have k extreme points for a k dimensional problem. The extreme point corresponding to the kth component is a point in the reference nondominated frontier whose kth component has the largest value. Assume that algorithm s has obtained a set of N points indexed by $1, \ldots, N$ as the approximated nondominated set. Add the extreme points to these points indexed by $N+1, \ldots, N+k$ (in k dimensional objective space). Let $f_{i,j}^{p,s}$ denote the jth component of the ith point of this collection. It is obvious that $f_{i,j}^{p,s}$ is a scalar value. Suppose that the values of $f_{i,j}^{p,s}$ were sorted by an increasing order in i for each j. Then, the spread metric of the algorithm s in solving the problem p is denoted by $\Gamma_{p,s}$ and is defined as follows:

$$\Gamma_{p,s} = \max_{j \in \{1,\ldots,m\}} \left(\max_{i \in \{1,\ldots,N+k\}} \{\delta_{i,j}^{p,s}\} \right),$$

where $\delta_{i,j}^{p,s} = (f_{i+1,j}^{p,s} - f_{i,j}^{p,s})$.

5.4 Epsilon indicator

The epsilon indicator is a criterion for representing the quality of the obtained nondominated frontiers. The epsilon indicator has two versions, multiplicative and additive. In this paper, we use the additive version. The additive epsilon indicator $I_{\varepsilon+}(A)$ is defined with respect to the nondominated reference set F_p as:

$$I_{\varepsilon+}(A) = I_{\varepsilon+}(A, F_p) = \inf\{\varepsilon | \forall y \in F_p \ \exists z \in A : z \preceq_{\varepsilon+} y\},$$

where A is the obtained nondominated set for the specified algorithm and $\preceq_{\varepsilon+}$ is the additive ε-dominance inequality which is defined as follows:

$$z^1 \preceq_{\varepsilon+} z^2 \Leftrightarrow \forall i \in \{1, \ldots, m\}, \ z_i^1 \leq \varepsilon + z_i^2,$$

For simplicity of the notation, from now on, BFGS, self-scaling BFGS, and Huang BFGS quasi-Newton methods for multiobjective optimization without any line search are denoted by BFGS, SS-BFGS, and H-BFGS, respectively. In addition, BFGS, self-scaling BFGS, and Huang BFGS quasi-Newton methods for multiobjective optimization equipped with the Armijo line search are denoted by BFGSLS, SS-BFGSLS, and H-BFGSLS, respectively. In the following, we use the above mentioned performance assessment criteria, the number of iterations, the number of function evaluations, expended CPU time, and some other criteria to compare the BFGS, BFGSLS, SS-BFGS, SS-BFGSLS, H-BFGS, and H-BFGSLS methods.

6 Numerical results

A MATLAB implementation of the proposed QNMO methods was tested on some well-known test problems from the literature. Table 1 describes these test problems. The first column of this table contains the names of these problems. The dimensions of the test problems are given in Column 2. The notations x_L and x_U in Columns 2 and 3 stand for the lower and upper bounds of the decision variables. The last column provides a literature reference. All tests were executed within MATLAB R2013a and were solved 100 times using 100 initial points from a uniform random distribution between the given lower and upper bounds specified in Table 1. Therefore, all results reported in Tables 2, 3 and 4 and depicted in Figs. 1, 2, 3, 4, 5 and 6 are the average number of the obtained results from these 100 initial points. The corresponding approximations of the nondominated frontier of all methods are plotted in Figs. 7, 8, 9 and 10 for some of the test problems. In order to have a good approximation of the nondominated frontiers in these figures, the test problems with two and three objective functions are solved for 500 and 300 initial points, respectively. As mentioned before,

Table 1 Description of all problems used in numerical implementations

Name	n	x_L	x_U	References
Hil	2	$[0, 0]$	$[5, 5]$	Hillermeier (2001)
DD1a	5	$-[1, \dots, 1]$	$[1, \dots, 1]$	Das and Dennis (1998)
DD1b	5	$-[5, \dots, 5]$	$[5, \dots, 5]$	Das and Dennis (1998)
DD1c	5	$-10[1, \dots, 1]$	$10[1, \dots, 1]$	Das and Dennis (1998)
DD1d	5	$-20[1, \dots, 1]$	$20[1, \dots, 1]$	Das and Dennis (1998)
KW2	2	$[-3, -3]$	$[3, 3]$	Kim and De Weck (2005)
JOS1a	50	$-[2, \dots, 2]$	$[2, \dots, 2]$	Ex. 1 Jin et al. (2001)
JOS1b	100	$-[2, \dots, 2]$	$[2, \dots, 2]$	Ex. 1 Jin et al. (2001)
JOS1c	200	$-[2, \dots, 2]$	$[2, \dots, 2]$	Ex. 1 Jin et al. (2001)
JOS1d	500	$-[2, \dots, 2]$	$[2, \dots, 2]$	Ex. 1 Jin et al. (2001)
JOS1e	100	$-10[1, \dots, 1]$	$10[1, \dots, 1]$	Ex. 1 Jin et al. (2001)
JOS1f	100	$-50[1, \dots, 1]$	$50[1, \dots, 1]$	Ex. 1 Jin et al. (2001)
JOS1g	100	$-100[1, \dots, 1]$	$100[1, \dots, 1]$	Ex. 1 Jin et al. (2001)
JOS1h	200	$-100[1, \dots, 1]$	$100[1, \dots, 1]$	Ex. 1 Jin et al. (2001)
PNR	2	$-[2, 2]$	$[2, 2]$	Preuss et al. (2006)
ZDT6a	3	$[0, 0, 0]$	$[1, 1, 1]$	Ex. 6 Zitzler et al. (2000)
ZDT6b	10	$[0, \dots, 0]$	$[1, \dots, 1]$	Ex. 6 Zitzler et al. (2000)
LTDZ	3	$[0, 0, 0]$	$[1, 1, 1]$	Laumanns et al. (2002)
TR1	3	$[0, 0, 0]$	$[1, 1, 1]$	Tappeta and Renaud (1999)
FDSa	10	$-[2, \dots, 2]$	$[2, \dots, 2]$	Fliege et al. (2009)
FDSb	50	$-[2, \dots, 2]$	$[2, \dots, 2]$	Fliege et al. (2009)
FDSc	100	$-[2, \dots, 2]$	$[2, \dots, 2]$	Fliege et al. (2009)
FDSd	200	$-[2, \dots, 2]$	$[2, \dots, 2]$	Fliege et al. (2009)

