

# A Wolfe line search algorithm for vector optimization\*

L. R. Lucambio Pérez<sup>†</sup>

L. F. Prudente<sup>†</sup>

July 27, 2018

## Abstract

In a recent paper, Lucambio Pérez and Prudente extended the Wolfe conditions for the vector-valued optimization. Here, we propose a line search algorithm for finding a step-size satisfying the strong Wolfe conditions in the vector optimization setting. Well definiteness and finite termination results are provided. We discuss practical aspects related to the algorithm and present some numerical experiments illustrating its applicability. Codes supporting this paper are written in Fortran 90 and are freely available for download.

**Keywords:** line search algorithm; Wolfe conditions; vector optimization.

**Mathematical Subject Classification(2010):** 90C29 90C52

## 1 Introduction

In the vector optimization, several objectives have to be minimized simultaneously with respect to a partial order induced by a closed convex pointed cone with non-empty interior. The particular case where the cone is the so-called Pareto cone, i.e. the positive orthant, corresponds to the multiobjective optimization. Many practical models in different areas seek to solve multiobjective and vector optimization problems, see for example [14, 19, 20, 26, 29, 31, 42, 43, 45]. Recently, the extension of iterative methods for scalar-valued to multiobjective and vector-valued optimization has received considerable attention from the optimization community. Some works dealing with this subject include the steepest descent and projected gradient [13, 18, 21–23, 25, 35], conjugate gradient [34], Newton [7, 11, 17, 24, 32], Quasi-Newton [3, 39, 40], subgradient [4], interior point [44], and proximal methods [6, 8–10, 12].

Line search and trust region techniques are essential components to globalize nonlinear descent optimization algorithms. Concepts such as Armijo and Wolfe conditions are the basis for a broad class of minimization methods that employ line search strategies. Hence, the development of efficient line search procedures that satisfy these kind of conditions is a crucial ingredient for the formulation of several practical algorithms for optimization. This is a well-explored topic in the classical optimization approach, see [1, 15, 16, 27, 28, 30, 36, 37]. In particular, we highlight the paper [37] where Moré and Thuente proposed an algorithm designed to find a step-size satisfying the strong scalar-valued Wolfe conditions. On the other hand, the Wolfe conditions were extended to the vector optimization setting only recently in [34]. In this work, the authors also introduced the Zoutendijk condition for vector optimization and proved that it holds for a general descent line search method that satisfies the vector-valued Wolfe conditions. The Zoutendijk condition is a powerful tool for analyzing the convergence of line search methods. These extensions opened perspectives for the formulation of new algorithms for vector optimization problems.

---

\*This work was partially supported by CAPES and FAPEG.

<sup>†</sup>Instituto de Matemática e Estatística, Universidade Federal de Goiás, Avenida Esperança, s/n, Campus Samambaia, Goiânia, GO - 74690-900, Brazil (E-mails: [lr1p@ufg.br](mailto:lr1p@ufg.br), [lfprudente@ufg.br](mailto:lfprudente@ufg.br)).

In the present paper, we take a step beyond [34] from the perspective of actual implementations of line search strategies. We propose a line search algorithm for finding a step-size satisfying the strong vector-valued Wolfe conditions. At each iteration, our algorithm works with a scalar function and uses an inner solver designed to find a step-size satisfying the strong scalar-valued Wolfe conditions. In the multiobjective optimization case, such scalar function corresponds to one of the objectives. Assuming that the inner solver stops in finite time, we show that the proposed algorithm is well defined and terminates its execution in a finite number of (outer) iterations. In our implementation, we use the algorithm of Moré and Thuente as the inner solver. Its finite termination is guaranteed except in pathological cases, see [37]. It should be mentioned that an early version of the algorithm proposed here was used in the numerical experiments of [34], where a family of conjugate gradients methods for vector optimization problems was introduced and tested. We present also a theoretical result: using the convergence result of our algorithm, we improve the theorem in [34] about the existence of stepsizes satisfying the vector-valued Wolfe conditions. We also discuss some practical aspects related to the algorithm, which support our implementation. Numerical experiments illustrating the applicability of the algorithm are presented. Our codes are written in Fortran 90, and are freely available for download at <https://lfprudente.mat.ufg.br/>.

This paper is organized as follows. In Section 2 we provide some useful definitions, notations and preliminary results. In Section 3 we describe the Wolfe line search algorithm for vector optimization and present its convergence analysis. Section 4 is devoted to practical aspects related to the algorithm, which support our implementation. Section 5 describes the numerical experiments. Finally we make some comments about this work in Section 6.

