

# A quasi-Newton method with Wolfe line searches for multiobjective optimization

L. F. Prudente\*      D. R. Souza\*

March 18, 2024

**Abstract:** We propose a BFGS method with Wolfe line searches for unconstrained multiobjective optimization problems. The algorithm is well defined even for general nonconvex problems. Global convergence and R-linear convergence to a Pareto optimal point are established for strongly convex problems. In the local convergence analysis, if the objective functions are locally strongly convex with Lipschitz continuous Hessians, the rate of convergence is Q-superlinear. In this respect, our method exactly mimics the classical BFGS method for single-criterion optimization.

**Keywords:** multiobjective optimization, Pareto optimality, quasi-Newton methods, BFGS method, Wolfe line search

**AMS subject classifications:** 49M15, 65K05, 90C29, 90C30

## 1 Introduction

In multiobjective optimization, we seek to minimize two or more objective functions simultaneously. Usually, in a problem of this class, there is no single point that minimizes all the functions at once. In that case, the objectives are said to be conflicting and we use the concept of *Pareto optimality* to characterize a solution of the problem. A point is called *Pareto optimal* if none of the objective functions can be improved without degrading another. Multiobjective optimization problems appear in many fields of science. We refer the reader to [3, 56] and references therein for some interesting practical applications.

Over the past two decades, several iterative methods for scalar-value optimization have been extended and analyzed for multiobjective optimization. This line of research was opened in [25] with the extension of the steepest descent method; see also [24]. Other algorithms include Newton [10, 24, 30, 59], quasi-Newton [1, 36, 42, 47, 49, 52, 53], conjugate gradient [28, 40], projected gradient [2, 22, 26, 27, 29], and proximal methods [5, 7–9, 11]. As a common feature, these methods enjoy convergence properties and do not transform the problem at hand into a parameterized scalar problem and then solve it, being attractive alternatives for scalarization [21] and heuristic approaches [37].

Quasi-Newton algorithms are one of the most popular classes of methods for solving an unconstrained single-objective optimization problem. Its long history began in 1959 with the work

---

\*Instituto de Matemática e Estatística, Universidade Federal de Goiás, CEP 74001-970, Goiânia, GO, Brazil, E-mails: [lfprudente@ufg.br](mailto:lfprudente@ufg.br), [danilo\\_souza@discente.ufg.br](mailto:danilo_souza@discente.ufg.br). This work was funded by FAPEG (Grants PPP03/15-201810267001725) and CNPq (Grants 424860/2018-0, 309628/2020-2).

of Davidon [16] and was popularized four years later by Fletcher and Powell [23]. Since then, quasi-Newton algorithms have attracted the attention of the scientific community primarily because they avoid computations of second derivatives and perform well in practice. Papers on the subject are too many to list, including contributions from the most prominent names in the optimization community. In a quasi-Newton method, the search direction is computed based on a quadratic model of the objective function, where the true Hessian is replaced by some approximation which is updated after each iteration. The most effective quasi-Newton update scheme is the BFGS formula, which was independently discovered by Broyden, Fletcher, Goldfarb, and Shanno in 1970. Under some proper assumptions, the BFGS algorithm with an adequate line search is globally and superlinearly convergent on strongly convex problems [6, 50]. On the other hand, it may not converge for general nonconvex functions [13, 14, 43].

The BFGS method for multiobjective optimization was first proposed in [49] and later also studied in [36, 42, 47, 52, 53]. Similarly to the single-criterion case, the search direction is defined as the solution of a problem involving quadratic models of the objective functions, where BFGS updates are used in place of the true Hessians. Despite the aforementioned papers, the convergence theory of the BFGS method for multiobjective optimization problems can still be considered incomplete. In the following we highlight some shortcomings of the existing references: (i) the algorithms are usually designed for strongly convex problems; (ii) the BFGS updates and their inverses are often assumed to be uniformly bounded (which seems unrealistic, see [48, Section 6.4]); (iii) some key intermediate steps in convergence analysis are often stated without proof. The aim of the present paper is to overcome all these drawbacks.

We propose a BFGS algorithm for unconstrained multiobjective optimization problems, where the step sizes satisfy the standard (vector) Wolfe conditions recently introduced in [40]. The Wolfe conditions are a key tool to preserve the positive definiteness of the Hessian approximations. As a consequence, our algorithm is well defined even for general nonconvex problems. Global and R-linear convergence to a Pareto optimal point are established for strongly convex problems. In the local convergence analysis, assuming that the objective functions are locally strongly convex with Lipschitz continuous Hessians, we prove that the rate of convergence is Q-superlinear. As an intermediate result, we show that the unit step size eventually satisfies the Wolfe conditions. Furthermore, a Dennis–Moré-type condition for multiobjective optimization problems will appear very clearly. We emphasize that all assumptions considered are natural extensions of those made for the scalar optimization case. Numerical experiments on convex and nonconvex multiobjective problems illustrating the potential practical advantages of our approach are presented.

The outline of this paper is as follows. Section 2 presents some basic concepts and results of multiobjective optimization. In section 3, we describe our algorithm and show that it is well defined even on general nonconvex problems. Global and local convergence results are discussed in sections 4 and 5, respectively. Numerical experiments are presented in section 6 and final remarks are made in section 7.

## 2 Preliminaries

**Notation.**  $\mathbb{R}$  and  $\mathbb{R}_{++}$  denote the set of real numbers and the set of positive real numbers, respectively. As usual,  $\mathbb{R}^n$  and  $\mathbb{R}^{n \times p}$  denote the set of  $n$ -dimensional real column vectors and the set of  $n \times p$  real matrices, respectively. The identity matrix of size  $n$  is denoted by  $I_n$ . If  $u, v \in \mathbb{R}^n$ , then  $u \leq v$  is to be understood in a componentwise sense, i.e.,  $u_i \leq v_i$  for all  $i = 1, \dots, n$ . For  $A \in \mathbb{R}^{n \times n}$ ,  $A \succ 0$  (resp.  $A \prec 0$ ) means that  $A$  is positive (resp. negative) definite.  $\|\cdot\|$  is the Euclidean norm. The cardinality of a set  $C$  is denoted by  $|C|$ . The ceiling and

floor functions are denoted by  $\lceil \cdot \rceil$  and  $\lfloor \cdot \rfloor$ , respectively; i.e., if  $x \in \mathbb{R}$ , then  $\lceil x \rceil$  is the least integer greater than or equal to  $x$  and  $\lfloor x \rfloor$  is the greatest integer less than or equal to  $x$ . Given two real sequences  $\{a_k\}$  and  $\{b_k\}$  (with  $b_k > 0$  for all  $k$ ), we write  $a_k = o(b_k)$  if  $\lim_{k \rightarrow \infty} a_k/b_k = 0$ . If  $K = \{k_1, k_2, \dots\} \subseteq \mathbb{N}$ , with  $k_j < k_{j+1}$  for all  $j \in \mathbb{N}$ , then we denote  $K \subset \mathbb{N}$ .

Given  $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$  a continuously differentiable function, we are interested in finding a *Pareto optimal* point of  $F$ . We denote this problem as

$$\min_{x \in \mathbb{R}^n} F(x). \quad (1)$$

A point  $x^* \in \mathbb{R}^n$  is *Pareto optimal* (resp. *weak Pareto optimal*) if there exists no other  $x \in \mathbb{R}^n$  with  $F(x) \leq F(x^*)$  and  $F(x) \neq F(x^*)$  (resp.  $F(x) < F(x^*)$ ). These concepts are also defined locally: We say that  $x^* \in \mathbb{R}^n$  is a *local Pareto optimal* (resp. *local weak Pareto optimal*) if there exists a neighborhood  $U \subset \mathbb{R}^n$  of  $x^*$  such that  $x^*$  is Pareto optimal (resp. weak Pareto optimal) for  $F$  restricted to  $U$ . A necessary (but in general not sufficient) condition for local weak Pareto-optimality of  $x^*$  is

$$-(\mathbb{R}_{++}^m) \cap \text{Image}(JF(x^*)) = \emptyset, \quad (2)$$

where  $JF(x^*)$  denotes the image set of the Jacobian of  $F$  at  $x^*$ . A point  $x^*$  that satisfies (2) is called a *Pareto critical* point. Note that, if  $x \in \mathbb{R}^n$  is not Pareto critical, then there exists a direction  $d \in \mathbb{R}^n$  such that  $\nabla F_j(x)^T d < 0$  for all  $j = 1, \dots, m$ . This implies that  $d$  is a *descent direction* for  $F$  at  $x$ , i.e., there exists  $\varepsilon > 0$  such that  $F(x + \alpha d) < F(x)$  for all  $\alpha \in ]0, \varepsilon]$ . We say that  $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$  is convex (resp. strongly convex) if its components  $F_j: \mathbb{R}^n \rightarrow \mathbb{R}$  are convex (resp. strongly convex), for all  $j = 1, \dots, m$ . The next result relates the concepts of criticality, optimality, and convexity.

**Lemma 2.1.** [24, Theorem 3.1] *The following statements hold:*

- (i) *if  $x^*$  is local weak Pareto optimal, then  $x^*$  is a critical point for  $F$ ;*
- (ii) *if  $F$  is convex and  $x^*$  is critical for  $F$ , then  $x^*$  is weak Pareto optimal;*
- (iii) *if  $F$  is twice continuously differentiable,  $\nabla^2 F_j(x) > 0$  for all  $j \in \{1, \dots, m\}$  and all  $x \in \mathbb{R}^n$ , and if  $x^*$  is critical for  $F$ , then  $x^*$  is Pareto optimal.*

Define  $\mathcal{D}: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  by

$$\mathcal{D}(x, d) := \max_{j=1, \dots, m} \nabla F_j(x)^T d.$$

The function  $\mathcal{D}$  characterizes the descent direction for  $F$  at  $x$ . Indeed, it is easy to see that if  $\mathcal{D}(x, d) < 0$ , then  $d$  is a descent direction for  $F$  at  $x$ . Moreover,  $x$  is a Pareto critical point if and only if  $\mathcal{D}(x, d) \geq 0$  for all  $d \in \mathbb{R}^n$ . Next we present other useful properties of function  $\mathcal{D}$  that can be trivially obtained from its definition.

**Lemma 2.2.** *The following statements hold:*

- (i) *for any  $x \in \mathbb{R}^n$  and  $\alpha \geq 0$ , we have  $\mathcal{D}(x, \alpha d) = \alpha \mathcal{D}(x, d)$ ;*
- (ii) *the mapping  $(x, d) \mapsto \mathcal{D}(x, d)$  is continuous.*

The quasi-Newton methods for solving (1) belong to a class of algorithms in which the search direction  $d(x)$  from a given  $x \in \mathbb{R}^n$  is defined as the solution of

$$\min_{d \in \mathbb{R}^n} \max_{j=1, \dots, m} \nabla F_j(x)^T d + \frac{1}{2} d^T B_j d, \quad (3)$$

where  $B_j \in \mathbb{R}^{n \times n}$  is some approximation of  $\nabla^2 F_j(x)$  for all  $j = 1, \dots, m$ , see [49]. If  $B_j \succ 0$  for all  $j = 1, \dots, m$ , then the objective function is strongly convex and hence (3) has a unique solution. We will denote the optimal value of problem (3) by  $\theta(x)$ , i.e.,

$$d(x) := \arg \min_{d \in \mathbb{R}^n} \max_{j=1, \dots, m} \nabla F_j(x)^T d + \frac{1}{2} d^T B_j d, \quad (4)$$

and

$$\theta(x) := \max_{j=1, \dots, m} \nabla F_j(x)^T d(x) + \frac{1}{2} d(x)^T B_j d(x). \quad (5)$$

In the particular case where  $B_j = I_n$  for all  $j = 1, \dots, m$ ,  $d(x)$  corresponds to the steepest descent direction, see [25]. In turn, if  $B_j = \nabla^2 F_j(x)$  for all  $j = 1, \dots, m$ ,  $d(x)$  turns out to be the Newton direction, see [24].