Table 2 The average number of iterations (Iter) and function evaluations (Feval) for BFGSLS, SS-BFGSLS, and H-BFGSLS QNMO methods in solving some test problems for all initial points are compared

Name	BFGSLS		SS-BFGSLS		H-BFGSLS	
	Iter	Feval	Iter	Feval	Iter	Feval
Hil	24.57	207.93	24.47	207.07	**2.25**	**4.06**
DD1a	6.34	6.53	11.27	55.28	**6.29**	**6.38**
DD1b	8.32	8.32	32.70	251.06	**8.28**	**8.28**
DD1c	66.85	544.26	54.24	426.94	7.71	8.15
DD1d	18.65	135.86	135.57	1202.29	**3.38**	**3.39**
KW2	7.48	52.45	7.35	52.33	**2.37**	**3.15**
JOS1a	42.68	280.20	28.53	130.62	**14.25**	**15.18**
JOS1b	43.17	293.24	**13.89**	**13.89**	14.04	14.59
JOS1c	47.11	306.70	33.39	174.36	**15.34**	**16.56**
JOS1d	26.26	95.83	**13.95**	**13.95**	17.58	18.41
JOS1e	89.53	701.14	89.41	749.69	**16.50**	**17.88**
JOS1f	186.00	1629.84	232.42	2105.45	**20.00**	**23.89**
JOS1g	257.03	2341.36	273.96	2406.39	**21.04**	**25.68**
JOS1h	240.16	2146.27	191.13	1694.08	**21.41**	**23.80**
PNR	11.04	58.87	15.97	106.64	**2.09**	**2.96**
ZDT6a	**1.92**	**1.92**	**1.92**	**1.92**	**1.92**	**1.92**
ZDT6b	**1.91**	**1.91**	**1.91**	**1.91**	**1.91**	**1.91**
LTDZ	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**
TR1	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**
FDSa	473.47	823.00	408.82	594.41	**371.58**	**373.28**
FDSb	69.07	458.04	70.76	549.14	**25.07**	**27.97**
FDSc	128.93	1042.23	203.37	1869.70	**26.60**	**30.55**
FDSd	227.41	2020.30	217.11	2011.50	**27.63**	**32.33**

For each test problem the numbers with the least amount of Iter and CIter are in bold

the initial points for all methods are selected uniformly in the feasible set in the decision space. For instance, see the scatter plot in Fig. 7 which depicts 300 initial points selected to obtain the nondominated frontiers of test Problem PNR by the proposed QNMO methods.

The stop rule is $|\alpha(x)| < \epsilon$, where $\epsilon = 5\sqrt{eps}$ and $eps \cong 2.2204 \times 10^{-16}$. The maximum number of iterations was set to 500. To avoid repetition from now on, an initial point which converges before 500 iterations is called (initial) convergent point and an initial point which does not converge in 500 iterations is called (initial) non-convergent point.

As seen in the Table 1, for each test problem we have box constraints of the form $x_L \leq x \leq x_U$. These constraints are considered under the direction search Subproblem (15) in the way that the new point lies in the same box, i.e., $x_L \leq x + d \leq x_U$ holds. Accordingly, in order to find a new direction in x, we practically use Subproblem (47) instead of Subproblem (15).

Table 3 Comparing the average number of iterations for the BFGS, SS-BFGS, and H-BFGS QNMO methods in solving all test problems for all initial points (Iter) and initial points that are converged during less than 500 iterations (CIter)

Name	BFGS		SS-BFGS		H-BFGS	
	Iter	CIter	Iter	CIter	Iter	CIter
Hil	3.81	3.81	3.97	3.97	**2.89**	**2.89**
DD1a	6.32	6.32	6.41	6.41	**6.29**	**6.29**
DD1b	8.32	8.32	8.77	8.77	**8.28**	**8.28**
DD1c	10.64	10.64	11.54	11.54	**7.86**	**7.86**
DD1d	4.78	4.78	14.38	14.38	**3.42**	**3.42**
KW2	4.62	4.62	**2.83**	**2.83**	3.1	3.1
JOS1a	**15.30**	**15.30**	24.06	19.25	**15.30**	**15.30**
JOS1b	14.43	14.43	**13.89**	**13.89**	14.43	14.43
JOS1c	16.56	16.56	33.39	**13.95**	16.56	16.56
JOS1d	17.80	17.80	**13.95**	**13.95**	17.80	17.80
JOS1e	20.22	20.22	74.26	**16.20**	19.87	19.87
JOS1f	62.57	39.55	122.51	**22.16**	66.44	43.62
JOS1g	132.78	40.97	257.00	**32.69**	132.06	40.07
JOS1h	108.4	55	134.47	**25.28**	101.93	57.7
PNR	11.16	6.22	9.40	4.44	**3.44**	**3.44**
ZDT6a	**1.92**	**1.92**	**1.92**	**1.92**	**1.92**	**1.92**
ZDT6b	**1.91**	**1.91**	**1.91**	**1.91**	**1.91**	**1.91**
LTDZ	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**
TR1	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**
FDSa	481.33	**126.60**	404.98	193.48	451.71	155.07
FDSb	26.19	26.19	**17.97**	**17.97**	25.86	25.86
FDSc	28.31	28.31	**19.69**	**19.69**	27.48	27.48
FDSd	29.19	29.19	**19.92**	**19.92**	28.80	28.80

Note that the number of function evaluations for all mentioned methods beside H-BFGS (for H-BFGS is equal to the number of iterations.) are zero

For each test problem the numbers with the least amount of Iter and CIter are in bold

$$\begin{aligned} &\min\, h(t,d) = t \\ &s.t \begin{cases} \nabla F_j(x)^T d + \frac{1}{2} d^T B^j(x) d \leq t, \quad (1 \leq j \leq m), \\ x_L - x \leq d \leq x_U - x, \\ (t,d) \in \mathbb{R} \times \mathbb{R}^n. \end{cases} \end{aligned} \tag{47}$$

In this paper, we have used the TOMLAB package for solving Subproblem (47).