**Notation.** The symbol  $\langle \cdot, \cdot \rangle$  is the usual inner product in  $\mathbb{R}^m$  and  $\| \cdot \|$  denotes the Euclidean norm. If  $\mathbb{K} = (k_1, k_2, \dots) \subseteq \mathbb{N}$  (with  $k_j < k_{j+1}$  for all  $j$ ), we denote  $\mathbb{K} \subseteq_{\infty} \mathbb{N}$ .

## 2 Preliminaries

A closed, convex, and pointed cone  $K \subset \mathbb{R}^m$  with non-empty interior induces a partial order in  $\mathbb{R}^m$  as following

$$u \preceq_K v \text{ (} u \prec_K v \text{) if and only if } v - u \in K \text{ (} v - u \in \text{int}(K)\text{),}$$

where  $\text{int}(K)$  denotes the interior of  $K$ . In vector optimization, given a function  $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$ , we seek to find a  $K$ -efficient or  $K$ -optimal point of  $F$ , that is, a point  $x \in \mathbb{R}^n$  for which there is no other  $y \in \mathbb{R}^n$  with  $F(y) \preceq_K F(x)$  and  $F(y) \neq F(x)$ . We denote this problem by

$$\text{Minimize}_K F(x), \quad x \in \mathbb{R}^n. \tag{1}$$

When  $F$  is continuously differentiable, a necessary condition for  $K$ -optimality of  $x \in \mathbb{R}^n$  is

$$-\text{int}(K) \cap \text{Image}(JF(x)) = \emptyset, \tag{2}$$

where  $JF(x)$  is the Jacobian of  $F$  at  $x$  and  $\text{Image}(JF(x))$  denotes the image on  $\mathbb{R}^m$  by  $JF(x)$ . A point  $x$  satisfying (2) is called  $K$ -critical. If  $x$  is not  $K$ -critical, then there is  $d \in \mathbb{R}^n$  such that  $JF(x)d \in -\text{int}(K)$ . In this case, it is possible to show that  $d$  is a  $K$ -descent direction for  $F$  at  $x$ , i.e., there exists  $T > 0$  such that  $0 < t < T$  implies that  $F(x + td) \prec_K F(x)$ , see [33].

The positive polar cone of  $K$  is defined by  $K^* := \{w \in \mathbb{R}^m \mid \langle w, y \rangle \geq 0, \forall y \in K\}$ . Let  $C \subset K^* \setminus \{0\}$  be a compact set such that

$$K^* = \text{cone}(\text{conv}(C)), \tag{3}$$

where  $\text{conv}(A)$  denotes the convex hull of a set  $A \subset \mathbb{R}^m$ , and  $\text{cone}(A)$  is the cone generated by  $A$ . When  $K$  is polyhedral,  $K^*$  is also polyhedral and the finite set of its extremal rays is

such that (3) holds. For example, in multiobjective optimization, where  $K = \mathbb{R}_+^m$ , we have  $K^* = K$  and  $C$  can be defined as the canonical basis of  $\mathbb{R}^m$ . For a generic cone  $K$ , we may take  $C = \{w \in K^* : \|w\| = 1\}$ .

Assume that  $F$  is continuously differentiable. Define  $\varphi: \mathbb{R}^m \rightarrow \mathbb{R}$  by

$$\varphi(y) := \sup \{ \langle w, y \rangle \mid w \in C \},$$

and  $f: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  by

$$f(x, d) := \varphi(JF(x)d) = \sup \{ \langle w, JF(x)d \rangle \mid w \in C \}.$$

Since  $C$  is compact,  $\varphi$  and  $f$  are well-defined. Function  $\varphi$  gives characterizations of  $-K$  and  $-\text{int}(K)$ , because

$$-K = \{y \in \mathbb{R}^m \mid \varphi(y) \leq 0\} \quad \text{and} \quad -\text{int}(K) = \{y \in \mathbb{R}^m \mid \varphi(y) < 0\}.$$

In its turn,  $f$  gives a characterization of  $K$ -descent directions of  $F$  at  $x$ , in the sense that  $d$  is a  $K$ -descent direction for  $F$  at  $x$  if and only if  $f(x, d) < 0$ , see [25].