In what follows, we assume that  $B_j \succ 0$  for all  $j = 1, \dots, m$ . In this case, (3) is equivalent to the following convex quadratic optimization problem:

$$\begin{aligned} \min_{(t, d) \in \mathbb{R} \times \mathbb{R}^n} \quad & t \\ \text{s. t.} \quad & \nabla F_j(x)^T d + \frac{1}{2} d^T B_j d \leq t, \quad \forall j = 1, \dots, m. \end{aligned} \quad (6)$$

The unique solution of (6) is given by  $(t, d) := (\theta(x), d(x))$ . Since (6) is convex and has a Slater point (e.g.,  $(1, 0) \in \mathbb{R} \times \mathbb{R}^n$ ), there exists a multiplier  $\lambda(x) \in \mathbb{R}^m$  such that the triple  $(t, d, \lambda) := (\theta(x), d(x), \lambda(x)) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m$  satisfies the Karush-Kuhn-Tucker system:

$$\sum_{j=1}^m \lambda_j = 1, \quad \sum_{j=1}^m \lambda_j [\nabla F_j(x) + B_j d] = 0,$$

and

$$\lambda_j \geq 0, \quad \nabla F_j(x)^T d + \frac{1}{2} d^T B_j d \leq t, \quad \lambda_j \left[ \nabla F_j(x)^T d + \frac{1}{2} d^T B_j d - t \right] = 0,$$

for all  $j = 1, \dots, m$ . Therefore, some manipulations yield

$$d(x) = - \left[ \sum_{j=1}^m \lambda_j(x) B_j \right]^{-1} \sum_{j=1}^m \lambda_j(x) \nabla F_j(x), \quad (7)$$

$$\sum_{j=1}^m \lambda_j(x) = 1, \quad \lambda_j(x) \geq 0, \quad \forall j = 1, \dots, m, \quad (8)$$

and

$$\theta(x) = -\frac{1}{2} d(x)^T \left[ \sum_{j=1}^m \lambda_j(x) B_j \right] d(x). \quad (9)$$

The following lemma shows that direction  $d(x)$  and the optimum value  $\theta(x)$  can be used to characterize Pareto critical points of problem (1).

**Lemma 2.3.** Let  $d : \mathbb{R}^n \rightarrow \mathbb{R}^n$  and  $\theta : \mathbb{R}^n \rightarrow \mathbb{R}$  given by (4) and (5), respectively. Assume that  $B_j \succ 0$  for all  $j = 1, \dots, m$ . Then, we have:

- (i)  $x$  is Pareto critical if and only if  $d(x) = 0$  and  $\theta(x) = 0$ ;
- (ii) if  $x$  is not Pareto critical, then  $d(x) \neq 0$  and  $\mathcal{D}(x, d(x)) < \theta(x) < 0$  (in particular,  $d(x)$  is a descent direction for  $F$  at  $x$ ).

**Proof.** See [24, Lemma 3.2] and [49, Lemma 2]. □

As mentioned, if  $B_j = I_n$  for all  $j = 1, \dots, m$ , the solution of (3) corresponds to the steepest descent direction, which will be denoted by  $d_{SD}(x)$ , i.e.,

$$d_{SD}(x) := \arg \min_{d \in \mathbb{R}^n} \max_{j=1, \dots, m} \nabla F_j(x)^T d + \frac{1}{2} \|d\|^2. \quad (10)$$

Taking into account the previous discussion, it follows that there exists  $\lambda^{SD}(x) \in \mathbb{R}^m$  such that

$$\sum_{j=1}^m \lambda_j^{SD}(x) = 1, \quad \lambda_j^{SD}(x) \geq 0, \quad \forall j = 1, \dots, m, \quad (11)$$

and

$$d_{SD}(x) = - \sum_{j=1}^m \lambda_j^{SD}(x) \nabla F_j(x). \quad (12)$$

In the following, we revise some useful properties related to  $d_{SD}(\cdot)$ .

**Lemma 2.4.** Let  $d_{SD} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  be given by (10). Then:

- (i)  $x$  is Pareto critical if and only if  $d_{SD}(x) = 0$ ;
- (ii) if  $x$  is not Pareto critical, then we have  $d_{SD}(x) \neq 0$  and  $\mathcal{D}(x, d_{SD}(x)) < -(1/2) \|d_{SD}(x)\|^2 < 0$  (in particular,  $d_{SD}(x)$  is a descent direction for  $F$  at  $x$ );
- (iii) the mapping  $d_{SD}(\cdot)$  is continuous;
- (iv) for any  $x \in \mathbb{R}^n$ ,  $-d_{SD}(x)$  is the minimal norm element of the set

$$\left\{ u \in \mathbb{R}^n \mid u = \sum_{j=1}^m \lambda_j \nabla F_j(x), \sum_{j=1}^m \lambda_j = 1, \lambda_j \geq 0 \text{ for all } j = 1, \dots, m \right\},$$

i.e., in the convex hull of  $\{\nabla F_1(x), \dots, \nabla F_m(x)\}$ .

**Proof.** For itens (i), (ii), and (iii), see [31, Lemma 3.3]. For item (iv), see [57, Corollary 2.3]. □

### 3 Algorithm

This section provides a detailed description of the main algorithm. At each iteration  $k$  and for each  $j = 1, \dots, m$ , the true Hessian  $\nabla^2 F_j(x^k)$  is approximated by a matrix  $B_j^k$  using a BFGS-type updating, see (15) below. The algorithm uses a line search procedure satisfying the (vector) Wolfe conditions. As we will see, this will be essential to obtain positive definite updates  $B_j^k$ 's for nonconvex problems.

---

**Algorithm 1. A BFGS algorithm with Wolfe line searches**

Let  $c_1 \in ]0, 1/2[$ ,  $c_2 \in ]c_1, 1[$ ,  $x^0 \in \mathbb{R}^n$ , and  $B_j^0 \succ 0$  for all  $j = 1, \dots, m$  be given. Initialize  $k \leftarrow 0$ .

**Step 1.** *Compute the search direction*

Compute  $d^k := d(x^k)$  and  $\theta(x^k)$  as in (4) and (5), respectively.

**Step 2.** *Stopping criterion*

If  $\theta(x^k) = 0$ , then STOP.

**Step 3.** *Line search procedure*

Compute a step size  $\alpha_k > 0$  (trying first  $\alpha_k = 1$ ) such that

$$F_j(x^k + \alpha_k d^k) \leq F_j(x^k) + c_1 \alpha_k \mathcal{D}(x^k, d^k), \quad \forall j = 1, \dots, m, \quad (13)$$

$$\mathcal{D}(x^k + \alpha_k d^k, d^k) \geq c_2 \mathcal{D}(x^k, d^k), \quad (14)$$

and set  $x^{k+1} := x^k + \alpha_k d^k$ .

**Step 4.** *Prepare the next iteration*

For each  $j = 1, \dots, m$ , define

$$\begin{aligned} B_j^{k+1} := & B_j^k - \frac{(\rho_j^k)^{-1} B_j^k s_k s_k^T B_j^k}{\left( (\rho_j^k)^{-1} - s_k^T y_j^k \right)^2 + (\rho_j^k)^{-1} s_k^T B_j^k s_k} \\ & + \frac{(s_k^T B_j^k s_k) y_j^k (y_j^k)^T}{\left( (\rho_j^k)^{-1} - s_k^T y_j^k \right)^2 + (\rho_j^k)^{-1} s_k^T B_j^k s_k} \\ & + \left( (\rho_j^k)^{-1} - s_k^T y_j^k \right) \frac{y_j^k s_k^T B_j^k + B_j^k s_k (y_j^k)^T}{\left( (\rho_j^k)^{-1} - s_k^T y_j^k \right)^2 + (\rho_j^k)^{-1} s_k^T B_j^k s_k}, \end{aligned} \quad (15)$$

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

$$\rho_j^k := \begin{cases} 1/s_k^T y_j^k, & \text{if } s_k^T y_j^k > 0 \\ 1/\left( \mathcal{D}(x^{k+1}, s_k) - \nabla F_j(x^k)^T s_k \right), & \text{otherwise.} \end{cases} \quad (16)$$

Set  $k \leftarrow k + 1$  and go to Step 1.

---

**Remark 3.1.** (i) In practice, at Step 1,  $d^k$  and  $\theta(x^k)$  are calculated by solving the scalar convex quadratic optimization problem (6). Thus, there are many algorithms capable of effectively dealing with this subproblem. (ii) At Step 2, from Lemma 2.3(i), Algorithm 1 stops at iteration  $k$  if and only if  $x^k$  is Pareto critical. (iii) At Step 3, we always try the step size  $\alpha_k = 1$  first. This detail turns out to be crucial in obtaining a fast convergence rate. Conditions (13)–(14) correspond to the (vector) standard Wolfe conditions recently introduced in [40]. It is possible to show that if  $d^k$  is a descent direction of  $F$  at  $x^k$  and  $F$  is bounded below along  $d^k$ , then there exist intervals of positive step sizes satisfying these conditions, see [40, Proposition 3.2]. The  $m$  inequalities in (13) stipulate that  $\alpha_k$  must give a *sufficient decrease*, proportional to both  $\alpha_k$  and  $\mathcal{D}(x^k, d^k)$ , in each objective function. In turn, the *curvature condition* (14) consists of a single inequality involving information from all objective functions. Let  $\alpha_t > 0$  be a trial step size that satisfies (13). If  $\alpha_t > 0$  is such that  $\nabla F_j(x^k + \alpha_t d^k)^T d^k$  is *strongly* negative for all  $j = 1, \dots, m$  (and hence also  $\mathcal{D}(x^k + \alpha_t d^k, d^k)$ ), it is a sign that we can simultaneously reduce all objectives by moving further along  $d^k$  and then  $\alpha_t$  is ruled out by condition (14). On the other hand, if there exists  $j \in \{1, \dots, m\}$  for which  $\nabla F_j(x^k + \alpha_t d^k)^T d^k$  is only *slightly* negative or even positive (more precisely, greater than or equal to the right hand side of (14)), we suspect that we cannot expect much more decrease in the objective  $F_j$  along direction  $d^k$ , so it makes sense to end the line search by accepting  $\alpha_t$  as the step size, i.e.,  $\alpha_k := \alpha_t$ . Note that, in the latter case, we have  $\mathcal{D}(x^k + \alpha_t d^k, d^k) \geq \nabla F_j(x^k + \alpha_t d^k)^T d^k \geq c_2 \mathcal{D}(x^k, d^k)$  and therefore condition (14) is satisfied for  $\alpha_t$ . In particular, we observe that (14) rules out unacceptably short step sizes that satisfy (13). We refer the reader to [48, Section 3.1] for a careful discussion about this issue in the scalar optimization approach. (iv) At Step 4, if  $\rho_j^k = 1/s_k^T y_j^k$ , then (15) reduces to the classical BFGS update for function  $F_j$ . This is certainly the case when  $F_j$  is strictly convex. Furthermore, in the scalar optimization case ( $m = 1$ ), the curvature condition (14) implies  $s_k^T y_1^k > 0$  and hence we have  $\rho_1^k = 1/s_k^T y_1^k$  and Algorithm 1 becomes the classical scalar BFGS algorithm with standard Wolfe line searches.

Let us provide a motivation for our choice of  $B_j^{k+1}$  in (15). The classical BFGS updating formula is commonly derived by working with the inverse Hessian approximation. For each  $j = 1, \dots, m$ , denote by  $H_j^k$  the approximation for  $[\nabla^2 F_j(x^k)]^{-1}$ . The updated BFGS approximation  $H_j^{k+1}$  is then naturally defined as

$$H_j^{k+1} := \left(I - \rho_j^k s_k (y_j^k)^T\right) H_j^k \left(I - \rho_j^k y_j^k s_k^T\right) + \rho_j^k s_k s_k^T, \quad (17)$$

with  $\rho_j^k$  given by (16); see, for example, [48]. Now, by taking the inverse of  $H_j^{k+1}$  in (17) (which can be easily done using the Sherman-Morrison formula), we obtain the update formula for  $B_j^{k+1}$  in (15).

The next theorem shows that Algorithm 1 is well defined without imposing any convexity assumptions on the objectives. Its proof basically consists in showing that  $B_j^{k+1}$  will be positive definite whenever  $B_j^k$  is positive definite, for each  $j = 1, \dots, m$ . From now on, we denote by  $\mathcal{L}(x^0)$  the level set  $\{x \in \mathbb{R}^n \mid F(x) \leq F(x^0)\}$ .

**Theorem 3.1.** *Suppose that  $F$  is bounded below in  $\mathcal{L}(x^0)$ . Then, Algorithm 1 is well-defined.*

**Proof.** The proof is by induction. Assume that  $B_j^k$  is positive definite for all  $j = 1, \dots, m$  (which trivially holds for  $k = 0$ ). Therefore, the subproblem in Step 1 is solvable. If  $x^k$  is Pareto critical then, by Lemma 2.3(i), Algorithm 1 stops at Step 2. Thus, let us assume that  $x^k$  is not Pareto

critical. In this case, Lemma 2.3(ii) implies that  $d^k$  is a descent direction of  $F$  at  $x^k$ . Since  $F$  is bounded below in  $\mathcal{L}(x^0)$ , by [40, Proposition 3.2], there exist intervals of positive step sizes satisfying conditions (13)–(14) in Step 3 and hence  $x^{k+1}$  can be properly defined. We now show that  $B_j^{k+1}$  in (15) remain definite positive for all  $j = 1, \dots, m$ . This will be done by showing that  $H_j^{k+1}$  in (17) is definite positive for all  $j = 1, \dots, m$ . Since  $s_k = \alpha_k d^k$ , by the definition of  $\mathcal{D}$ , Lemma 2.2(i), and (14), we have

$$\begin{aligned} \mathcal{D}(x^{k+1}, s_k) - \nabla F_j(x^k)^T s_k &\geq \alpha_k [\mathcal{D}(x^{k+1}, d^k) - \mathcal{D}(x^k, d^k)] \\ &\geq -\alpha_k (1 - c_2) \mathcal{D}(x^k, d^k) > 0, \end{aligned}$$

and hence  $\rho_j^k$  defined in (16) is positive for all  $j = 1, \dots, m$ . Let  $j \in \{1, \dots, m\}$  and  $0 \neq z \in \mathbb{R}^n$ . Then, by (17),

$$z^T H_j^{k+1} z = \left( z - \rho_j^k z^T s_k y_j^k \right)^T H_j^k \left( z - \rho_j^k z^T s_k y_j^k \right) + \rho_j^k (z^T s_k)^2 \geq 0,$$

where the inequality follows from the fact that  $H_j^k \succ 0$  and  $\rho_j^k > 0$ . If  $z^T H_j^{k+1} z = 0$ , then  $z - \rho_j^k z^T s_k y_j^k = 0$  and  $z^T s_k = 0$ , which imply that  $z = 0$ , giving a contradiction with the definition of  $z$ . Therefore,  $z^T H_j^{k+1} z > 0$  and hence  $H_j^{k+1}$  is definite positive for each  $j = 1, \dots, m$ .  $\square$

The following example illustrates that, in multicriteria optimization, in contrast to the scalar case,  $s_k^T y_j^k$  can be nonpositive for some  $j \in \{1, \dots, m\}$  even when a Wolfe line search is used. This justifies our choice for  $\rho_j^k$  in (16), since a key condition for the well-definedness of Algorithm 1 is that  $\rho_j^k > 0$  for all  $j = 1, \dots, m$  (see Theorem 3.1).