In this section, all QNMO methods mentioned in Sect. 3 are compared from two points of view: (i) CPU times and the computational effort; (ii) The quality of the approximated nondominated frontiers. Tables 2, 3, 4 and Figs. 2, 3, 4 and 5 are devoted to compare our 6 methods from the first point of view and Figs. 1 and 6, 7, 8, 9, 10 are dedicated to compare all methods from the second point of view. In Tables 2, 3 and 4, corresponding to all test problems, the algorithms with the best performance

Table 4 Comparing the average number of iterations (Iter) and function evaluations (Feval) for BFGSLS, SS-BFGSLS, and H-BFGSLS QNMO methods in solving some test problems only for initial points that are converged during less than 500 iterations

Name	BFGSLS		SS-BFGSLS		H-BFGSLS	
	Iter	Feval	Iter	Feval	Iter	Feval
Hil	4.76	11.10	4.66	10.78	**2.25**	**4.06**
DD1a	6.34	6.53	6.33	**6.33**	**6.29**	6.38
DD1b	8.32	8.32	**8.10**	8.69	8.28	**8.28**
DD1c	13.31	35.59	15.48	46.38	**7.71**	**8.15**
DD1d	3.76	3.76	7.53	15.82	**3.38**	**3.39**
KW2	2.5051	3.3838	2.4137	3.25	**2.37**	**3.15**
JOS1a	18.61	38.32	**13.95**	**13.95**	14.25	15.18
JOS1b	14.01	14.02	**13.89**	**13.89**	14.04	14.59
JOS1c	23.27	73.93	**13.95**	**13.95**	15.34	16.56
JOS1d	21.47	56.49	**13.95**	**13.95**	17.58	18.41
JOS1e	22.71	62.35	22.57	81.49	**16.50**	**17.88**
JOS1f	31.34	129.00	22.18	59.93	**20.00**	**23.89**
JOS1g	23.59	26.86	**19.06**	**19.06**	21.04	25.68
JOS1h	27.56	56.67	45.78	274.48	**21.41**	**23.80**
PNR	6.10	10.02	6.09	10.07	**2.09**	**2.96**
ZDT6a	**1.92**	**1.92**	**1.92**	**1.92**	**1.92**	**1.92**
ZDT6b	**1.91**	**1.91**	**1.91**	**1.91**	**1.91**	**1.91**
LTDZ	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**
TR1	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**
FDSa	121.00	373.86	185.59	212.31	**85.74**	**98.42**
FDSb	51.11	277.71	38.45	225.23	**25.07**	**30.01**
FDSc	52.93	279.58	76.24	599.3	**26.6**	**30.55**
FDSd	80.63	549.57	77.78	616.04	**27.63**	**32.33**

For each test problem the numbers with the least amount of Iter and CIter are in bold

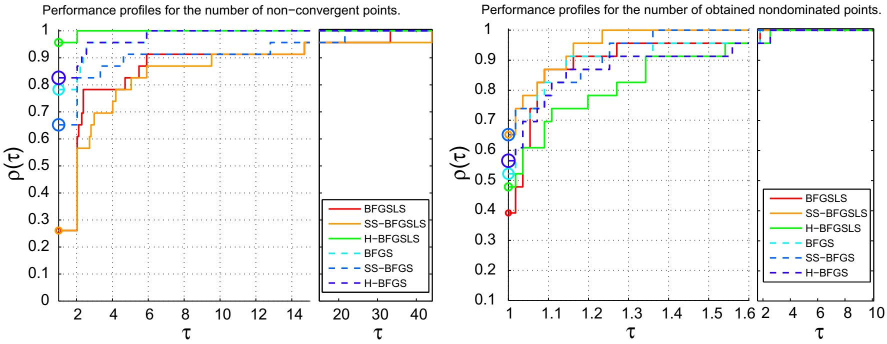


Fig. 1 The left figure depicts the performance profiles of the average number of non-convergent points for QNMO methods and the right figure depicts the performance profiles of the average number of nondominated points for QNMO methods

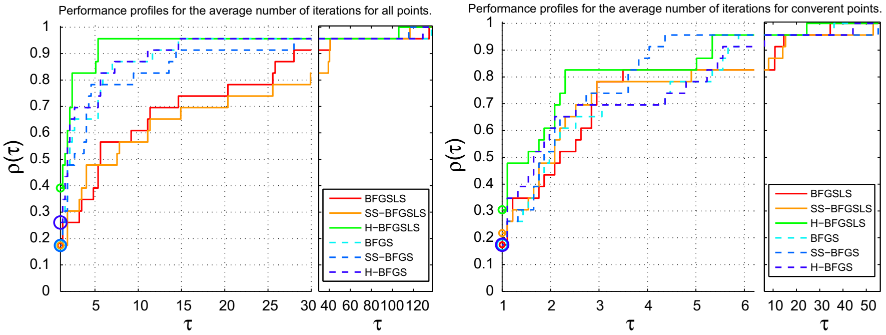


Fig. 2 Comparison of the performance profiles of the average number of iterations for QNMO methods. The left and right figures depict the performance profiles of the average number of iterations for all points and convergent points, respectively

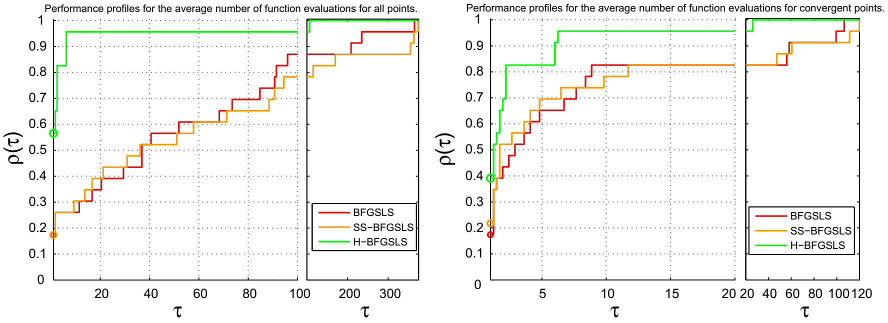


Fig. 3 Comparison of the performance profiles of the average number of function evaluations for all QNMO methods. The left and right figures depict the performance profiles of the average number of function evaluations for all points and convergent points, respectively

are specified. The approximated nondominated frontiers of some test problems that obtained by the proposed QNMO Methods are compared in Figs. 7, 8, 9 and 10.

As seen in Figs. 1, 2, 3 and 4, the variations of the profiles in the small values of τ are more than their variations in the large values of τ. Moreover, from a large values of τ onwards, the profiles nearly turn into a horizontal line. Thus, in order to have a better comparison, we zoomed in on the smaller values of τ. As mentioned in Sect. 5.1, the value of $\rho_s(1)$ in all profiles is more important than the other values. Thus, in order to have a more convenient comparison among the profiles, in this article the point is specified by a tiny circle that has the same color as the color of the related profile.