Let  $d \in \mathbb{R}^n$  be a  $K$ -descent direction for  $F$  at  $x$ , and  $C$  as in (3). It is said that  $\alpha > 0$  satisfies the standard Wolfe conditions if

$$\begin{aligned} F(x + \alpha d) &\preceq_K F(x) + \rho \alpha f(x, d)e, \\ f(x + \alpha d, d) &\geq \sigma f(x, d), \end{aligned} \tag{4}$$

where  $0 < \rho < \sigma < 1$  and  $e \in K$  is a vector such that

$$0 < \langle w, e \rangle \leq 1 \text{ for all } w \in C. \tag{5}$$

The strong Wolfe conditions require  $\alpha > 0$  to satisfy

$$\begin{aligned} F(x + \alpha d) &\preceq_K F(x) + \rho \alpha f(x, d)e, \\ |f(x + \alpha d, d)| &\leq \sigma |f(x, d)|, \end{aligned} \tag{6a}$$

$$\tag{6b}$$

see [34, Definition 3.1]. It is easy to see that if  $\alpha > 0$  satisfies the strong Wolfe conditions, then the standard Wolfe conditions also holds for  $\alpha$ .

Observe that  $e := \tilde{e}/\gamma$ , where  $\tilde{e}$  is any vector belonging to  $\text{int}(K)$  and  $\gamma$  is the maximum value of the continuous linear function  $\langle w, \tilde{e} \rangle$  over the compact set  $C$ , fulfills (5), see [34]. For multiobjective optimization, where  $K = \mathbb{R}_+^m$  and  $C$  is given by the canonical basis of  $\mathbb{R}^m$ , we may take  $e = [1, \dots, 1]^T \in \mathbb{R}^m$ . It is worth mentioning that if  $\alpha > 0$  and

$$\langle w, F(x + \alpha d) \rangle \leq \langle w, F(x) \rangle + \alpha \rho f(x, d) \quad \text{for all } w \in C,$$

then  $\alpha$  satisfies condition (6a) for any  $e \in K$  such that (5) holds.

In [34, Proposition 3.2] the authors showed that if  $C$  is finite and  $F$  is bounded below, in the sense that there exists  $\mathcal{F} \in \mathbb{R}^m$  such that

$$F(x + \alpha d) \succeq_K \mathcal{F} \quad \text{for all } \alpha > 0,$$

then there exist intervals of positive stepsizes satisfying the standard and strong Wolfe conditions. In the next section we will improve this result by showing that the boundedness of  $\langle w, F(x + \alpha d) \rangle$ , for one  $w \in C$ , is sufficient to guarantee the existence of stepsizes satisfying the Wolfe conditions.

Now consider a general line search method for vector optimization

$$x^{\ell+1} = x^\ell + \alpha_\ell d^\ell, \quad \ell \geq 0 \tag{7}$$

where  $d^\ell$  is a  $K$ -descent direction for  $F$  at  $x^\ell$ , and the step-size  $\alpha_\ell$  satisfies the standard Wolfe conditions (4). Under mild assumptions, the general method (7) satisfies the vector Zoutendijk condition given by

$$\sum_{\ell \geq 0} \frac{f^2(x^\ell, d^\ell)}{\|d^\ell\|^2} < \infty,$$

see [34, Proposition 3.3]. This condition can be used to derive convergence results for several line search algorithms. Particularly, in [34], the concepts of Wolfe and Zoutendijk conditions were widely used in the study of non-linear conjugate gradient methods in the vector optimization setting.

### 3 The Wolfe line search algorithm

Hereafter, we assume that  $F$  is continuous differentiable,  $d \in \mathbb{R}^n$  is a  $K$ -descent direction for  $F$  at  $x$ , and  $C := \{w_1, \dots, w_p\} \subset K^* \setminus \{0\}$  is such that

$$K^* = \text{cone}(\text{conv}(\{w_1, \dots, w_p\})).$$

Let  $\mathcal{I} := \{1, \dots, p\}$  be the set of indexes associated with the elements of  $C$ . Define  $\phi_i: \mathbb{R} \rightarrow \mathbb{R}$  by

$$\phi_i(\alpha) := \langle w_i, F(x + \alpha d) \rangle, \quad (8)$$

for all  $i \in \mathcal{I}$ , and

$$\phi'_{\max}(0) := \max\{\phi'_i(0) : i \in \mathcal{I}\}.$$

Observe that  $f(x + \alpha d, d) = \max\{\phi'_i(\alpha) : i \in \mathcal{I}\}$ , and so  $\phi'_{\max}(0) = f(x, d)$ .

Our algorithm seeks to find a step-size  $\alpha > 0$  such that

$$\phi_i(\alpha) \leq \phi_i(0) + \alpha \rho \phi'_{\max}(0) \quad \text{for all } i \in \mathcal{I}, \quad (9a)$$

$$|\max\{\phi'_i(\alpha) : i \in \mathcal{I}\}| \leq -\sigma \phi'_{\max}(0). \quad (9b)$$

It is easy to verify that any  $\alpha > 0$  satisfying conditions (9) also satisfies the strong Wolfe conditions (6) with any  $e \in K$  such that (5) holds.