**Example 3.1.** Let  $\beta \geq 1$ ,  $c_1 = 10^{-4}$ ,  $c_2 = 0.9$ ,  $x^0 = 0$ , and  $B_1^0 = B_2^0 = 1$  be given and consider the application of Algorithm 1 to the problem (1) with  $F: \mathbb{R} \rightarrow \mathbb{R}^2$  defined by

$$F_1(x) := \frac{x^2}{3} - x \quad \text{and} \quad F_2(x) := \begin{cases} -x, & \text{if } x < 0 \\ (1 - \beta)x^3 + (\beta - 1)x^2 - x, & \text{if } 0 \leq x < 1, \\ -\beta x + \beta - 1, & \text{if } 1 \leq x < 2, \\ \beta x^2 - 5\beta x + 5\beta - 1, & \text{if } x \geq 2. \end{cases}$$

We note that  $F$  is continuously differentiable and bounded below in  $\mathbb{R}$ . Direct calculation show that  $d^0 = 1$  and  $\alpha_0 = 1$  satisfies the Wolfe conditions (13)–(14), implying that  $x^1 = 1$ . Then, we have  $s_0^T y_1^0 = 2/3$  and  $s_0^T y_2^0 = 1 - \beta \leq 0$ . If we take  $\rho_j^k = 1/s_k^T y_j^k$  in (15), the algorithm breaks down by trying to divide by zero, in the case where  $\beta = 1$ . For  $\beta > 1$ , we would have  $B_2^1 = 1 - \beta \prec 0$ .

We end this section by establishing that Algorithm 1 satisfies the Zoutendijk-type condition introduced in [40]. This will be an important result for the convergence analysis.

**Proposition 3.2.** Suppose that  $F$  is bounded below in  $\mathcal{L}(x^0)$  and the Jacobian  $JF$  is Lipschitz continuous in an open set  $\mathcal{N}$  containing the level set  $\mathcal{L}(x^0)$ , i.e., there exists  $L > 0$  such that  $\|JF(x) - JF(y)\| \leq L\|x - y\|$  for all  $x, y \in \mathcal{N}$ . Assume that Algorithm 1 generates an infinite sequence  $\{x^k, d^k\}$ . Then,

$$\sum_{k \geq 0} \frac{\mathcal{D}(x^k, d^k)^2}{\|d^k\|^2} < \infty.$$

**Proof.** See [40, Proposition 3.3].  $\square$



## 4 Global Convergence

In this section, we present global convergence results for Algorithm 1. As is usual in the scalar case, we assume that the objective functions are strongly convex, as formally stated below.

**Assumption 4.1.** (i)  $F$  is twice continuously differentiable. (ii) The level set  $\mathcal{L}(x^0)$  is convex and there exist constants  $a, b > 0$  such that

$$a\|z\|^2 \leq z^T \nabla^2 F_j(x) z \leq b\|z\|^2, \quad \forall j = 1, \dots, m, \quad (18)$$

for all  $z \in \mathbb{R}^n$  and  $x \in \mathcal{L}(x^0)$ .

Note that, under Assumption 4.1,  $s_k^T y_j^k > 0$  and hence  $\rho_j^k = 1/s_k^T y_j^k$  for all  $j = 1, \dots, m$  and  $k \geq 0$ . In this case, (15) always reduces to the classical BFGS update for  $F_j$ . Furthermore, the assumptions of Theorem 3.1 and Proposition 3.2 are trivially satisfied.

Hereafter, for all  $k \geq 1$  and  $j = 1, \dots, m$ , we denote by  $\beta_j^k$  the angle between  $s_k$  and  $B_j^k s_k$ , i.e.,

$$\cos \beta_j^k := \frac{s_k^T B_j^k s_k}{\|s_k\| \|B_j^k s_k\|}, \quad \forall k \geq 1, \quad \forall j = 1, \dots, m.$$

The following lemma presents a key result: we prove that  $\cos \beta_j^k$  stays away from 0, *simultaneously* for all objectives, for an arbitrary fraction  $p$  of the iterates. As far as we know, in the single-criterion optimization, this result was first shown in [6, Theorem 2.1]; see also [50, Lemma 4].

**Lemma 4.1.** Suppose that Assumption 4.1 holds and that  $\{x^k\}$  is a sequence generated by Algorithm 1. Then, for any  $p \in ]0, 1[$ , there exists a constant  $\delta > 0$  such that, for any  $k \geq 1$ , the relation

$$\cos \beta_j^\ell \geq \delta, \quad \forall j = 1, \dots, m,$$

holds for at least  $\lceil p(k+1) \rceil$  values of  $\ell \in \{0, 1, \dots, k\}$ .

**Proof.** Let  $k \geq 1$  and  $p \in ]0, 1[$  be given and set  $\varepsilon := 1 - p$  and  $\bar{p} := 1 - \varepsilon/m$ . For each  $j = 1, \dots, m$ , we can proceed as in [6, Theorem 2.1] with  $\bar{p} \in ]0, 1[$  to show that there exists  $\delta_j > 0$  such that  $\cos \beta_j^\ell \geq \delta_j$  for at least  $\lceil \bar{p}(k+1) \rceil$  values of  $\ell \in \{0, 1, \dots, k\}$ . Define  $\delta := \min_{j=1, \dots, m} \delta_j$ ,  $\mathcal{G}_j^k := \{\ell \in \{0, 1, \dots, k\} \mid \cos \beta_j^\ell \geq \delta\}$ , and  $\mathcal{B}_j^k := \{\ell \in \{0, 1, \dots, k\} \mid \cos \beta_j^\ell < \delta\}$  for all  $j = 1, \dots, m$ . Note that  $\mathcal{G}_j^k \cap \mathcal{B}_j^k = \emptyset$  and  $k+1 = |\mathcal{G}_j^k| + |\mathcal{B}_j^k|$  for all  $j = 1, \dots, m$ . Therefore, by the definition of  $\bar{p}$  and using some properties of the ceiling and floor functions, we have

$$|\mathcal{G}_j^k| \geq \lceil \bar{p}(k+1) \rceil = (k+1) + \lceil -\frac{\varepsilon}{m}(k+1) \rceil = (k+1) - \lfloor \frac{\varepsilon}{m}(k+1) \rfloor,$$

and

$$|\mathcal{B}_j^k| \leq \lfloor \frac{\varepsilon}{m}(k+1) \rfloor,$$

for all  $j = 1, \dots, m$ . Thus,  $|\cup_{j=1}^m \mathcal{B}_j^k| \leq m \lfloor \frac{\varepsilon}{m}(k+1) \rfloor \leq \varepsilon(k+1)$ . As consequence, since we also have  $k+1 = |\cap_{j=1}^m \mathcal{G}_j^k| + |\cup_{j=1}^m \mathcal{B}_j^k|$ , by the definition of  $\varepsilon$ , it follows that

$$|\cap_{j=1}^m \mathcal{G}_j^k| \geq (k+1) - \varepsilon(k+1) = (1 - \varepsilon)(k+1) = p(k+1),$$

completing the proof.  $\square$

The next lemma presents a useful technical result.

**Lemma 4.2.** *Suppose that Assumption 4.1 holds and that  $\{x^k\}$  is a sequence generated by Algorithm 1. Then, for all  $k \geq 0$ ,*

$$\mathcal{D}(x^k, d^k) \leq -\frac{\delta_k}{2} \|d^k\| \|d_{SD}(x^k)\|,$$

where  $\delta_k := \min_{j=1, \dots, m} \cos \beta_j^k$ .

**Proof.** Let  $k \geq 0$  be given. By the definitions of  $\delta_k$ ,  $\cos \beta_j^k$ , and  $s_k$ , we have

$$\delta_k \leq \cos \beta_j^k = \frac{s_k^T B_j^k s_k}{\|s_k\| \|B_j^k s_k\|} = \frac{(d^k)^T B_j^k d^k}{\|d^k\| \|B_j^k d^k\|}, \quad \forall j = 1, \dots, m.$$

Therefore,

$$(d^k)^T B_j^k d^k \geq \delta_k \|d^k\| \|B_j^k d^k\|, \quad \forall j = 1, \dots, m.$$

Hence, from Lemma 2.3(ii) and (9), we obtain

$$-\mathcal{D}(x^k, d^k) > -\theta(x^k) = \frac{1}{2} \sum_{j=1}^m \lambda_j^k (d^k)^T B_j^k d^k \geq \frac{\delta_k}{2} \|d^k\| \sum_{j=1}^m \lambda_j^k \|B_j^k d^k\|.$$

Thus, the triangle inequality, together with (7), (8), and Lemma 2.4(iv), implies

$$-\mathcal{D}(x^k, d^k) \geq \frac{\delta_k}{2} \|d^k\| \sum_{j=1}^m \lambda_j^k \|B_j^k d^k\| = \frac{\delta_k}{2} \|d^k\| \sum_{j=1}^m \lambda_j^k \|\nabla F_j(x^k)\| \geq \frac{\delta_k}{2} \|d^k\| \|d_{SD}(x^k)\|,$$

obtaining the desired result.  $\square$

We are now ready to present the main convergence result of this section.

**Theorem 4.3.** *Suppose that Assumption 4.1 holds and that  $\{x^k\}$  is a sequence generated by Algorithm 1. Then,  $\{x^k\}$  converges to a Pareto optimal point  $x^*$  of  $F$ .*

**Proof.** By Lemma 4.1, there exist a constant  $\delta > 0$  and  $K \subseteq \mathbb{N}$  such that

$$\cos \beta_j^k \geq \delta, \quad \forall k \in K \quad \text{and} \quad \forall j = 1, \dots, m.$$

Therefore, Lemma 4.2 implies that

$$\mathcal{D}(x^k, d^k) \leq -\frac{\delta}{2} \|d^k\| \|d_{SD}(x^k)\|, \quad \forall k \in K.$$

Thus, from Proposition 3.2, we have

$$\infty > \sum_{k \geq 0} \frac{\mathcal{D}(x^k, d^k)^2}{\|d^k\|^2} \geq \sum_{k \in K} \frac{\mathcal{D}(x^k, d^k)^2}{\|d^k\|^2} \geq \sum_{k \in K} \frac{\delta^4}{4} \|d_{SD}(x^k)\|^2,$$

and hence

$$\lim_{k \in K} d_{SD}(x^k) = 0. \tag{19}$$

Now, since  $\mathcal{L}$  is compact and  $x^k \in \mathcal{L}$  for all  $k \in K$ , there exist  $K_1 \subset \mathbb{N}$  and  $x^* \in \mathcal{L}$  such that  $\lim_{k \in K_1} x^k = x^*$ . Thus, by (19) and Lemma 2.4(iii), we obtain  $d_{SD}(x^*) = 0$ . Hence, by Lemma 2.1(iii), we conclude that  $x^*$  is Pareto optimal.