Tow criteria that may be of interest in comparing QNMO methods are the number of non-convergent points and the number of obtained nondominated points. All proposed QNMO methods are compared in these criteria in Fig. 1. According to this figure, in the number of the nondominated points, the methods SS-BFGSLS, SS-BFGS, H-BFGSLS, BFGS, H-BFGS, and BFGSLS have the best performance in a decreasing order. As seen in Fig. 1, the H-BFGSLS method has a considerably better performance than the other methods in the number of non-convergent points such that the value of

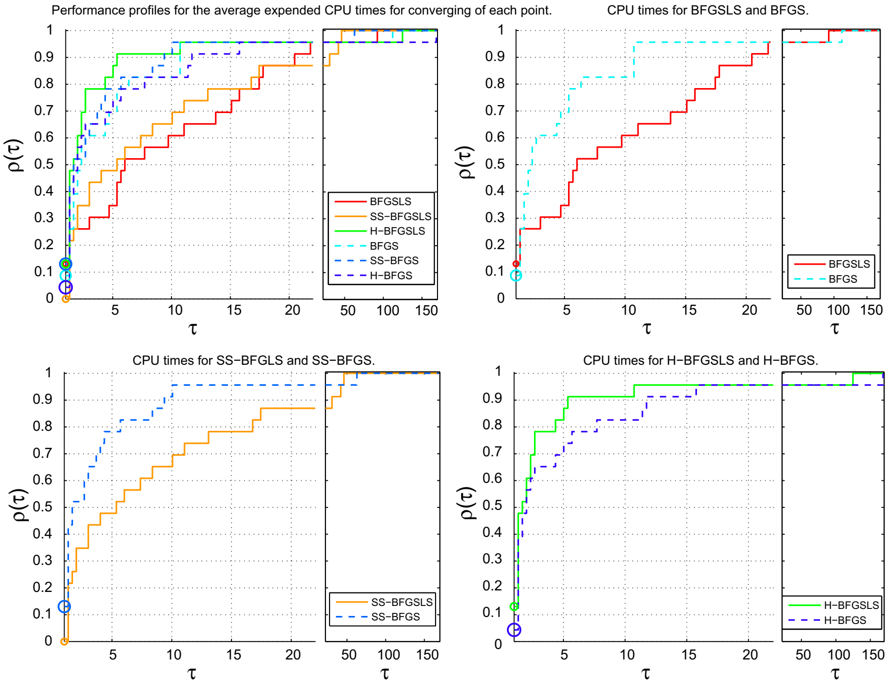


Fig. 4 Comparison of the performance profiles of the average expended CPU times for QNMO methods. For a better comparison, each of these methods, with line search and without any line search, are depicted in individual figures

$\rho(\tau)$ for the H-BFGSLS method is almost equal to one for all values of τ. Moreover, the worst performance in this regard is due to the SS-BFGSLS and BFGSLS methods.

Figures 2, 3, 4 and 5 and Tables 2, 3, 4 compare the methods in three aspects: the speed of convergence, the number of iterations, and the number of function evaluations. Looking at Figs. 2, 3, 4 and 5 and Tables 2, 3, 4, it is evident that the H-BFGSLS method has considerably a better performance than the other methods. The worst performance in this regard is due to the BFGSLS method. In the three points of view mentioned above, Figures 4 and 5 show that, besides the H-BFGSLS method, the other methods in the absence of line search have a better performance than when Armijo line search is used.

Figure 6 compares the methods in the purity metric, spread metric, and epsilon indicator. These factors are used to compare the quality of the obtained nondominated frontiers. Based on these factors, the best performance is due to the SS-BFGS and SS-BFGSLS methods, and the worst performance is due to the H-BFGSLS, H-BFGS, BFGSLS, and BFGS methods in a decreasing order.

In order to have a good comparison, the obtained nondominated frontiers related to some of the test problems presented in Table 1 are depicted in Figs. 7, 8, 9 and 10. Also, in Figs. 7, 8 and 9, the obtained nondominated frontiers are compared to the reference frontier.

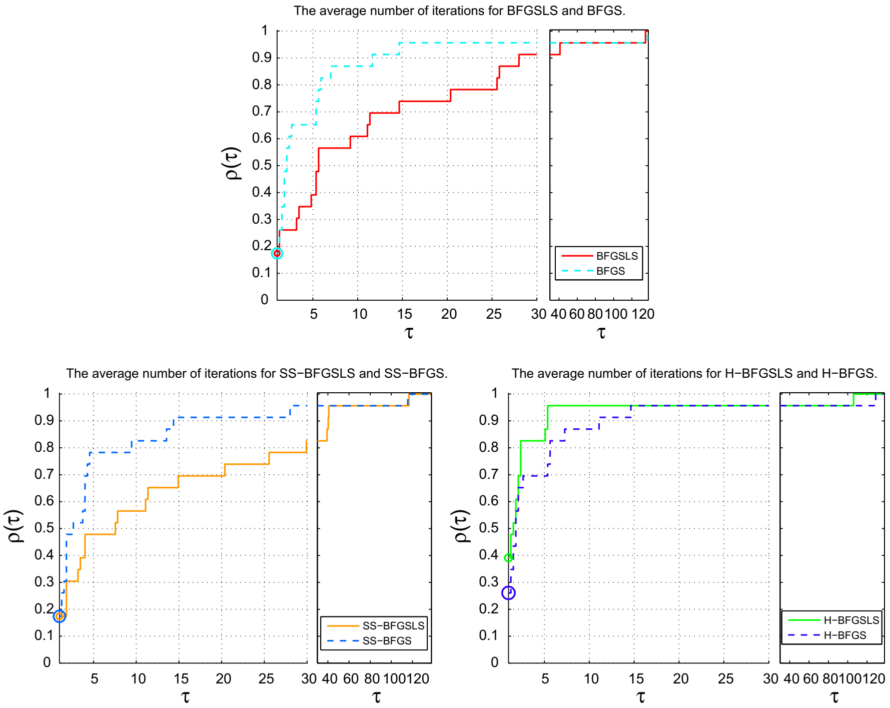


Fig. 5 Performance profiles of the average number of iterations for QNMO methods with line search and without any line search

7 Discussion and conclusion

In this paper, a new QNMO method in the self-scaling Broyden class and a new QNMO method in the Huang class were presented. Based on the observed results in Sect. 6, these methods have some advantages compared to the methods in the Broyden class. In order to have a good comparison, all methods are compared both in the presence of the Armijo line search rule and without any line search.