The fluxogram of Algorithm 1 in Figure 1 helps to understand the structure of the line search procedure.

Let us explain the main steps of Algorithm 1. The main reason to require an upper bound  $\alpha_{\max} > 0$  to the trial stepsizes is to guarantee finite termination in the presence of unbounded below functions  $\phi_i$ . At Step 1.1, Algorithm 1 tests whether  $\alpha_k$  satisfies conditions (9). If this is the case, the algorithm stops with  $\alpha = \alpha_k$ , declaring convergence. This is the stopping criterion related to success, where a step-size satisfying the strong Wolfe conditions is found. The second stopping criterion, at Step 1.2, tests for termination at  $\alpha_{\max}$ . Algorithm 1 stops at  $\alpha_k = \alpha_{\max}$  if conditions (10) hold for all  $i \in \mathcal{I}$ . This is a reasonable stopping criterion because, as we will show, it is possible to determine an acceptable step-size when these conditions do not hold. In this case, only condition (9a) is verified at  $\alpha$  and Algorithm 1 finishes with a warning message.

The boolean variable `brackt` indicates whether the working interval contains stepsizes satisfying conditions (9). If `brackt = true` at iteration  $k$ , then it turns out that  $(0, \alpha_k)$  brackets desired stepsizes. On the other hand, `brackt = false` means that there is no guarantee that  $(0, \alpha_k)$  contains an acceptable step-size. Test (11) at Step 2 is the key for defining `brackt`. As we will see, once `brackt` is set equal to `true` for the first time, it no longer changes and the algorithm always executes Step 2. Variable `brackt` also controls the flow of Algorithm 1.

Basically, Algorithm 1 works in two stages. The first stage corresponds to a *bracketing phase* where the algorithm seeks to identify an interval containing desirable stepsizes. Given parameter  $\alpha_{\max} > 0$  and  $\alpha_0 \in (0, \alpha_{\max})$ , it initializes with `brackt = false` and keeps increasing

---

**Algorithm 1:** The Wolfe line search algorithm for vector optimization

---

Let  $0 < \rho < \bar{\rho} < \bar{\sigma} < \sigma < 1$ ,  $\delta > 1$ ,  $\mathcal{I} = \{1, \dots, p\}$ , and  $\alpha_{\max} > 0$ .

**Step 0.** *Initialization*

Choose  $\alpha_0 \in (0, \alpha_{\max})$ . Set **brackt**  $\leftarrow$  **false**, and initialize  $k \leftarrow 0$ .

**Step 1.** *Stopping criteria*

- 1.1. If  $\phi_i(\alpha_k) \leq \phi_i(0) + \alpha_k \rho \phi'_{\max}(0)$  for all  $i \in \mathcal{I}$ , and  $|\max\{\phi'_i(\alpha_k) : i \in \mathcal{I}\}| \leq -\sigma \phi'_{\max}(0)$ , then **stop** with  $\alpha = \alpha_k$  declaring **convergence**.
- 1.2. If  $\alpha_k = \alpha_{\max}$ ,

$$\phi_i(\alpha_{\max}) \leq \phi_i(0) + \alpha_{\max} \rho \phi'_{\max}(0) \quad \text{and} \quad \phi'_i(\alpha_{\max}) < \sigma \phi'_{\max}(0), \quad (10)$$

for all  $i \in \mathcal{I}$ , then **stop** with  $\alpha = \alpha_{\max}$  declaring **warning**.

**Step 2.** *Solving the subproblem*

If there exists  $i \in \mathcal{I}$  such that

$$\phi_i(\alpha_k) > \phi_i(0) + \alpha_k \rho \phi'_{\max}(0) \quad \text{or} \quad \phi'_i(\alpha_k) > -\sigma \phi'_{\max}(0), \quad (11)$$

then:

2.1. **brackt**  $\leftarrow$  **true**;

2.2.  $\alpha_{\max} \leftarrow \alpha_k$ ;

2.3. using an inner solver, compute  $\alpha_{k+1} \in (0, \alpha_{\max})$  satisfying

$$\phi_{i_k}(\alpha_{k+1}) \leq \phi_{i_k}(0) + \alpha_{k+1} \bar{\rho} \phi'_{\max}(0) \quad \text{and} \quad \left| \phi'_{i_k}(\alpha_{k+1}) \right| \leq -\bar{\sigma} \phi'_{\max}(0), \quad (12)$$

where  $i_k \in \mathcal{I}$  is an index such that (11) holds.