Let us show that  $\lim_{k \rightarrow \infty} x^k = x^*$ . Suppose for contradiction that there exist  $\bar{x} \in \mathcal{L}$  with  $\bar{x} \neq x^*$  and  $K_2 \subset \mathbb{N}$  such that  $\lim_{k \in K_2} x^k = \bar{x}$ . We first claim that  $F(\bar{x}) \neq F(x^*)$ . Indeed, if  $F(\bar{x}) = F(x^*)$ , by Assumption 4.1, for all  $t \in [0, 1]$ , we have

$$F_j(t\bar{x} + (1-t)x^*) < tF_j(\bar{x}) + (1-t)F_j(x^*) = F_j(x^*), \quad \forall j = 1, \dots, m,$$

contradicting the fact that  $x^*$  is a Pareto optimal point. Hence,  $F(\bar{x}) \neq F(x^*)$  holds, as claimed. Now, since  $x^*$  is Pareto optimal, there exists  $j_0 \in \{1, \dots, m\}$  such that  $F_{j_0}(x^*) < F_{j_0}(\bar{x})$ . Therefore, remembering that  $\lim_{k \in K_1} x^k = x^*$  and  $\lim_{k \in K_2} x^k = \bar{x}$ , we can choose  $k_1 \in K_1$  and  $k_2 \in K_2$  with  $k_1 < k_2$  so that  $F_{j_0}(x^{k_1}) < F_{j_0}(x^{k_2})$ . This contradicts the sufficient decrease condition (13) which, in particular, implies that  $\{F_j(x^k)\}$  is a decreasing sequence for all  $j = 1, \dots, m$ . Thus,  $\lim_{k \rightarrow \infty} x^k = x^*$  and the proof is complete.  $\square$

In the remainder of this section, our aim is to show that  $\{x^k\}$  converges to  $x^*$  rapidly enough that

$$\sum_{k \geq 0} \|x^k - x^*\| < \infty. \quad (20)$$

To the best of our knowledge, this is the first work to establish this result for multiobjective optimization. As we will see, (20) plays an important role in the superlinear convergence. We start with some technical results.

**Lemma 4.4.** *Suppose that Assumption 4.1 holds and that  $\{x^k\}$  is a sequence generated by Algorithm 1. Let  $x^*$  be as in Theorem 4.3. Then, for all  $k \geq 0$ , we have*

$$(i) \quad \|x^k - x^*\| \leq \frac{2}{a} \|d_{SD}(x^k)\|;$$

$$(ii) \quad \|s_k\| \geq \frac{(1-c_2)}{2b} \delta_k \|d_{SD}(x^k)\|, \text{ where } \delta_k \text{ is given as in Lemma 4.2.}$$

**Proof.** Let  $k \geq 0$  be given and consider  $\lambda^{SD}(x^k) \in \mathbb{R}^m$  as in (11)-(12), i.e.  $\lambda^{SD}(x^k)$  is such that  $d_{SD}(x^k) = -\sum_{j=1}^m \lambda_j^{SD}(x^k) \nabla F_j(x^k)$ . Define the scalar-valued function  $F_{SD}: \mathbb{R}^n \rightarrow \mathbb{R}$  by

$$F_{SD}(x) := \sum_{j=1}^m \lambda_j^{SD}(x^k) F_j(x).$$

Therefore, by taking  $z := x^* - x^k$ , it follows from (11) and (18) that

$$\int_0^1 (1-\tau) z^T \nabla^2 F_{SD}(x^k + \tau z) z d\tau \geq \frac{a}{2} \|z\|^2.$$

Thus, by evaluating the above integral (which can be done by integration by parts), and considering that  $d_{SD}(x^k) = -\nabla F_{SD}(x^k)$ , we obtain

$$F_{SD}(x^*) - F_{SD}(x^k) + d_{SD}(x^k)^T (x^* - x^k) \geq \frac{a}{2} \|x^* - x^k\|^2.$$

Since  $F_j(x^*) \leq F_j(x^k)$  for all  $j = 1, \dots, m$ , we have  $F_{SD}(x^*) - F_{SD}(x^k) \leq 0$  and hence

$$\frac{a}{2} \|x^* - x^k\|^2 \leq d_{SD}(x^k)^T (x^* - x^k) \leq \|d_{SD}(x^k)\| \|x^* - x^k\|,$$

proving part (i).

Let us consider part (ii). Let  $k \geq 0$  be given and define

$$\bar{G}_j^k := \int_0^1 \nabla^2 F_j(x^k + \tau s_k) d\tau, \quad \forall j = 1, \dots, m.$$

Then,

$$\bar{G}_j^k s_k = y_j^k, \quad \forall j = 1, \dots, m. \quad (21)$$

Now, by (14) and the definition of  $\mathcal{D}$ , it follows that

$$\begin{aligned} (c_2 - 1) \mathcal{D}(x^k, d^k) &\leq \mathcal{D}(x^{k+1}, d^k) - \mathcal{D}(x^k, d^k) \\ &\leq \max_{j=1, \dots, m} (\nabla F_j(x^{k+1}) - \nabla F_j(x^k))^T d^k = \max_{j=1, \dots, m} (y_j^k)^T d^k, \end{aligned}$$

where the second inequality follows from the fact that, for any  $u, v \in \mathbb{R}^m$ , we have  $\max_j (u_j - v_j) \geq \max_j u_j - \max_j v_j$ . Therefore, by (21), we obtain

$$(c_2 - 1) \mathcal{D}(x^k, d^k) \leq \max_{j=1, \dots, m} s_k^T \bar{G}_j^k d^k = \alpha_k \max_{j=1, \dots, m} (d^k)^T \bar{G}_j^k d^k \leq b \alpha_k \|d^k\|^2 = b \|s_k\| \|d^k\|,$$

where the second inequality comes from (18). Hence, using Lemma 4.2 and taking into account that  $c_2 < 1$ , we have

$$-(c_2 - 1) \frac{\delta_k}{2} \|d^k\| \|d_{SD}(x^k)\| \leq b \|s_k\| \|d^k\|,$$

concluding the proof.  $\square$

**Theorem 4.5.** Suppose that Assumption 4.1 holds and that  $\{x^k\}$  is a sequence generated by Algorithm 1. Let  $x^*$  be as in Theorem 4.3. Then,  $\{x^k\}$  converges  $R$ -linearly to  $x^*$ . As a consequence, (20) holds.

**Proof.** Let  $\lambda^{SD}(x^*) \in \mathbb{R}^m$  be a steepest descent multiplier associated to  $x^*$  as in (11)-(12), i.e.,  $d_{SD}(x^*) = -\sum_{j=1}^m \lambda_j^{SD}(x^*) \nabla F_j(x^*)$ . Let us define the scalar-valued function  $F_*: \mathbb{R}^n \rightarrow \mathbb{R}$  given by

$$F_*(x) := \sum_{j=1}^m \lambda_j^{SD}(x^*) F_j(x).$$

Note that

$$\nabla F_*(x^*) = \sum_{j=1}^m \lambda_j^{SD}(x^*) \nabla F_j(x^*) = -d_{SD}(x^*) = 0, \quad (22)$$

where the last equality comes from Lemma 2.4(i). Now, by doing a second-order Taylor series expansion of  $F_j$  around  $x^*$  and using (18), we obtain

$$\begin{aligned} \nabla F_j(x^*)^T (x^k - x^*) + \frac{a}{2} \|x^k - x^*\|^2 &\leq F_j(x^k) - F_j(x^*) \\ &\leq \nabla F_j(x^*)^T (x^k - x^*) + \frac{b}{2} \|x^k - x^*\|^2, \end{aligned}$$

for all  $j = 1, \dots, m$  and for all  $k \geq 0$ . By multiplying this expression by  $\lambda_j^{SD}(x^*)$ , summing over all indices  $j = 1, \dots, m$ , and taking into account (11) and (22), we obtain

$$\frac{a}{2}\|x^k - x^*\|^2 \leq F_*(x^k) - F_*(x^*) \leq \frac{b}{2}\|x^k - x^*\|^2, \quad \forall k \geq 0. \quad (23)$$

From the right hand side of (23) and Lemma 4.4(i), we obtain

$$F_*(x^k) - F_*(x^*) \leq \frac{2b}{a^2}\|d_{SD}(x^k)\|^2, \quad \forall k \geq 0. \quad (24)$$

On the other hand, similarly to (23), the sufficient descent condition (13) implies

$$F_*(x^{k+1}) \leq F_*(x^k) + c_1\alpha_k\mathcal{D}(x^k, d^k), \quad \forall k \geq 0.$$

By subtracting the term  $F_*(x^*)$  in both sides of this inequality and using Lemma 4.2, we have

$$F_*(x^{k+1}) - F_*(x^*) \leq F_*(x^k) - F_*(x^*) - \frac{c_1}{2}\delta_k\|s_k\|\|d_{SD}(x^k)\|,$$

for all  $k \geq 0$ , where  $\delta_k = \min_{j=1, \dots, m} \cos \beta_j^k$ . Therefore, by Lemma 4.4(ii), it follows that

$$F_*(x^{k+1}) - F_*(x^*) \leq F_*(x^k) - F_*(x^*) - \frac{c_1(1-c_2)}{4b}\delta_k^2\|d_{SD}(x^k)\|^2, \quad \forall k \geq 0.$$

Hence, by (24), we have

$$F_*(x^{k+1}) - F_*(x^*) \leq \left(1 - \frac{c_1(1-c_2)a^2}{8b^2}\delta_k^2\right) (F_*(x^k) - F_*(x^*)), \quad \forall k \geq 0. \quad (25)$$

For each  $k \geq 0$ , define  $\bar{r}_k := 1 - c_1(1-c_2)a^2\delta_k^2/(8b^2)$ . It is easy to see that  $\bar{r}_k \in ]0, 1]$ , for all  $k \geq 0$ .

Now, let  $p \in ]0, 1[$  be given. Then, by Lemma 4.1, there exists a constant  $\delta > 0$  such that, for any  $k \geq 1$ , the number of elements  $\ell \in \{0, 1, \dots, k\}$  for which  $\delta_\ell \geq \delta$  is at least  $\lceil p(k+1) \rceil$ . Hence, by defining  $\mathcal{G}_k := \{\ell \in \{0, 1, \dots, k\} \mid \delta_\ell \geq \delta\}$ , we have  $|\mathcal{G}_k| \geq \lceil p(k+1) \rceil$  and

$$\bar{r}_\ell \leq 1 - \frac{c_1(1-c_2)a^2\delta^2}{8b^2} := \bar{r} < 1, \quad \forall \ell \in \mathcal{G}_k.$$

Therefore, by (25) and taking into account that  $F_*(x^0) - F_*(x^*) > 0$ , we obtain, for all  $k \geq 1$ ,

$$\begin{aligned} F_*(x^{k+1}) - F_*(x^*) &\leq \left[ \prod_{\ell=0}^k \bar{r}_\ell \right] (F_*(x^0) - F_*(x^*)) \leq \left[ \prod_{\ell \in \mathcal{G}_k} \bar{r}_\ell \right] (F_*(x^0) - F_*(x^*)) \\ &\leq \left[ \prod_{\ell \in \mathcal{G}_k} \bar{r} \right] (F_*(x^0) - F_*(x^*)) \leq \bar{r}^{\lceil p(k+1) \rceil} (F_*(x^0) - F_*(x^*)), \end{aligned}$$

where the second inequality follows from the fact that  $\bar{r}_\ell \leq 1$  for all  $\ell \notin \mathcal{G}_k$ . Thus, by defining  $r^{1/p} := \bar{r}$ , we have

$$F_*(x^{k+1}) - F_*(x^*) \leq r^{k+1} (F_*(x^0) - F_*(x^*)), \quad \forall k \geq 1. \quad (26)$$

By combining the left hand side of (23) with the above inequality, we find that

$$\|x^{k+1} - x^*\| \leq \left[ \frac{2}{a} \left( F_*(x^0) - F_*(x^*) \right) \right]^{1/2} (r^{1/2})^{k+1},$$

and hence  $\{x^k\}$  converges R-linearly to  $x^*$ . Finally, by summing this expression and taking into account that  $r < 1$ , we conclude that (20) holds.  $\square$

**Remark 4.1.** Note that (26) implies that  $\{F_*(x^k)\}$  is R-linearly convergent to  $F_*(x^*)$ . It is worth mentioning that Theorem 4.5 can be seen as the extension of [6, Theorem 3.1] for multiobjective optimization; see also [50, Lemma 5].

## 5 Superlinear Local Convergence

Now we study the local convergence properties of Algorithm 1. The results of this section also apply to nonconvex problems, although it is not possible to establish global convergence in this general case. We will assume that  $\{x^k\}$  converges to a Pareto optimal point  $x^*$  and prove, under suitable assumptions, that the rate of convergence is Q-superlinear.

**Assumption 5.1.** (i)  $F$  is twice continuously differentiable. (ii) The sequence  $\{x^k\}$  generated by Algorithm 1 converges to a Pareto optimal point  $x^*$ . (iii) For each  $j = 1, \dots, m$ ,  $\nabla^2 F_j(x^*)$  is positive definite and  $L$ -Lipschitz continuous at  $x^*$ . Thus, there exist a neighborhood  $U$  of  $x^*$  and positive constants  $a$ ,  $b$ , and  $L$  such that

$$a\|z\|^2 \leq z^T \nabla^2 F_j(x) z \leq b\|z\|^2, \quad \forall j = 1, \dots, m, \quad (27)$$

and

$$\|\nabla^2 F_j(x) - \nabla^2 F_j(x^*)\| \leq L\|x - x^*\|, \quad \forall j = 1, \dots, m, \quad (28)$$

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

Essentially, Assumption 5.1 says that, in a neighborhood  $U$  of  $x^*$ ,  $F$  is strongly convex and the Hessians  $\nabla^2 F_j$  ( $j = 1, \dots, m$ ) are Lipschitz continuous at  $x^*$ . Throughout this section, we assume, without loss of generality, that  $\{x^k\} \subset U$ , i.e., (27) and (28) hold at  $x^k$  for all  $k \geq 0$ . Since Assumption 5.1 is more restrictive than Assumption 4.1, the results of the previous section remain true and will be used here without further explanation.