A comprehensive comparative study of QNMO methods has been presented. All methods are compared from two general points of view: (1) The computational effort and the speed of convergence (2) The quality of the obtained nondominated frontiers. The obtained numerical results show that from the first point of view, the best methods are the H-BFGSLS method followed by the SS-BFGS method. The worst performance in this regard is due to the BFSGLS method. In the second case, the factors of the number of the obtained nondominated points, the purity metric, spread metric, and epsilon indicator are used. From the second point of view, the SS-BFGSLS method followed by the SS-BFGS method have the best performance. Although the SS-BFGSLS method has a better performance in obtaining the nondominated frontier than the SS-BFGS method, it is considerably weaker than the SS-BFGS method regarding the speed of the convergence. Moreover, although the H-BFGSLS method has a weaker performance in terms of the quality of the obtained nondominated

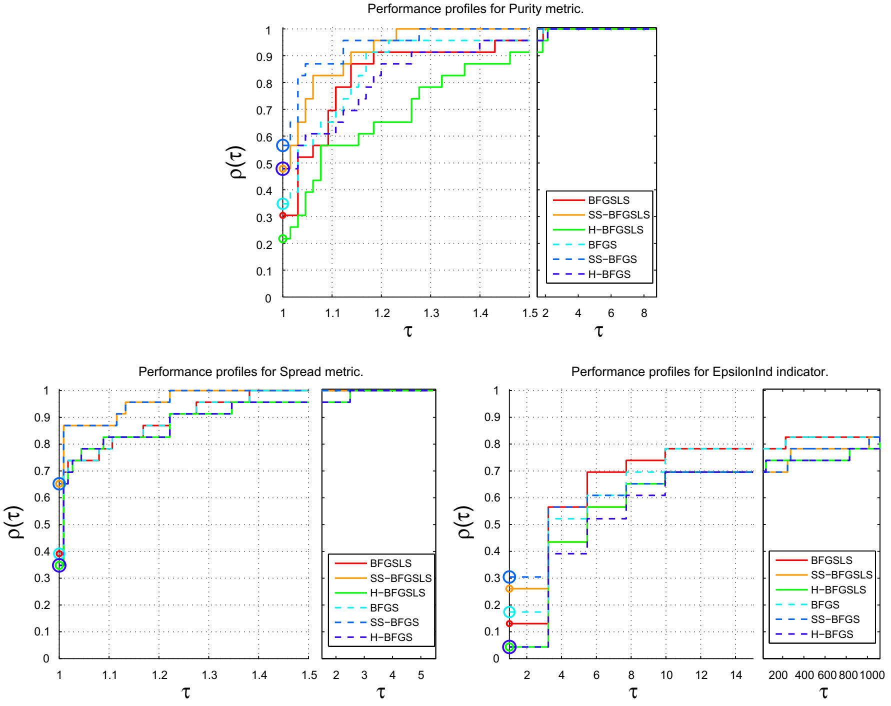


Fig. 6 Comparison of the performance profiles of the purity metric, spread metric, and epsilon indicator for QNMO methods

frontier than the other methods, it has the best performance in the speed of convergence.

The methods of the Broyden and the self-scaling Broyden classes, where no line search was used (Algorithm 2), have a higher speed of convergence compared to the case where the Armijo line search is used. Lack of any line search in these methods reduces the quality of the obtained nondominated frontier. However, this weakness is negligible when compared with the enhancement of the speed of convergence. The increase in the speed of convergence in the case of no line search is the result of the difference in using the Armijo line search for single objective and multiobjective problems. In order to choose the step length in the Armijo rule, we should find the largest number satisfying the Armijo inequality (or inequalities). The number of inequalities in the Armijo rule for single objective problems is only one, while for multiobjective case it equals to the number of objectives of the problem. Thus, in the multiobjective case, the Armijo step length is a number that simultaneously satisfies all of the inequalities. This fact shrinks the step length and, as a result, the speed of convergence is significantly decreased in multiobjective case.

In the methods of the Broyden and the self-scaling Broyden class only gradient information was used for approximating the Hessian matrix, while in the method of the Huang class, in addition to gradient information, function value information