**Step 3.** *Trying to find an interval that brackets desired stepsizes*

If **brackt** = **false**, then choose  $\alpha_{k+1} \in [\min\{\delta \alpha_k, \alpha_{\max}\}, \alpha_{\max}]$ .

**Step 4.** *Beginning a new outer iteration*

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

---

the trial step-size until either  $\alpha_{k_0}$ , for some  $k_0 \geq 0$ , satisfies conditions (9) or it is identified that interval  $(0, \alpha_{k_0})$  brackets desired stepsizes. In this process, we require that the upper limit  $\alpha_{\max}$  be used as a trial value in a finite number of iterations. In this first stage, **brackt** is always equal to **false** and Algorithm 1 executes Step 3. Once it has been identified that interval the  $(0, \alpha_{k_0})$  brackets a desirable step-size, variable **brackt** is set equal to **true** and the algorithm enters the second stage, called *selection phase*. In this phase, it always executes Step 2 and, thereafter, the generated sequence  $\{\alpha_k\}_{k \geq k_0}$  is decreasing. At Step 2.3, Algorithm 1 works only with one function  $\phi_{i_k}$  for some  $i_k \in \mathcal{I}$ , and the subproblem consists of calculating  $\alpha_{k+1} \in (0, \alpha_k)$  satisfying (12). If  $\alpha_{k+1}$  does not satisfy conditions (9), then it is possible to show that  $(0, \alpha_{k+1})$  also contains desirable stepsizes, becoming the new working interval.

Note that (12) are the strong Wolfe conditions at  $\alpha_{k+1}$  for the scalar function  $\phi_{i_k}$ , with the auxiliary parameters  $0 < \bar{\rho} < \bar{\sigma} < 1$  and  $\phi'_{\max}(0)$  instead of  $\phi'_{i_k}(0)$ , see for example [38]. In practical implementations, the calculation of  $\alpha_{k+1}$  at Step 2.3 is performed by means of an iterative scheme. In this section, we assume that the inner solver is able to compute  $\alpha_{k+1} \in (0, \alpha_k)$  in a finite number of (inner) steps. In this case, we say that the inner solver has the