The next theorem establishes that the Dennis–Moré condition [18] holds individually for each objective function  $F_j$  (see (30) below). The proof of this result is quite straightforward from the scalar case and its details will therefore be omitted. We emphasize, however, that (20) plays an essential role in this task. We also include in the statement of the theorem an intermediate step (see (29) below) that will be evoked in forthcoming results.

**Theorem 5.1.** Suppose that Assumption 5.1 holds. Then,

$$\lim_{k \rightarrow \infty} \frac{s_k^T B_j^k s_k}{s_k^T \nabla^2 F_j(x^*) s_k} = 1, \quad \forall j = 1, \dots, m. \quad (29)$$

and

$$\lim_{k \rightarrow \infty} \frac{\|(B_j^k - \nabla^2 F_j(x^*))d^k\|}{\|d^k\|} = 0, \quad \forall j = 1, \dots, m, \quad (30)$$

or, equivalently,

$$\lim_{k \rightarrow \infty} \frac{\|(B_j^k - \nabla^2 F_j(x^k))d^k\|}{\|d^k\|} = 0, \quad \forall j = 1, \dots, m. \quad (31)$$

**Proof.** Using (20), the proof follows similarly to [48, Theorem 6.6] (see also [6, Theorem 3.2]).  $\square$

Let  $\lambda^k := \lambda(x^k) \in \mathbb{R}^m$  be the Lagrange multiplier associated to  $x^k$  of problem (6) fulfilling (7)–(9). From now on, we define, for all  $k \geq 0$ ,

$$F_\lambda^k(x) := \sum_{j=1}^m \lambda_j^k F_j(x) \quad \text{and} \quad B_\lambda^k := \sum_{j=1}^m \lambda_j^k B_j^k. \quad (32)$$

In the following, we show that the family of functions  $\{F_\lambda^k(x)\}_{k \geq 0}$  satisfies a Dennis–Moré-type condition.

**Theorem 5.2.** *Suppose that Assumption 5.1 holds. For each  $k \geq 0$ , consider  $F_\lambda^k: \mathbb{R}^n \rightarrow \mathbb{R}$  and  $B_\lambda^k$  as in (32). Then,*

$$\lim_{k \rightarrow \infty} \frac{\|(B_\lambda^k - \nabla^2 F_\lambda^k(x^*))d^k\|}{\|d^k\|} = 0 \quad (33)$$

or, equivalently,

$$\lim_{k \rightarrow \infty} \frac{\|\nabla F_\lambda^k(x^k) + \nabla^2 F_\lambda^k(x^k)d^k\|}{\|d^k\|} = 0. \quad (34)$$

**Proof.** From the definitions of  $B_\lambda^k$  and  $F_\lambda^k$ , using the triangle inequality, and taking into account (8), we have

$$\begin{aligned} \lim_{k \rightarrow \infty} \frac{\|(B_\lambda^k - \nabla^2 F_\lambda^k(x^*))d^k\|}{\|d^k\|} &= \lim_{k \rightarrow \infty} \frac{\|\sum_{j=1}^m \lambda_j^k (B_j^k - \nabla^2 F_j(x^*))d^k\|}{\|d^k\|} \\ &\leq \lim_{k \rightarrow \infty} \sum_{j=1}^m \lambda_j^k \frac{\|(B_j^k - \nabla^2 F_j(x^*))d^k\|}{\|d^k\|} \\ &\leq \lim_{k \rightarrow \infty} \max_{j=1, \dots, m} \frac{\|(B_j^k - \nabla^2 F_j(x^*))d^k\|}{\|d^k\|}. \end{aligned}$$

This inequality, together with (30), yields (33). Let us prove that (33) implies (34). First, note that, by (7), we have  $B_\lambda^k d^k = -\nabla F_\lambda^k(x^k)$  and hence (34) is equivalent to

$$\lim_{k \rightarrow \infty} \frac{\|(B_\lambda^k - \nabla^2 F_\lambda^k(x^k))d^k\|}{\|d^k\|} = 0. \quad (35)$$

Now, it is easy to see that

$$\begin{aligned} \lim_{k \rightarrow \infty} \frac{\|(B_\lambda^k - \nabla^2 F_\lambda^k(x^k))d^k\|}{\|d^k\|} &\leq \lim_{k \rightarrow \infty} \frac{\|(B_\lambda^k - \nabla^2 F_\lambda^k(x^*))d^k\|}{\|d^k\|} \\ &\quad + \lim_{k \rightarrow \infty} \|\nabla^2 F_\lambda^k(x^*) - \nabla^2 F_\lambda^k(x^k)\| \end{aligned} \quad (36)$$

and

$$\begin{aligned} \lim_{k \rightarrow \infty} \|\nabla^2 F_\lambda^k(x^*) - \nabla^2 F_\lambda^k(x^k)\| &= \lim_{k \rightarrow \infty} \left\| \sum_{j=1}^m \lambda_j^k (\nabla^2 F_j(x^*) - \nabla^2 F_j(x^k)) \right\| \\ &\leq \lim_{k \rightarrow \infty} \sum_{j=1}^m \lambda_j^k \|\nabla^2 F_j(x^*) - \nabla^2 F_j(x^k)\| \\ &\leq \lim_{k \rightarrow \infty} \max_{j=1, \dots, m} \|\nabla^2 F_j(x^*) - \nabla^2 F_j(x^k)\| = 0, \end{aligned}$$

because  $\nabla^2 F_j(\cdot)$  is continuous for every  $j = 1, \dots, m$ . Therefore, (35) follows from (36) and (33). The proof that (34) implies (33) can be obtained similarly.  $\square$

The following auxiliary result provides some properties related to the length of direction  $d^k$  and the optimal value  $\theta(x^k)$ .

**Lemma 5.3.** *Suppose that Assumption 5.1 holds. Then, there exist positive constants  $\bar{a}$  and  $\bar{b}$  such that, for all  $k$  sufficiently large, we have:*

$$\bar{a}\|d^k\|^2 \leq |\theta(x^k)| \leq \bar{b}\|d^k\|^2. \quad (37)$$

Moreover,

$$\lim_{k \rightarrow \infty} \|d^k\| = 0. \quad (38)$$

**Proof.** By (29), taking  $\gamma \in ]0, 1[$ , we obtain

$$1 - \gamma \leq \frac{s_k^T B_j^k s_k}{s_k^T \nabla^2 F_j(x^*) s_k} \leq 1 + \gamma, \quad \forall j = 1, \dots, m, \quad (39)$$

for all  $k$  sufficiently large. On the other hand, by (27), it follows that

$$\frac{1}{b} \frac{s_k^T B_j^k s_k}{\|s_k\|^2} \leq \frac{s_k^T B_j^k s_k}{s_k^T \nabla^2 F_j(x^*) s_k} \leq \frac{1}{a} \frac{s_k^T B_j^k s_k}{\|s_k\|^2}, \quad \forall k \geq 0, \quad \forall j = 1, \dots, m.$$

Therefore, by using (39) and recalling that  $s_k = \alpha_k d^k$ , we have

$$a(1 - \gamma) \leq \frac{s_k^T B_j^k s_k}{\|s_k\|^2} = \frac{(d^k)^T B_j^k d^k}{\|d^k\|^2} \leq b(1 + \gamma), \quad \forall j = 1, \dots, m,$$

for all  $k$  sufficiently large. Hence, by (8) and (9), we obtain

$$\frac{a(1 - \gamma)}{2} \|d^k\|^2 \leq |\theta(x^k)| = \frac{1}{2} \sum_{j=1}^m \lambda_j^k (d^k)^T B_j^k d^k \leq \frac{b(1 + \gamma)}{2} \|d^k\|^2,$$

for all  $k$  sufficiently large. By defining  $\bar{a} := a(1 - \gamma)/2$  and  $\bar{b} := b(1 + \gamma)/2$ , we prove (37).

Finally, by (37), we have

$$0 \leq \lim_{k \rightarrow \infty} \bar{a}\|d^k\| \leq \lim_{k \rightarrow \infty} \frac{|\theta(x^k)|}{\|d^k\|} \leq \lim_{k \rightarrow \infty} \frac{|\mathcal{D}(x^k, d^k)|}{\|d^k\|} = 0,$$

where the second inequality follows from the fact that  $\mathcal{D}(x^k, d^k) < \theta(x^k) < 0$  (see Lemma 2.3(ii)) and the final equality is a consequence of Proposition 3.2. This concludes the proof.  $\square$

The following result shows that the unit step size satisfies the Wolfe conditions (13)–(14) as the iterates converge to  $x^*$ .

**Theorem 5.4.** *Suppose that Assumption 5.1 holds. Then, the step size  $\alpha_k = 1$  is admissible for all  $k$  sufficiently large.*



**Proof.** Let  $j \in \{1, \dots, m\}$  be an arbitrary index. By Taylor's formula, we have

$$\begin{aligned}
F_j(x^k + d^k) &= F_j(x^k) + \nabla F_j(x^k)^T d^k + \frac{1}{2}(d^k)^T \nabla^2 F_j(x^k) d^k + o(\|d^k\|^2) \\
&= F_j(x^k) + \nabla F_j(x^k)^T d^k + \frac{1}{2}(d^k)^T B_j^k d^k \\
&\quad + \frac{1}{2}(d^k)^T \left( \nabla^2 F_j(x^k) - B_j^k \right) d^k + o(\|d^k\|^2) \\
&= F_j(x^k) + \nabla F_j(x^k)^T d^k + \frac{1}{2}(d^k)^T B_j^k d^k + o(\|d^k\|^2),
\end{aligned}$$

where the last equality is a consequence of (31). Therefore, by (5),

$$F_j(x^k + d^k) \leq F_j(x^k) + t\theta(x^k) + (1-t)\theta(x^k) + o(\|d^k\|^2),$$

where  $t := 2c_1 < 1$ . Thus, by (37), we have, for all  $k$  sufficiently large,

$$\begin{aligned}
F_j(x^k + d^k) &\leq F_j(x^k) + t\theta(x^k) - \bar{a}(1-t)\|d^k\|^2 + o(\|d^k\|^2), \\
&= F_j(x^k) + t\theta(x^k) + \left[ -\bar{a}(1-t) + \frac{o(\|d^k\|^2)}{\|d^k\|^2} \right] \|d^k\|^2.
\end{aligned}$$

For  $k$  large enough, the term in square brackets is negative and then

$$F_j(x^k + d^k) \leq F_j(x^k) + t\theta(x^k).$$

On the other hand, combining (7)–(9), we obtain

$$\theta(x^k) = \frac{1}{2} \sum_{j=1}^m \lambda_j^k \nabla F_j(x^k)^T d^k \leq \frac{1}{2} \mathcal{D}(x^k, d^k).$$

It follows that from the last two inequalities and the definition of  $t$  that

$$F_j(x^k + d^k) \leq F_j(x^k) + c_1 \mathcal{D}(x^k, d^k),$$

for all  $k$  sufficiently large. Since  $j \in \{1, \dots, m\}$  was arbitrary, we conclude that the step size  $\alpha_k = 1$  satisfies (13) for all  $k$  sufficiently large.