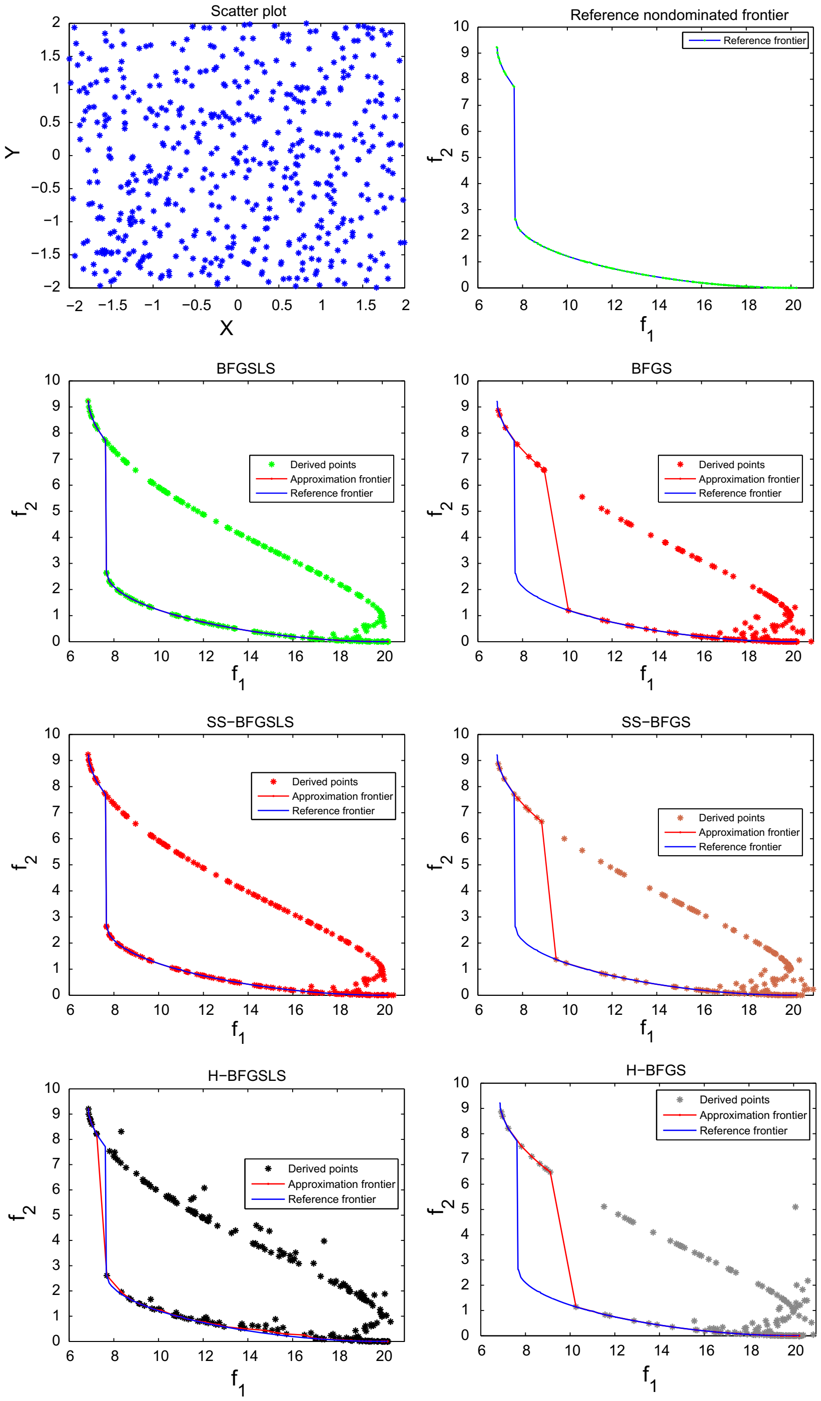


Fig. 7 Comparison between the nondominated frontiers obtained from QNMO methods for Problem PNR

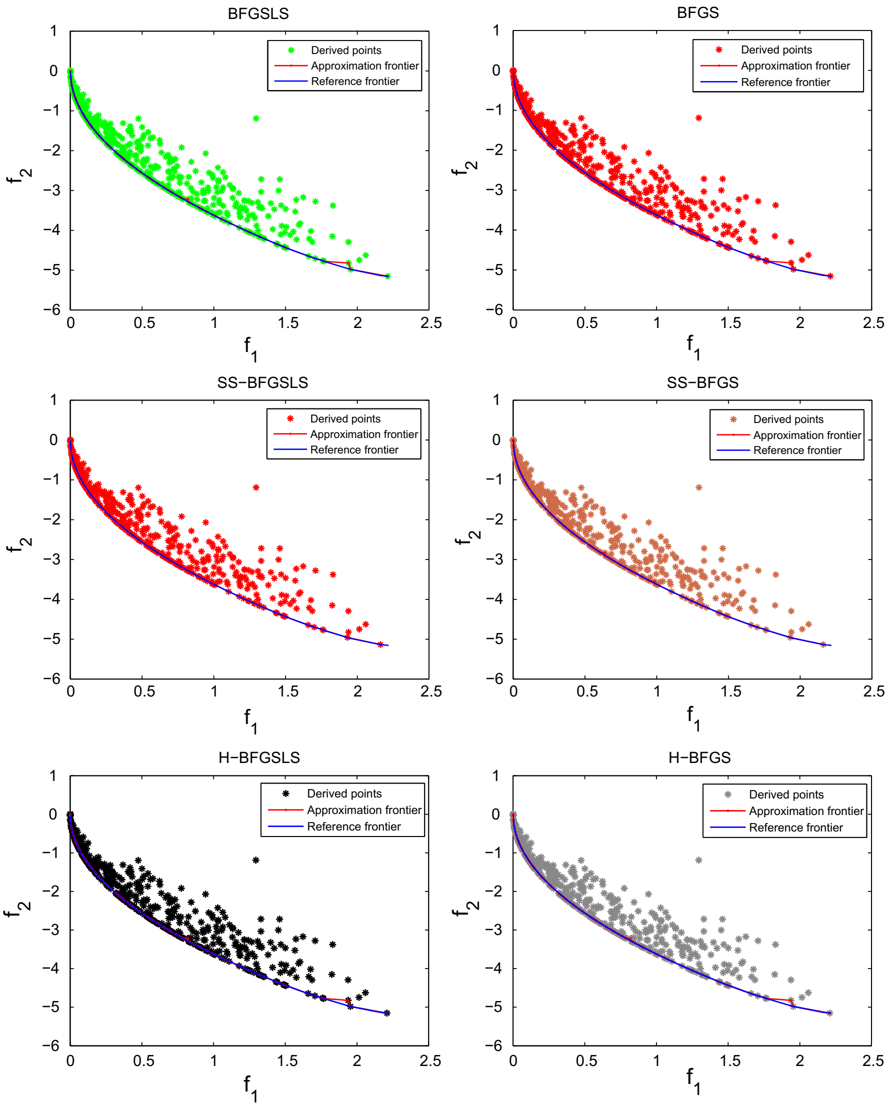


Fig. 8 Comparison between the nondominated frontiers obtained from QNMO methods for Problem DD1a

was also employed. As a result of this fact, according to Theorem 2, the method of the Huang class has a higher accuracy in approximating the Hessian matrix than the methods of the Broyden and the self-scaling Broyden classes. For this reason, the number of function evaluations in the H-BFGSLS method is less than the BFGSLS and SS-BFGSLS methods (see Table 2).

Unlike the methods of the Broyden and the self-scaling Broyden classes, using the Armijo line search for the method of the Huang class has been useful compared to the