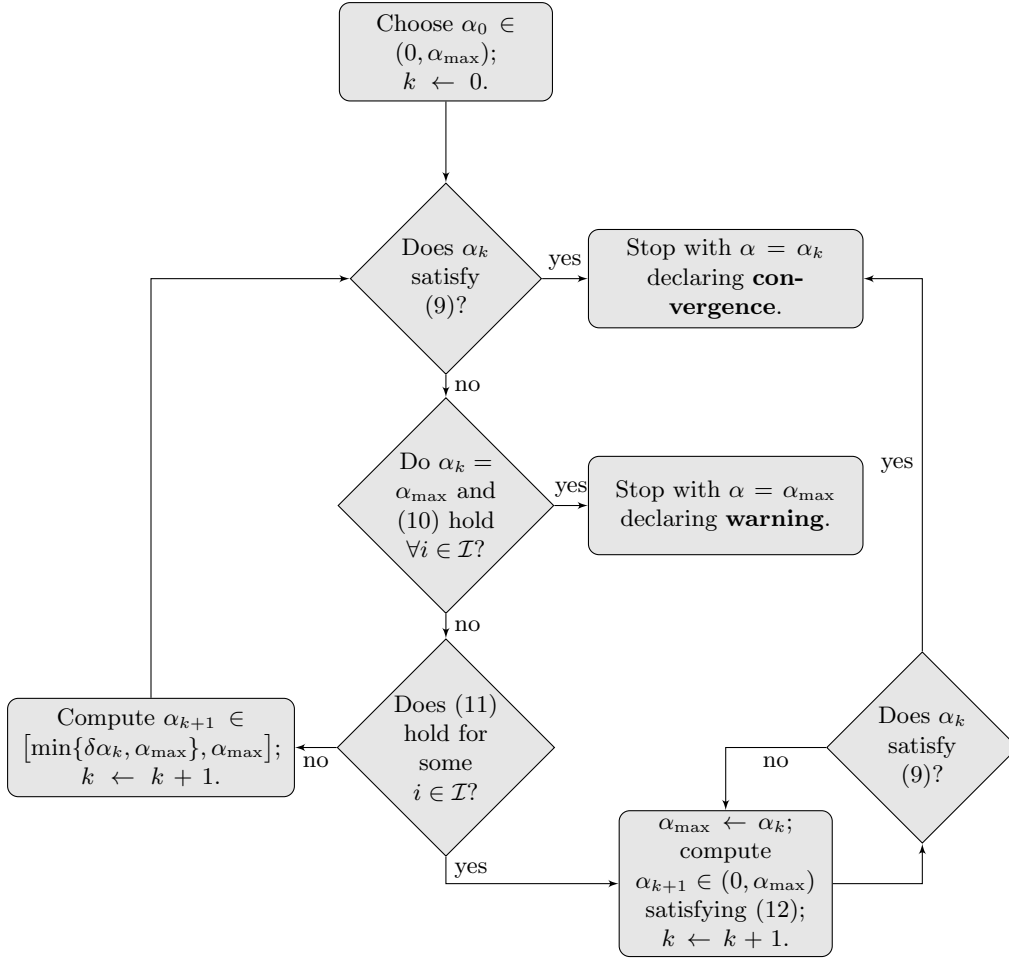


Figure 1: Fluxogram of Algorithm 1.

*finite termination property.* We will come back to this issue in Section 4 and we will see that this property is verified, for example, except in pathological cases, for the well-known algorithm of Moré and Thuente [37].

Lemma 1 establishes that condition (9a) is satisfied for small enough stepsizes.

**Lemma 1.** *There exists  $\Lambda > 0$  such that*

$$\phi_i(\alpha) \leq \phi_i(0) + \alpha \rho \phi'_{\max}(0) \quad \text{for all } i \in \mathcal{I}, \quad (13)$$

for all  $\alpha \in (0, \Lambda)$ .

*Proof.* Given  $i \in \mathcal{I}$ , it is well known that there exists  $\Lambda_i > 0$  such that

$$\phi_i(\alpha) \leq \phi_i(0) + \alpha \rho \phi'_i(0),$$

for all  $\alpha \in (0, \Lambda_i)$ , see for example [5]. Since  $\phi'_{\max}(0) \geq \phi'_i(0)$ , it follows that

$$\phi_i(\alpha) \leq \phi_i(0) + \alpha \rho \phi'_{\max}(0),$$

for all  $\alpha \in (0, \Lambda_i)$ . Define  $\Lambda := \min\{\Lambda_i : i \in \mathcal{I}\}$ . Then, (13) clearly holds for all  $\alpha \in (0, \Lambda)$ , completing the proof.  $\square$

The following lemma is connected with the solvability of the subproblems at Step 2.3. In particular, Lemma 2 implies that it is possible to compute  $\alpha_{k+1} \in (0, \alpha_k)$  satisfying (12), provided that (11) holds.

**Lemma 2.** Let  $i \in \mathcal{I}$  and  $\beta > 0$ . If

$$\phi_i(\beta) > \phi_i(0) + \beta\rho\phi'_{\max}(0) \quad \text{or} \quad \phi'_i(\beta) > -\sigma\phi'_{\max}(0), \quad (14)$$

then there exist  $\alpha_* \in (0, \beta)$  and a neighborhood  $V$  of  $\alpha_*$  such that

$$\phi_i(\alpha) \leq \phi_i(0) + \alpha\bar{\rho}\phi'_{\max}(0) \quad \text{and} \quad |\phi'_i(\alpha)| \leq -\bar{\sigma}\phi'_{\max}(0), \quad (15)$$

for all  $\alpha \in V$ .

*Proof.* Assume that (14) holds. Then,

$$\phi_i(\beta) > \phi_i(0) + \beta\bar{\rho}\phi'_{\max}(0) \quad \text{or} \quad \phi'_i(\beta) > -\bar{\rho}\phi'_{\max}(0),$$

because  $0 < \rho < \bar{\rho} < \bar{\sigma} < \sigma$ , and  $\phi'_{\max}(0) < 0$ . Let us define the auxiliary function  $\psi : \mathbb{R} \rightarrow \mathbb{R}$  by

$$\psi(\alpha) := \phi_i(\alpha) - \phi_i(0) - \alpha\bar{\rho}\phi'_{\max}(0).$$