Consider now the curvature condition (14). From the definition of  $F_\lambda^k$  in (32), we have

$$\begin{aligned}
-\sum_{j=1}^m \lambda_j^k \nabla F_j(x^k)^T d^k &= (d^k)^T \sum_{j=1}^m \lambda_j^k \nabla^2 F_j(x^k) d^k - \sum_{j=1}^m \lambda_j^k \left[ \nabla^2 F_j(x^k) d^k + \nabla F_j(x^k) \right]^T d^k \\
&= (d^k)^T \sum_{j=1}^m \lambda_j^k \nabla^2 F_j(x^k) d^k - \left[ \nabla F_\lambda^k(x^k) + \nabla^2 F_\lambda^k(x^k) d^k \right]^T d^k.
\end{aligned}$$

Therefore, by (27), (8), and (34), we obtain

$$-\sum_{j=1}^m \lambda_j^k \nabla F_j(x^k)^T d^k \geq a\|d^k\|^2 + o(\|d^k\|^2) = \|d^k\|^2 \left[ a + \frac{o(\|d^k\|^2)}{\|d^k\|^2} \right].$$

Hence, by (38), for  $k$  sufficiently large, it follows that

$$-\sum_{j=1}^m \lambda_j^k \nabla F_j(x^k)^T d^k \geq \frac{a}{2} \|d^k\|^2. \quad (40)$$

On the other hand, by applying the mean value theorem to the scalar function  $\nabla F_\lambda^k(\cdot)^T d^k$ , there exists  $v^k := x^k + t_k d^k$  for some  $t_k \in ]0, 1[$  such that

$$\nabla F_\lambda^k(x^k + d^k)^T d^k = \nabla F_\lambda^k(x^k)^T d^k + (d^k)^T \nabla^2 F_\lambda^k(v^k) d^k.$$

Therefore,

$$\frac{|\nabla F_\lambda^k(x^k + d^k)^T d^k|}{\|d^k\|^2} \leq \frac{\|\nabla F_\lambda^k(x^k) + \nabla^2 F_\lambda^k(x^k) d^k\|}{\|d^k\|} + \|\nabla^2 F_\lambda^k(v^k) - \nabla^2 F_\lambda^k(x^k)\|. \quad (41)$$

Now, by the definitions of  $F_\lambda^k$  and  $v^k$ , and taking into account (8) and (38), we obtain

$$\begin{aligned} \lim_{k \rightarrow \infty} \|\nabla^2 F_\lambda^k(v^k) - \nabla^2 F_\lambda^k(x^k)\| &= \lim_{k \rightarrow \infty} \left\| \sum_{j=1}^m \lambda_j^k (\nabla^2 F_j(x^k + t_k d^k) - \nabla^2 F_j(x^k)) \right\| \\ &\leq \lim_{k \rightarrow \infty} \sum_{j=1}^m \lambda_j^k \|\nabla^2 F_j(x^k + t_k d^k) - \nabla^2 F_j(x^k)\| \\ &\leq \lim_{k \rightarrow \infty} \max_{j=1, \dots, m} \|\nabla^2 F_j(x^k + t_k d^k) - \nabla^2 F_j(x^k)\| = 0. \end{aligned}$$

Hence, it follows from (34) and (41) that

$$\lim_{k \rightarrow \infty} \frac{|\nabla F_\lambda^k(x^k + d^k)^T d^k|}{\|d^k\|^2} = 0.$$

Thus, for  $k$  large enough, we have

$$|\nabla F_\lambda^k(x^k + d^k)^T d^k| \leq c_2 \frac{a}{4} \|d^k\|^2,$$

which, together with (40), yields

$$\sum_{j=1}^m \lambda_j^k \nabla F_j(x^k + d^k)^T d^k = \nabla F_\lambda^k(x^k + d^k)^T d^k \geq -c_2 \frac{a}{4} \|d^k\|^2 \geq \frac{c_2}{2} \sum_{j=1}^m \lambda_j^k \nabla F_j(x^k)^T d^k.$$

Finally, remembering that  $\theta(x^k) = (1/2) \sum_{j=1}^m \lambda_j^k \nabla F_j(x^k)^T d^k$ , it follows from the last inequality and Lemma 2.3(ii) that

$$\mathcal{D}(x^k + d^k, d^k) \geq \sum_{j=1}^m \lambda_j^k \nabla F_j(x^k + d^k)^T d^k \geq c_2 \theta(x^k) \geq c_2 \mathcal{D}(x^k, d^k),$$

for all  $k$  sufficiently large, concluding the proof.  $\square$

The next lemma presents a useful inequality that will be used in our main result. Its proof can be found in [19, Lemma 4.1.15] and will therefore be omitted here.

**Lemma 5.5.** *Suppose that Assumption 5.1 holds. Then, for all  $j = 1, \dots, m$ , there holds:*

$$\begin{aligned} & \|\nabla F_j(x^{k+1}) - \nabla F_j(x^k) - \nabla^2 F_j(x^*)(x^{k+1} - x^k)\| \\ & \leq L \frac{\|x^{k+1} - x^*\| + \|x^k - x^*\|}{2} \|x^{k+1} - x^k\|, \end{aligned}$$

where  $L$  is given in (28).

We are now able to prove the superlinear convergence of Algorithm 1. This result is based on [18, Theorem 2.2].

**Theorem 5.6.** *Suppose that Assumption 5.1 holds. Then,  $\{x^k\}$  converges to  $x^*$   $Q$ -superlinearly.*

**Proof.** By Theorem 5.4, we may assume, without loss of generality, that  $\alpha_k = 1$  and hence  $d^k = x^{k+1} - x^k$ , for all  $k$ . Thus, by (7), we have  $B_\lambda^k(x^{k+1} - x^k) = -\nabla F_\lambda^k(x^k)$  and then

$$\begin{aligned} (B_\lambda^k - \nabla^2 F_\lambda^k(x^*))(x^{k+1} - x^k) &= \nabla F_\lambda^k(x^{k+1}) - \nabla F_\lambda^k(x^k) \\ &\quad - \nabla^2 F_\lambda^k(x^*)(x^{k+1} - x^k) - \nabla F_\lambda^k(x^{k+1}). \end{aligned}$$

Therefore,

$$\begin{aligned} \frac{\|\nabla F_\lambda^k(x^{k+1})\|}{\|x^{k+1} - x^k\|} &\leq \frac{\|(B_\lambda^k - \nabla^2 F_\lambda^k(x^*))(x^{k+1} - x^k)\|}{\|x^{k+1} - x^k\|} \\ &\quad + \frac{\|\nabla F_\lambda^k(x^{k+1}) - \nabla F_\lambda^k(x^k) - \nabla^2 F_\lambda^k(x^*)(x^{k+1} - x^k)\|}{\|x^{k+1} - x^k\|}. \end{aligned} \tag{42}$$

Note that the second term on the right hand side of this inequality is less than or equal to

$$\max_{j=1, \dots, m} \frac{\|\nabla F_j(x^{k+1}) - \nabla F_j(x^k) - \nabla^2 F_j(x^*)(x^{k+1} - x^k)\|}{\|x^{k+1} - x^k\|}$$

and, by Lemma 5.5, this last expression is less than or equal to

$$L \frac{\|x^{k+1} - x^*\| + \|x^k - x^*\|}{2}.$$

Hence, taking limits on both sides of (42) and using (33), we get

$$\lim_{k \rightarrow \infty} \frac{\|\nabla F_\lambda^k(x^{k+1})\|}{\|x^{k+1} - x^k\|} = 0. \tag{43}$$

On the other hand, by the definition of  $F_\lambda^k$ , Lemma 2.4(iv), and Lemma 4.4(i), we find that

$$\begin{aligned} \frac{\|\nabla F_\lambda^k(x^{k+1})\|}{\|x^{k+1} - x^k\|} &\geq \frac{\|\sum_{j=1}^m \lambda_j^k \nabla F_j(x^{k+1})\|}{\|x^{k+1} - x^*\| + \|x^k - x^*\|} \geq \frac{\|d_{SD}(x^{k+1})\|}{\|x^{k+1} - x^*\| + \|x^k - x^*\|} \\ &\geq \frac{a}{2} \frac{\|x^{k+1} - x^*\|}{\|x^{k+1} - x^*\| + \|x^k - x^*\|} = \frac{a}{2} \frac{1}{1 + \frac{\|x^k - x^*\|}{\|x^{k+1} - x^*\|}}. \end{aligned}$$

Therefore, by using (43), we obtain

$$\lim_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0,$$

and hence that the rate of convergence is  $Q$ -superlinear.  $\square$

## 6 Numerical Experiments

This section presents some numerical results in order to illustrate the potential practical advantages of Algorithm 1. We are mainly interested in verifying the effectiveness of using a Wolfe line search procedure and updating the Hessian approximations at each iteration in a BFGS scheme. For this purpose, we considered the following methods in the reported tests:

- Algorithm 1 (BFGS-Wolfe): our proposed scheme in which the Hessian approximations are updated at each iteration by (15) and the step sizes are calculated satisfying the Wolfe conditions (13)-(14).
- Standard BFGS-Armijo: a BFGS algorithm in which the Hessian approximations are updated, for each  $j = 1, \dots, m$ , by

$$B_j^{k+1} := \begin{cases} B_j^k - \frac{B_j^k s_k s_k^T B_j^k}{s_k^T B_j^k s_k} + \frac{y_j^k (y_j^k)^T}{s_k^T y_j^k}, & \text{if } s_k^T y_j^k \geq \varepsilon \min\{1, |\theta(x^k)|\}, \\ B_j^k, & \text{otherwise,} \end{cases} \quad (44)$$

where  $\varepsilon > 0$  is an algorithmic parameter and the step sizes are calculated satisfying the Armijo-type condition (13). In our experiments, we used  $\varepsilon = 10^{-6}$ .

- Standard BFGS-Wolfe: a BFGS algorithm in which the Hessian approximations are updated by (44) and the step sizes are calculated satisfying the Wolfe conditions (13)-(14).

To the best of our knowledge, until the present work, *cautious updates* as in (44) were the only alternatives to BFGS methods when applied to nonconvex multiobjective problems. The update scheme (44) was proposed in [52] and is similar to the one used in [38] for scalar optimization.

We implemented the algorithms in Fortran 90. The search directions  $d(x^k)$  and the optimal values  $\theta(x^k)$  were calculated by solving subproblem (6) using Algencan [4], an augmented Lagrangian code for general nonlinear programming. For computing a step size satisfying the Wolfe conditions (13)-(14), we used the algorithm proposed in [41]. This algorithm involves several quadratic/cubic polynomial interpolations of the objective functions, combines backtracking and extrapolation strategies, and is capable of calculating the step size in a finite number of (inner) iterations. Interpolations techniques were also used to compute step sizes that satisfy only the Armijo-type condition (13). We used  $c_1 = 10^{-4}$ ,  $c_2 = 0.1$ , and set  $B_j^0 = I_n$  for all  $j = 1, \dots, m$ . We stopped the algorithms at  $x^k$  reporting convergence when  $|\theta(x^k)| \leq 5 \times \text{eps}^{1/2}$ , where  $\text{eps} = 2^{-52} \approx 2.22 \times 10^{-16}$  is the machine precision. The maximum number of allowed iterations was set to 2000. Our codes are freely available at <https://github.com/lfrudente/bfgs>.

Although the BFGS method enjoys global convergence only under convexity assumptions, favorable numerical experiences are also commonly observed for nonconvex problems. Thus, the set of test problems chosen includes convex and nonconvex problems commonly found in the multiobjective optimization literature. Table 1 shows their main characteristics. The first two columns contain the problem name and the corresponding reference. Columns “ $n$ ” and “ $m$ ” give the number of variables and the number of objectives, respectively. “Conv.” indicates whether the problem is convex or not. Many problems have box constraints in their original definitions. In some of them, the objectives are unbounded outside the box. In these cases, we added a term that penalizes the lack of fulfillment of the constraints to each objective. If we denote the box by  $\{x \in \mathbb{R}^n \mid l \leq x \leq u\}$  where  $l, u \in \mathbb{R}^n$ , the penalty term is defined by

$\frac{\mu}{3} [\|\max\{0, x - u\}\|_3^3 + \|\max\{0, -x + l\}\|_3^3]$ , where  $\mu = 10^{10}$  and the maximum is taken componentwise. This forces the iterates to remain inside the box. Column “Penal.” reports whether a given problem was penalized or not. The starting points were taken belonging to the corresponding boxes. We point out that the boxes are not considered by the algorithms themselves.

Problem	Ref.	$n$	$m$	Conv.	Penal.
AP1	[1]	2	3	Y	N
AP2	[1]	1	2	Y	N
AP3	[1]	2	2	N	N
AP4	[1]	3	3	Y	N
BK1	[33]	2	2	Y	N
DD1	[15]	5	2	N	Y
DGO1	[33]	1	2	N	N
DGO2	[33]	1	2	Y	Y
DTLZ1	[17]	7	3	N	Y
DTLZ2	[17]	7	3	N	Y
DTLZ3	[17]	7	3	N	Y
DTLZ4	[17]	7	3	N	Y
FA1	[33]	3	3	N	Y
Far1	[33]	2	2	N	N
FDS	[24]	5	3	Y	N
FF1	[33]	2	2	N	N
Hil1	[32]	2	2	N	N
IKK1	[33]	2	3	Y	N
IM1	[33]	2	2	N	Y
JOS1	[34]	2	2	Y	N
JOS4	[34]	20	2	N	Y
KW2	[35]	2	2	N	Y
LE1	[33]	2	2	N	N
Lov1	[39]	2	2	Y	N
Lov2	[39]	2	2	N	Y
Lov3	[39]	2	2	N	N
Lov4	[39]	2	2	N	N
Lov5	[39]	3	2	N	N
Lov6	[39]	6	2	N	Y
LTDZ	[33]	3	3	N	Y
MGH9 <sup>a</sup>	[46]	3	15	N	Y
MGH16 <sup>a</sup>	[46]	4	5	N	N
MGH26 <sup>a</sup>	[46]	4	4	N	N
MGH33 <sup>a</sup>	[46]	10	10	Y	N

Problem	Ref.	$n$	$m$	Conv.	Penal.
MHHM2	[33]	2	3	Y	N
MLF1	[33]	1	2	N	Y
MLF2	[33]	2	2	N	N
MMR1	[44]	2	2	N	Y
MMR2	[44]	2	2	N	Y
MMR3	[44]	2	2	N	Y
MMR4	[44]	3	2	N	Y
MOP2	[33]	2	2	N	N
MOP3	[33]	2	2	N	N
MOP5	[33]	2	3	N	N
MOP6	[33]	2	2	N	Y
MOP7	[33]	2	3	Y	N
PNR	[51]	2	2	Y	N
QV1	[33]	10	2	N	N
SD	[55]	4	2	Y	Y
SK1	[33]	1	2	N	N
SK2	[33]	4	2	N	N
SLCDT1	[54]	2	2	N	N
SLCDT2	[54]	10	3	Y	N
SP1	[33]	2	2	Y	N
SSFYY2	[33]	1	2	N	N
TKLY1	[33]	4	2	N	Y
Toi4 <sup>a</sup>	[58]	4	2	Y	N
Toi8 <sup>a</sup>	[58]	3	3	Y	N
Toi9 <sup>a</sup>	[58]	4	4	N	N
Toi10 <sup>a</sup>	[58]	4	3	N	N
VU1	[33]	2	2	N	N
VU2	[33]	2	2	Y	Y
ZDT1	[60]	30	2	Y	Y
ZDT2	[60]	30	2	N	Y
ZDT3	[60]	30	2	N	Y
ZDT4	[60]	30	2	N	Y
ZDT6	[60]	10	2	N	Y
ZLT1	[33]	10	5	Y	N

<sup>a</sup> This is an adaptation of a single-objective optimization problem to the multiobjective setting that can be found in [45].