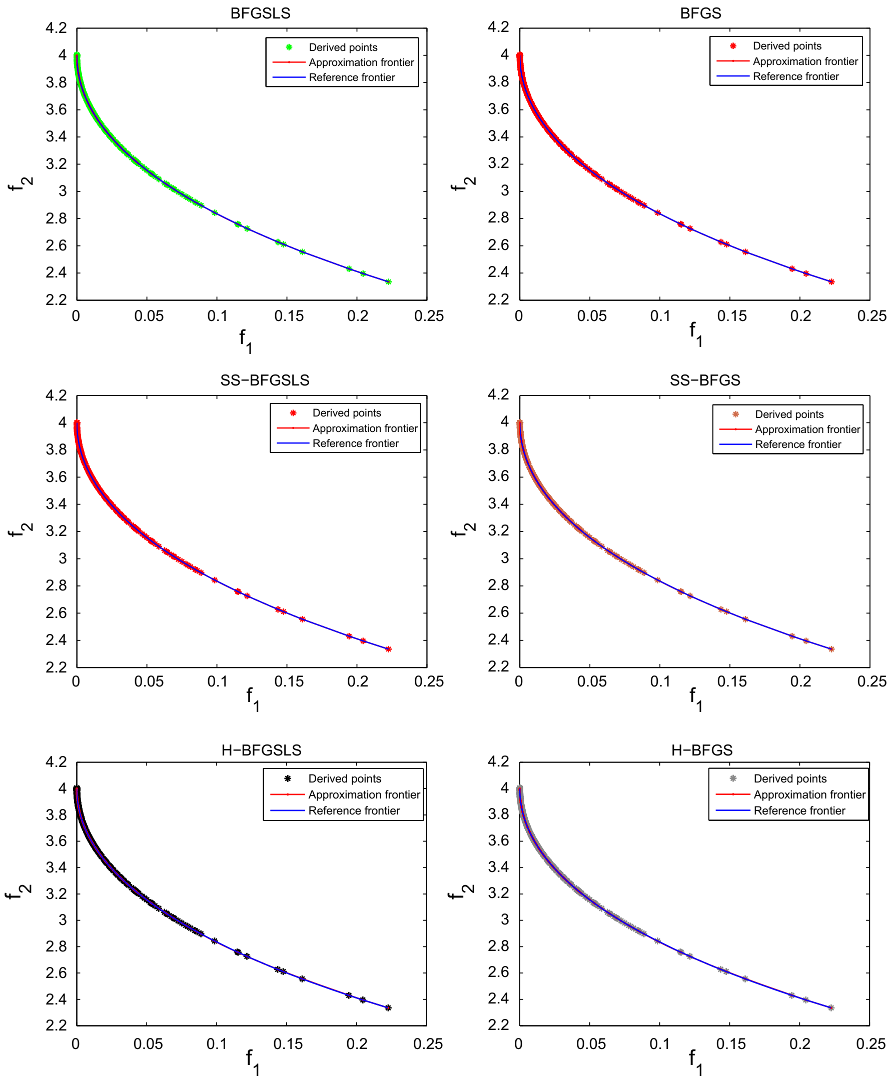
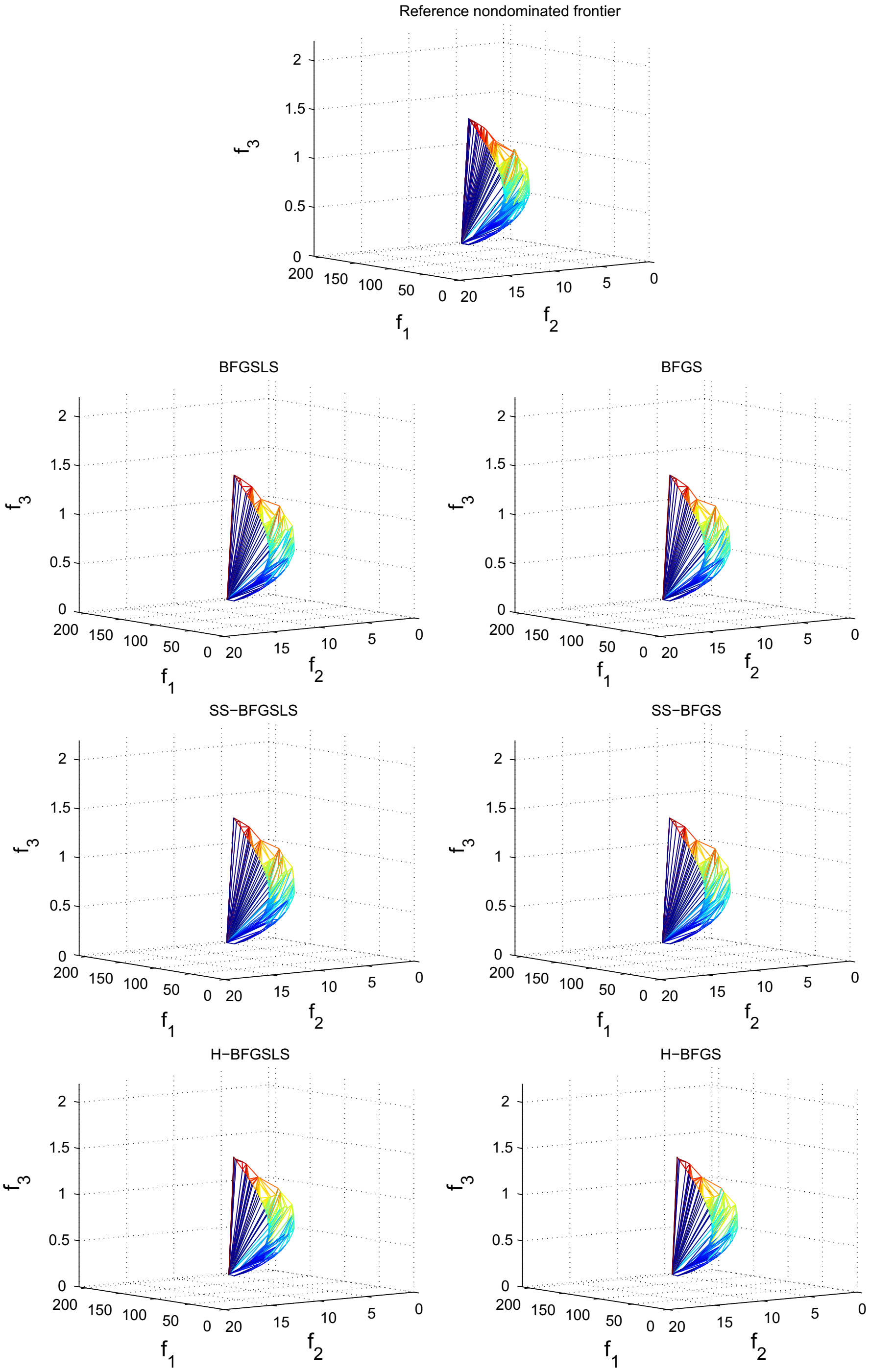


Fig. 9 Comparison between the nondominated frontiers obtained from QNMO methods for Problem JOS1a

case where no line search is used in both aspects of the speed of convergence and the quality of the obtained nondominated frontier.

As a result of the above discussions and the observed results, we can conclude that, generally, the H-BFGSLS and SS-BFGS methods have a better performance than the other methods proposed in this paper.