Suppose that  $\phi_i(\beta) \geq \phi_i(0) + \beta\bar{\rho}\phi'_{\max}(0)$  or, equivalently,  $\psi(\beta) \geq 0$ . Since  $\psi(0) = 0$  and  $\psi'(0) < 0$ , by continuity arguments, we have  $\psi(\alpha) < 0$  for  $\alpha > 0$  sufficiently small. Define  $\tilde{\alpha} \in (0, \beta]$  by

$$\tilde{\alpha} := \min \{ \alpha \in (0, \beta] : \psi(\alpha) = 0 \}.$$

Note that, by the intermediate value theorem,  $\tilde{\alpha}$  is well defined. We clearly have  $\psi(\alpha) < 0$  for  $\alpha \in (0, \tilde{\alpha})$ . Since  $\psi(0) = \psi(\tilde{\alpha}) = 0$ , by the mean value theorem, there is  $\alpha_* \in (0, \tilde{\alpha})$  such that  $\psi'(\alpha_*) = 0$ . Hence,

$$\phi_i(\alpha_*) < \phi_i(0) + \alpha_*\bar{\rho}\phi'_{\max}(0)$$

and  $\phi'_i(\alpha_*) = \bar{\rho}\phi'_{\max}(0)$ . Thus,

$$|\phi'_i(\alpha_*)| < -\bar{\sigma}\phi'_{\max}(0),$$

because  $\phi'_{\max}(0) < 0$  and  $\bar{\rho} < \bar{\sigma}$ . Therefore, by continuity arguments, there exists a neighborhood  $V$  of  $\alpha_*$  such that (15) holds for all  $\alpha \in V$ .

Now consider that  $\phi'_i(\beta) > -\bar{\rho}\phi'_{\max}(0)$  or, equivalently,  $\psi'(\beta) > 0$ . Since  $\psi'(0) < 0$ , by the intermediate value theorem, there is  $\hat{\alpha} \in (0, \beta)$  such that  $\psi'(\hat{\alpha}) = 0$ . Thus,  $\phi'_i(\hat{\alpha}) = \bar{\rho}\phi'_{\max}(0)$ . Hence,

$$|\phi'_i(\hat{\alpha})| < -\bar{\sigma}\phi'_{\max}(0)$$

because  $\phi'_{\max}(0) < 0$  and  $\bar{\rho} < \bar{\sigma}$ . If  $\phi_i(\hat{\alpha}) < \phi_i(0) + \hat{\alpha}\bar{\rho}\phi'_{\max}(0)$ , then the thesis holds for  $\alpha_* = \hat{\alpha}$ . Otherwise, we can proceed as in the first part of the proof to obtain  $\alpha_* \in (0, \hat{\alpha})$  for which (15) holds.  $\square$

Next we show the well definedness of Algorithm 1. We prove that if Algorithm 1 does not stop at Step 1, then one and only one step between Step 2 and Step 3 is executed. Furthermore, we show in Theorem 1 that, once Step 2 is executed and `brackt` is set equal to `true` for the first time, the algorithm also executes Step 2 in the subsequent iterations.

**Theorem 1.** Assume that the inner solver has the finite termination property. Then, Algorithm 1 is well defined.

*Proof.* We first observe that it is possible to perform Steps 2 and 3. Algorithm 1 executes Step 2 in iteration  $k$  when (11) holds for some  $i \in \mathcal{I}$ . In this case, by Lemma 2 and the finite termination property of the inner solver, Algorithm 1 finds  $\alpha_{k+1} \in (0, \alpha_k)$  satisfying (12). At Step 3, we simply define  $\alpha_{k+1}$  belonging to the interval  $[\min\{\delta\alpha_k, \alpha_{\max}\}, \alpha_{\max}]$ . In this last case we obtain  $\alpha_{k+1} > \alpha_k$ , because  $\delta > 1$ .

We claim that if there exists  $k_0 \in \mathbb{N}$  for which (11) holds for  $\alpha_{k_0}$ , then for  $k > k_0$  either the algorithm stops in iteration  $k$  or (11) also holds at  $\alpha_k$ . This means that, once `brackt` is set

equal to `true` for the first time, the algorithm always executes Step 2. Assume that (11) holds at  $\alpha_k$ . Thus, Algorithm 1 finds  $\alpha_{k+1}$  satisfying (12). If the algorithm does not stop at  $\alpha_{k+1}$ , by Step 1.1, we obtain

$$\phi_i(\alpha_{k+1}) > \phi_i(0) + \alpha_{k+1}\rho\phi'_{\max}(0) \quad \text{for some } i \in \mathcal{I} \quad (16)$$

or

$$|\max\{\phi'_i(\alpha_{k+1}) : i \in \mathcal{I}\}| > -\sigma\phi'_{\max}(0). \quad (17)$$

It is straightforward to show that (16) implies that (11) holds for  $\alpha_{k+1}$ . So, assume that (17) takes place. By (12), we have

$$|\phi'_{i_k}(\alpha_{k+1})| \leq -\bar{\sigma}\phi'_{\max}(0),$$

which, together with (17), implies that there exists  $i \in \mathcal{I}$  such that

$$\phi'_i(\alpha_{k+1}) > -\sigma\phi'_{\max}(0),$$

because  $0 < \bar{\sigma} < \sigma$ . Therefore, the test at Step 2 also holds for  $\alpha_{k+1}$ . Hence, by an inductive argument, if there exists a first  $k_0 \in \mathbb{N}$  for which (11) holds for  $\alpha_{k_0}$ , then (11) also holds for  $\alpha_k$  with  $k > k_0$ .