Table 1: List of test problems.

Given a multiobjective optimization problem, we are especially interested in estimating its Pareto frontier. Toward this goal, a strategy often used is to run the algorithm at hand from several different starting points. In view of this application, we considered 300 random starting points for each problem in Table 1. Each instance was seen as an independent problem and was solved by all algorithms. Figure 1 shows the results using performance profiles [20]. We compared the algorithms with respect to: (a) number of iterations; (b) CPU time; (c) number of objective function evaluations; (d) number of derivative evaluations.

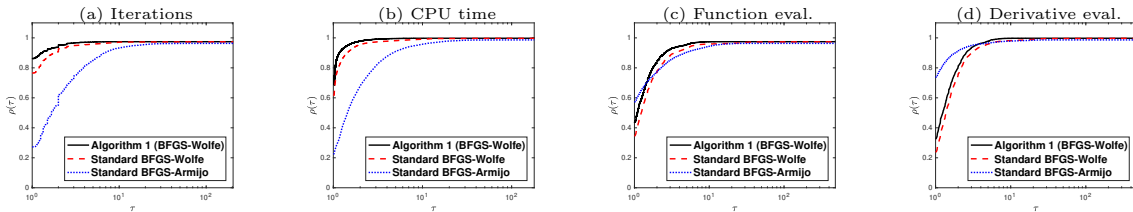


Figure 1: Performance profiles considering 300 starting points for each test problem using as the performance measurement: (a) number of iterations; (b) CPU time; (c) number of functions evaluations; (d) number of derivative evaluations.

We start our analysis by noting that all algorithms proved to be robust on the chosen set of test problems, which illustrates the practical capability of the BFGS method even for nonconvex

problems. Algorithm 1, Standard BFGS-Armijo and Standard BFGS-Wolfe algorithms successfully solved 99.8%, 99.7%, and 98.6% of the problem instances. Regarding efficiency, taking into account the number of iterations, Algorithm 1 (86.2%) had the best performance followed by Standard BFGS-Wolfe (76.5%) and Standard BFGS-Armijo (27.3%) algorithms, see Figure 1(a). This was directly reflected in CPU time (efficiency of 68.7%, 61.6%, and 22.6% for Algorithm 1, Standard BFGS-Wolfe algorithm, and Standard BFGS-Armijo algorithm, respectively), as seen in Figure 1(b). Concerning the number of function and derivative evaluations, the Standard BFGS-Armijo algorithm was the most efficient, see Figures 1(c) and (d). This was somewhat expected as the Wolfe line search uses more information from the objectives than the Armijo line search. Even so, the Standard BFGS-Armijo algorithm was quickly outperformed by the other algorithms in terms of the number of function evaluations. The strong correlation between the number of iterations and CPU time can be explained by the fact that, in our experiments, the computational cost is largely dominated by the solutions of the subproblems to calculate the search directions. In fact, Algorithm 1 and the Standard BFGS-Armijo algorithm spent, on average, 93.2% and 94.5% of the total CPU time on solving the subproblems, and only 2.0% and 0.7% on calculating the step sizes, respectively. Therefore, at least in our tests, the computational effort in the line searches can be neglected and the use of Wolfe step sizes is justified due to its impact on decreasing the number of iterations and, consequently, the CPU time. Another issue of interest concerns the length of step sizes calculated by the Wolfe and Armijo line search procedures. Figure 2 shows the histograms (normalized by relative probability) containing the frequency distribution of all step sizes calculated by Algorithm 1 and the Standard BFGS-Armijo algorithm. As can be seen, in 83.61% of the iterations, the Wolfe step size was greater than or equal to one. We point out that, due to the use of extrapolation strategies, even step sizes larger than one can be considered in the Wolfe line search procedure. A similar frequency (78.18%) for the unit step was observed for the Armijo step sizes. In contrast, step sizes smaller than 0.1 were observed in 8.66% and 18.28% for the Wolfe and Armijo line searches, respectively. This corroborates the discussion in Remark 3.1 about the Wolfe conditions preventing the method from taking excessively small step sizes when *larger* step sizes are possible.

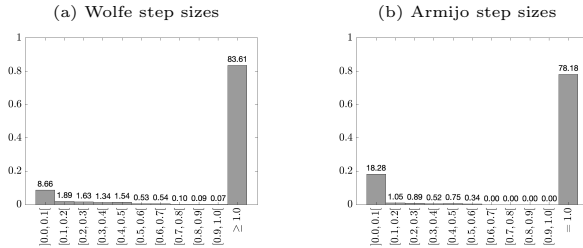


Figure 2: Histograms (normalized by relative probability) containing the frequency distribution of all step sizes calculated by: (a) Algorithm 1 (Wolfe step sizes); (b) Standard BFGS-Armijo algorithm (Armijo step sizes).

We now compare the ability of the solvers to properly generate Pareto frontiers. For that, we use the well-known *Purity* and ( $\Gamma$  and  $\Delta$ ) *Spread* metrics. Roughly speaking, given a solver and a problem, the *Purity* metric measures the ability of the solver to find points on the Pareto frontier of the problem, while a *Spread* metric seeks to measure the ability of the solver to obtain well-distributed points along the Pareto frontier. We refer the reader to [12] for a detailed explanation of these metrics and their uses along with performance profiles. The results in Figure 3 show that Algorithm 1 and the Standard BFGS-Wolfe algorithm outperformed the Standard BFGS-Armijo

algorithm in relation to the Purity metric, while no significant difference was observed for the Spread metrics.

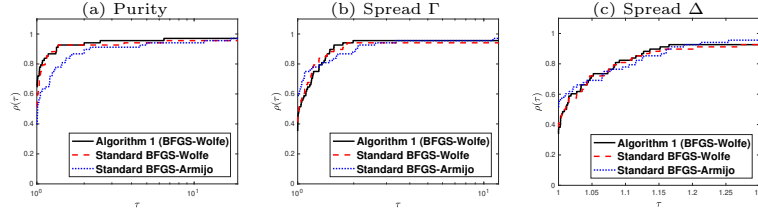


Figure 3: Metric performance profiles: (a) Purity; (b) Spread  $\Gamma$ ; (c) Spread  $\Delta$ .

Figure 4 shows the outline of the Image sets and Pareto frontiers obtained by Algorithm 1 for problems Hil1, KW2, MMR3, and MOP6. In the graphics, a full point represents a final iterate while the beginning of a straight segment represents the corresponding starting point. Pareto optimal points also have a square marker. As expected, since these problems are nonconvex, Algorithm 1 converges to some non-optimal Pareto critical points.

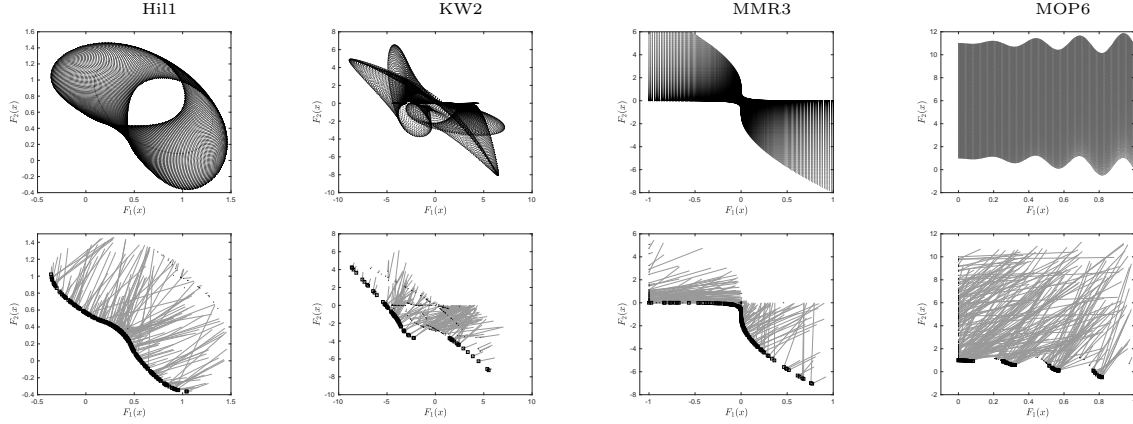


Figure 4: Image sets and Pareto frontiers obtained by Algorithm 1 for the nonconvex problems Hil1, KW2, MMR3, and MOP6.

Finally, we check the behavior of the methods as the number of objective functions increases. For this, we compare the performance of Algorithm 1 and the Standard BFGS-Armijo algorithm in larger instances of problems DTLZ1, DTLZ2, DTLZ3, DTLZ4, MGH26, Toi9, and Toi10. These are the customizable problems in dimension  $m$  of Table 1. MGH26, Toi9, and Toi10 are extensions of scalar optimization problems also known as *Trigonometric*, *Shifted TRIDIA*, and *Rosenbrock*, respectively. The first three columns of Table 2 identify the problem and the considered dimensions. For each instance, we ran both algorithms from 10 different random starting points. We emphasize that the algorithms reported convergence in all cases. The table gives the averages of: number of iterations (it), CPU time in seconds (time), number of function (nfev) and derivative (ngev) evaluations. The smallest reported data for each instance is highlighted in bold. We point out that we considered each evaluation of an objective (resp. objective gradient) in the calculation of nfev (resp. ngev). As can be seen, for MGH26 and DTLZ problems, Algorithm 1 strongly outperformed the Standard BFGS-Armijo algorithm with respect to the number of iterations and CPU time. Typically, taking into account these performance measures, Algorithm 1 uses less than half of the computational resources required by the Standard BFGS-Armijo

algorithm in this group of problems. Even with respect to the number of function and derivative evaluations, an advantage for Algorithm 1 was, in general, observed. Regarding Toi9 and Toi10 problems, the algorithms presented more homogeneous performances. While the Standard BFGS-Armijo algorithm was the most efficient in terms of function and derivative evaluations, Algorithm 1 always required fewer iterations, resulting in CPU time savings in four of the six instances.