Reference nondominated frontier

BFGSLS

BFGS

SS−BFGSLS

SS−BFGS

H−BFGSLS

H−BFGS

Fig. 10 Comparison between the nondominated frontiers obtained from QNMO methods for Problem FDS with 5 variable $-2 \leq x_i \leq 2$

References

Bandyopadhyay S, Pal SK, Aruna B (2004) Multiobjective GAs, quantitative indices, and pattern classification. IEEE Trans Syst Man Cybern B Cybern 34(5):2088–2099

Basirzadeh H, Morovati V, Sayadi A (2014) A quick method to calculate the super-efficient point in multiobjective assignment problems. J Math Comput Sci 10:157–162

Basseur M (2006) Design of cooperative algorithms for multi-objective optimization: application to the flow-shop scheduling problem. 4OR 4(3):255–258

Benson H, Sayin S (1997) Towards finding global representations of the efficient set in multiple objective mathematical programming. Nav Res Logist 44(1):47–67

Custodio AL, Madeira JFA, Vaz AIF, Vicente LN (2011) Direct multisearch for multiobjective optimization. SIAM J Optim 21(3):1109–1140

Da Silva CG, Climaco J, Almeida Filho A (2010) The small world of efficient solutions: empirical evidence from the bi-objective 0, 1-knapsack problem. 4OR 8(2):195–211

Das I, Dennis JE (1998) Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems. SIAM J Optim 8(3):631–657

Dolan ED, More JJ (2002) Benchmarking optimization software with performance profiles. Math Program 91:201–213

Drummond LG, Iusem AN (2004) A projected gradient method for vector optimization problems. Comput Optim Appl 28:5–29

Drummond LMG, Svaiter BF (2005) A steepest descent method for vector optimization. J Comput Appl Math 175:395–414

Ehrgott M (2005) Multicriteria optimization. Springer, Berlin

Eskelinen P, Miettinen K (2012) Trade-off analysis approach for interactive nonlinear multiobjective optimization. OR Spectrum 34(4):803–816

Fliege J (2004) Gap-free computation of pareto-points by quadratic scalarizations. Math Methods Oper Res 59(1):69–89

Fliege J (2006) An efficient interior-point method for convex multicriteria optimization problems. Math Oper Res 31(4):825–845

Fliege J, Svaiter BF (2000) Steepest descent methods for multicriteria optimization. Math Methods Oper Res 51:479–494

Fliege J, Drummond LMG, Svaiter BF (2009) Newton's method for multiobjective optimization. SIAM J Optim 20:602–626

Fliege J, Heseler A (2002) Constructing approximations to the efficient set of convex quadratic multiobjective problems. Ergebnisberichte Angewandte Mathematik 211, Univ. Dortmund, Germany

Gopfert A, Nehse R (1990) Vektoroptimierung: theorie, verfahren und anwendungen. B. G. Teubner Verlag, Leipzig

Hillermeier C: Nonlinear Multiobjective Optimization: a generalized homotopy approach. ISNM 25, Berlin (2001)

Jin Y, Olhofer M, Sendhoff B (2001) Dynamic weighted aggregation for evolutionary multiobjective optimization: why does it work and how? In: Proceedings of the genetic and evolutionary computation conference, pp 1042–1049 (2001)

Kim IY, De Weck OL (2005) Adaptive weighted sum method for bi-objective optimization: pareto front generation. Struct Multidiscip Optim 29:149–158

Knowles J, Thiele L, Zitzler E (2006) A tutorial on the performance assessment of stochastic multiobjective optimizers, TIK Report 214. Computer Engineering and Networks Laboratory, ETH Zurich

Kuk H, Tanino T, Tanaka M (1997) Trade-off analysis for vector optimization problems via scalarization. J Inf Optim Sci 18(1):75–87

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

Luc DT (1988) Theory of vector optimization. Springer, Berlin

Morovati V, Pourkarimi L, Basirzadeh H (2016) Barzilai and Borwein's method for multiobjective optimization problems. Numer Algorithms 72(3):539–604

Nocedal J, Wright S (2006) Numerical optimization. Springer, New York

Povalej Z (2014) Quasi-Newton's method for multiobjective optimization. J Comput Appl Math 255:765–777

Preuss M, Naujoks B, Rudolph G (206) Pareto set and EMOA behavior for simple multimodal multiobjective functions. In: Runarsson TP et al (eds) Proceedings of the ninth international conference on parallel problem solving from nature (PPSN IX), Springer, Berlin, pp 513–522 (2006)

Qu S, Goh M, Chan FTS (2011) Quasi-Newton methods for solving multiobjective optimization. Oper Res Lett 39:397–399

Qu S, Goh M, Liang B (2013) Trust region methods for solving multiobjective optimisation. Optim Method Softw 28(4):796–811

Sakawa M, Yano H (1990) Trade-off rates in the hyperplane method for multiobjective optimization problems. Eur J Oper Res 44(1):105–118

Sayadi-Bander A, Morovati V, Basirzadeh H (2015) A super non-dominated point for multi-objective transportation problem. Appl Appl Math 10(1):544–551

Sayadi-bander A, Kasimbeyli R, Pourkarimi L (2017) A coradiant based scalarization to characterize approximate solutions of vector optimization problems with variable ordering structures. Oper Res Lett 45(1):93–97

Sayadi-bander A, Pourkarimi L, Kasimbeyli R, Basirzadeh H (2017) Coradiant sets and ε-efficiency in multiobjective optimization. J Glob Optim 68(3):587–600. https://doi.org/10.1007/s10898-016-0495-4

Schandl B, Klamroth K, Wiecek MM (2001) Norm-based approximation in bicriteria programming. Comput Optim Appl 20(1):23–42

Segura C, Coello CAC, Miranda G, Len C (2013) Using multi-objective evolutionary algorithms for single-objective optimization. 4OR 11(3):201–228

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

Tappeta RV, Renaud JE (1999) Interactive multiobjective optimization procedure. AIAA J 37(7):881–889

Villacorta KDV, Oliveira PR, Soubeyran A (2014) A trust-region method for unconstrained multiobjective problems with applications in satisficing processes. J Optim Theory Appl 160:865–889

Zhang J, Xu C (2001) Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations. J Comput Appl Math 137(2):269–278

Zhang JZ, Deng NY, Chen LH (1999) New quasi-Newton equation and related methods for unconstrained optimization. J Optim Theory Appl 102(1):147–167

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

Zitzler E, Thiele L, Laumanns M, Fonseca CM, da Fonseca VG (2003) Performance assessment of multi-objective optimizers: an analysis and review. IEEE Trans Evol Comput 7(2):117–132