Problem	$n$	$m$	Algorithm 1 (BFGS-Wolfe)				Standard BFGS-Armijo			
			it	time	nfev	ngev	it	time	nfev	ngev
DTLZ1	500	5	<b>6.5</b>	<b>31.1</b>	<b>84.1</b>	<b>71.7</b>	30.0	118.9	244.4	155.0
	500	8	<b>5.2</b>	<b>37.5</b>	<b>93.0</b>	<b>86.1</b>	10.7	57.3	134.4	93.6
	500	10	<b>2.9</b>	<b>14.4</b>	<b>73.0</b>	<b>68.2</b>	7.8	32.9	131.2	88.0
DTLZ2	500	5	<b>2.7</b>	<b>6.8</b>	<b>49.6</b>	<b>44.8</b>	10.9	36.2	123.8	59.5
	500	8	<b>3.1</b>	<b>11.0</b>	<b>84.0</b>	<b>73.6</b>	8.9	55.5	129.5	79.2
	500	10	<b>3.1</b>	<b>13.5</b>	<b>94.8</b>	<b>87.4</b>	9.1	68.2	167.9	101.0
DTLZ3	500	5	<b>4.7</b>	<b>6.7</b>	<b>57.4</b>	<b>50.1</b>	33.7	53.0	208.0	173.5
	500	8	<b>7.6</b>	<b>21.1</b>	<b>116.9</b>	<b>108.5</b>	81.0	287.1	719.4	656.0
	500	10	<b>5.8</b>	<b>17.6</b>	<b>113.6</b>	<b>105.9</b>	23.5	160.3	326.0	245.0
DTLZ4	500	5	<b>2.4</b>	<b>8.4</b>	54.1	48.7	4.2	14.2	<b>30.4</b>	<b>26.0</b>
	500	8	<b>2.9</b>	<b>13.4</b>	89.8	81.5	5.0	23.5	<b>58.2</b>	<b>48.0</b>
	500	10	<b>2.4</b>	<b>13.5</b>	<b>84.8</b>	76.0	6.0	33.5	98.5	<b>70.0</b>
MGH26	100	100	<b>14.8</b>	<b>4.4</b>	<b>3714.3</b>	<b>3339.6</b>	36.5	9.5	3750.0	3750.0
	200	200	<b>16.3</b>	<b>42.3</b>	<b>9739.5</b>	<b>8056.8</b>	49.1	122.9	10020.0	10020.0
	400	400	<b>21.6</b>	<b>480.2</b>	26987.7	<b>22211.8</b>	59.3	1298.9	<b>24120.0</b>	24120.0
Toi9	100	100	<b>4.5</b>	2.7	1014.5	868.8	5.4	<b>2.4</b>	<b>640.2</b>	<b>640.0</b>
	300	300	<b>4.2</b>	<b>63.4</b>	2387.2	2165.4	4.6	71.7	<b>1765.1</b>	<b>1680.0</b>
	400	400	<b>3.8</b>	<b>119.9</b>	2628.4	2442.9	4.7	134.2	<b>2370.4</b>	<b>2280.0</b>
Toi10	100	99	<b>11.3</b>	<b>9.1</b>	1895.7	1733.9	12.3	11.2	<b>1333.3</b>	<b>1316.7</b>
	200	199	<b>8.7</b>	85.8	3269.6	2937.9	10.8	<b>70.3</b>	<b>2421.6</b>	<b>2348.2</b>
	300	299	<b>10.2</b>	<b>282.9</b>	5357.8	<b>4796.8</b>	15.6	336.4	<b>4992.0</b>	4963.4

Table 2: Performance of Algorithm 1 (BFGS-Wolfe) and Standard BFGS-Armijo algorithm on larger instances of problems DTLZ1, DTLZ2, DTLZ3, DTLZ4, MGH26, Toi9, and Toi10.

## 7 Conclusions

In this work, we proposed a new BFGS update scheme for multiobjective optimization problems. This scheme generates positive definite Hessian approximations whenever the initial approximations are positive definite and the step sizes satisfy the Wolfe conditions. As a result, Algorithm 1 is well defined even for general nonconvex problems. As far as we know, this is the first BFGS-type algorithm designed to nonconvex multiobjective problems that updates the Hessian approximations at each iteration. We provided a comprehensive study of the main global and local convergence properties of the method, using assumptions that are natural extensions of those made for the scalar minimization case. Our numerical experiments suggest that the techniques used here potentially provide a non-negligible acceleration of the BFGS method. We hope that these techniques can also be useful for other variants of quasi-Newton methods for multiobjective optimization.

## Acknowledgements

This work was funded by FAPEG (Grants PPP03/15-201810267001725) and CNPq (Grants 424860/2018-0, 309628/2020-2, 405349/2021-1).



## References

- [1] M. A. Ansary and G. Panda. A modified Quasi-Newton method for vector optimization problem. *Optimization*, 64(11):2289–2306, 2015.
- [2] J. Bello Cruz, L. Lucambio Pérez, and J. Melo. Convergence of the projected gradient method for quasiconvex multiobjective optimization. *Nonlinear Anal.*, 74(16):5268 – 5273, 2011.
- [3] V. Bhaskar, S. K. Gupta, and A. K. Ray. Applications of multiobjective optimization in chemical engineering. *Rev. Chem. Eng.*, 16(1):1–54, 2000.
- [4] E. Birgin and J. Martinez. *Practical Augmented Lagrangian Methods for Constrained Optimization*. SIAM, Philadelphia, 2014.
- [5] H. Bonnel, A. N. Iusem, and B. F. Svaiter. Proximal methods in vector optimization. *SIAM J. Optim.*, 15(4):953–970, 2005.
- [6] R. H. Byrd and J. Nocedal. A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. *SIAM J. Numer. Anal.*, 26(3):727–739, 1989.
- [7] L. C. Ceng, B. S. Mordukhovich, and J. C. Yao. Hybrid approximate proximal method with auxiliary variational inequality for vector optimization. *J. Optimiz. Theory App.*, 146(2):267–303, 2010.
- [8] L. C. Ceng and J. C. Yao. Approximate proximal methods in vector optimization. *Eur. J. Oper. Res.*, 183(1):1–19, 2007.
- [9] T. D. Chuong. Generalized proximal method for efficient solutions in vector optimization. *Numer. Funct. Anal. Optim.*, 32(8):843–857, 2011.
- [10] T. D. Chuong. Newton-like methods for efficient solutions in vector optimization. *Comput. Optim. Appl.*, 54(3):495–516, 2013.
- [11] T. D. Chuong, B. S. Mordukhovich, and J. C. Yao. Hybrid approximate proximal algorithms for efficient solutions in vector optimization. *J. Nonlinear Convex Anal.*, 12(2):257–285, 8 2011.
- [12] A. L. Custódio, J. F. A. Madeira, A. I. F. Vaz, and L. N. Vicente. Direct multisearch for multiobjective optimization. *SIAM J. Optim.*, 21(3):1109–1140, 2011.
- [13] Y.-H. Dai. Convergence properties of the BFGS algorithm. *SIAM J. Optim.*, 13(3):693–701, 2002.
- [14] Y.-H. Dai. A perfect example for the BFGS method. *Math. Program.*, 138(1-2):501–530, 2013.
- [15] I. Das and J. Dennis. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J. Optim.*, 8(3):631–657, 1998.
- [16] W. C. Davidon. Variable metric methods for minimization, aec. *Research and Development Report, No. ANL-5990, Argonne Nat’l Lab., Argonne, Illinois*, 1959.

- [17] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multiobjective optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, pages 105–145. Springer, London, 2005.
- [18] J. E. Dennis and J. J. Moré. A characterization of superlinear convergence and its application to quasi-newton methods. *Math. Comp.*, 28(126):549–560, 1974.
- [19] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, 1996.
- [20] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91(2):201–213, 2002.
- [21] G. Eichfelder. *Adaptive Scalarization Methods in Multiobjective Optimization*. Springer Berlin Heidelberg, 2008.
- [22] N. S. Fazzio and M. L. Schuverdt. Convergence analysis of a nonmonotone projected gradient method for multiobjective optimization problems. *Optim. Lett.*, 13(6):1365–1379, 2019.
- [23] R. Fletcher and M. J. D. Powell. A Rapidly Convergent Descent Method for Minimization. *Comput. J.*, 6(2):163–168, 08 1963.
- [24] J. Fliege, L. M. Graña Drummond, and B. F. Svaiter. Newton’s method for multiobjective optimization. *SIAM J. Optim.*, 20(2):602–626, 2009.
- [25] J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Math. Methods Oper. Res.*, 51(3):479–494, 2000.
- [26] E. H. Fukuda and L. M. Graña Drummond. On the convergence of the projected gradient method for vector optimization. *Optimization*, 60(8-9):1009–1021, 2011.
- [27] E. H. Fukuda and L. M. Graña Drummond. Inexact projected gradient method for vector optimization. *Comput. Optim. Appl.*, 54(3):473–493, 2013.
- [28] M. L. N. Gonçalves and L. F. Prudente. On the extension of the Hager–Zhang conjugate gradient method for vector optimization. *Comput. Optim. Appl.*, 76(3):889–916, 2020.
- [29] L. M. Graña Drummond and A. N. Iusem. A projected gradient method for vector optimization problems. *Comput. Optim. Appl.*, 28(1):5–29, 2004.
- [30] L. M. Graña Drummond, F. M. P. Raupp, and B. F. Svaiter. A quadratically convergent Newton method for vector optimization. *Optimization*, 63(5):661–677, 2014.
- [31] L. M. Graña Drummond and B. F. Svaiter. A steepest descent method for vector optimization. *J. Comput. Appl. Math.*, 175(2):395–414, 2005.
- [32] C. Hillermeier. Generalized homotopy approach to multiobjective optimization. *J. Optimiz. Theory App.*, 110(3):557–583, 2001.
- [33] S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE T. Evolut. Comput.*, 10(5):477–506, 2006.

- [34] Y. Jin, M. Olhofer, and B. Sendhoff. Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how? In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, GECCO'01, page 1042–1049, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [35] I. Kim and O. de Weck. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Struct. Multidiscip. O.*, 29(2):149–158, Feb 2005.
- [36] K. K. Lai, S. K. Mishra, and B. Ram. On q-quasi-Newton's method for unconstrained multiobjective optimization problems. *Mathematics*, 8(4), 2020.
- [37] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evol. Comput.*, 10(3):263–282, 2002.
- [38] D.-H. Li and M. Fukushima. On the global convergence of the bfgs method for nonconvex unconstrained optimization problems. *SIAM J. Optim.*, 11(4):1054–1064, 2001.
- [39] A. Lovison. Singular continuation: Generating piecewise linear approximations to Pareto sets via global analysis. *SIAM J. Optim.*, 21(2):463–490, 2011.
- [40] L. R. Lucambio Pérez and L. F. Prudente. Nonlinear conjugate gradient methods for vector optimization. *SIAM J. Optim.*, 28(3):2690–2720, 2018.
- [41] L. R. Lucambio Pérez and L. F. Prudente. A Wolfe line search algorithm for vector optimization. *ACM Trans. Math. Softw.*, 45(4):23, 2019.
- [42] N. Mahdavi-Amiri and F. Salehi Sadaghiani. A superlinearly convergent nonmonotone quasi-newton method for unconstrained multiobjective optimization. *Optim. Method Softw.*, 35(6):1223–1247, 2020.
- [43] W. F. Mascarenhas. The BFGS method with exact line searches fails for non-convex objective functions. *Math. Program.*, 99(1):49–61, 2004.
- [44] E. Miglierina, E. Molho, and M. Recchioni. Box-constrained multi-objective optimization: A gradient-like method without a priori scalarization. *Eur. J. Oper. Res.*, 188(3):662–682, 2008.
- [45] K. Mita, E. H. Fukuda, and N. Yamashita. Nonmonotone line searches for unconstrained multiobjective optimization problems. *J. Global Optim.*, 75:63–90, 2019.
- [46] J. J. Moré, B. S. Garbow, and K. E. Hillstom. Testing unconstrained optimization software. *ACM Trans. Math. Softw.*, 7(1):17–41, Mar. 1981.
- [47] V. Morovati, H. Basirzadeh, and L. Pourkarimi. Quasi-newton methods for multiobjective optimization problems. *4OR*, 16(3):261–294, Nov. 2017.
- [48] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.
- [49] Z. Povalej. Quasi-Newton method for multiobjective optimization. *J. Comput. Appl. Math.*, 255:765–777, 2014.

- [50] M. J. Powell. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In *Nonlinear programming, SIAM-AMS proceedings*, volume 9, 1976.
- [51] M. Preuss, B. Naujoks, and G. Rudolph. Pareto set and EMOA behavior for simple multimodal multiobjective functions. In T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, editors, *Parallel Problem Solving from Nature - PPSN IX*, pages 513–522, Berlin, Heidelberg, 2006. Springer.
- [52] S. Qu, M. Goh, and F. T. Chan. Quasi-Newton methods for solving multiobjective optimization. *Oper. Res. Lett.*, 39(5):397–399, 2011.
- [53] S. Qu, C. Liu, M. Goh, Y. Li, and Y. Ji. Nonsmooth multiobjective programming with quasi-Newton methods. *Eur. J. Oper. Res.*, 235(3):503 – 510, 2014.
- [54] O. Schütze, M. Laumanns, C. A. Coello Coello, M. Dellnitz, and E.-G. Talbi. Convergence of stochastic search algorithms to finite size Pareto set approximations. *J. Global Optim.*, 41(4):559–577, 2008.
- [55] W. Stadler and J. Dauer. Multicriteria optimization in engineering: A tutorial and survey. In *Structural Optimization: Status And Promise*, chapter 10, pages 209–249. American Institute of Aeronautics and Astronautics, 1992.
- [56] T. Stewart, O. Bandte, H. Braun, N. Chakraborti, M. Ehrgott, M. Göbelt, Y. Jin, H. Nakayama, S. Poles, and D. Di Stefano. Real-world applications of multiobjective optimization. In J. Branke, K. Deb, K. Miettinen, and R. Słowiński, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 285–327. Springer, Berlin, Heidelberg, 2008.
- [57] B. F. Svaiter. The multiobjective steepest descent direction is not Lipschitz continuous, but is Hölder continuous. *Oper. Res. Lett.*, 46(4):430–433, 2018.
- [58] P. L. Toint. Test problems for partially separable optimization and results for the routine PSPMIN. *The University of Namur, Department of Mathematics, Belgium, Tech. Rep.*, 1983.
- [59] J. Wang, Y. Hu, C. K. Wai Yu, C. Li, and X. Yang. Extended Newton methods for multiobjective optimization: majorizing function technique and convergence analysis. *SIAM J. Optim.*, 29(3):2388–2421, 2019.
- [60] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, 8(2):173–195, 2000.