We prove the well definiteness by showing that Step 2 or Step 3 is necessarily executed if Algorithm 1 does not stop at Step 1. Assume that the algorithm does not stop in iteration  $k$ . If  $\alpha_k = \alpha_{\max}$ , then (11) holds for some  $i \in \mathcal{I}$  and Step 2 is executed. Indeed, if

$$\phi_i(\alpha_{\max}) \leq \phi_i(0) + \alpha_{\max}\rho\phi'_{\max}(0) \quad \text{for all } i \in \mathcal{I},$$

then

$$|\max\{\phi'_i(\alpha_{\max}) : i \in \mathcal{I}\}| > -\sigma\phi'_{\max}(0) \quad (18)$$

and

$$\phi'_i(\alpha_{\max}) \geq \sigma\phi'_{\max}(0) \quad \text{for some } i \in \mathcal{I}, \quad (19)$$

by Steps 1.1 and 1.2, respectively. By (18) and (19), there exists  $i \in \mathcal{I}$  such that

$$\phi'_i(\alpha_{\max}) > -\sigma\phi'_{\max}(0),$$

implying that (11) holds for  $\alpha_{\max}$ . Now assume  $\alpha_k < \alpha_{\max}$ . If (11) holds for  $\alpha_k$ , then Algorithm 1 executes Step 2. By the previous discussion, this is certainly the case if `brackt` = `true`. Note that if Step 2 is executed, then `brackt` is set equal to `true` and the algorithm goes directly to Step 4. Finally, if `brackt` = `false` and the test at Step 2 is not satisfied, then Algorithm 1 executes Step 3. Hence, one and only one step between Step 2 and Step 3 is executed, concluding the proof.  $\square$

Theorem 2 is our main result. We show that, in a finite number of iterations, Algorithm 1 finishes with  $\alpha = \alpha_{\max}$ , or finds a step-size satisfying the strong Wolfe conditions.

**Theorem 2.** *Assume that the inner solver has the finite termination property. Then, Algorithm 1 terminates the execution in a finite number of outer iterations satisfying one of the following conditions:*

1. sequence  $\{\alpha_k\}_{k \geq 0}$  is increasing and Algorithm 1 stops with  $\alpha = \alpha_k$  for which conditions (9) hold;
2. sequence  $\{\alpha_k\}_{k \geq 0}$  is increasing and Algorithm 1 stops with  $\alpha = \alpha_{\max}$ ;
3. there exists  $k_0 \in \mathbb{N}$  such that sequence  $\{\alpha_k\}_{k \geq k_0}$  is decreasing and Algorithm 1 stops with  $\alpha = \alpha_k$  for which conditions (9) hold.



*Proof.* We consider two possibilities:

- (i) the test at Step 2 is not satisfied in all iterations  $k \geq 0$ ;
- (ii) there exists a first  $k_0 \in \mathbb{N}$  for which the test at Step 2 holds.

Consider case (i). In this case, `brackt` is always equal to `false` and Algorithm 1 executes Step 3 in all iterations  $k \geq 0$ . Since

$$\alpha_{k+1} \in [\min\{\delta\alpha_k, \alpha_{\max}\}, \alpha_{\max}], \quad (20)$$

where  $\delta > 1$ , it follows that the generated sequence  $\{\alpha_k\}_{k \geq 0}$  is increasing. If, eventually for some  $k$ ,  $\alpha_k$  satisfies the stopping criterion at Step 1.1, then Algorithm 1 stops with  $\alpha = \alpha_k$  for which conditions (9) hold, satisfying the first possibility of the theorem. Otherwise, by (20), the upper bound  $\alpha_{\max}$  is used as a trial value in a finite number of iterations. We claim that Algorithm 1 stops at Step 1.2 with  $\alpha = \alpha_{\max}$ . Indeed, on the contrary, Step 2 is executed as shown in Theorem 1. Last case corresponds to the second condition of the theorem.

Now consider case (ii). As shown in Theorem 1, for  $k \geq k_0$ , either Algorithm 1 stops at  $\alpha_k$  or Step 2 is executed in iteration  $k$ . Since, by Step 2.3,  $\alpha_{k+1} \in (0, \alpha_k)$ , it turns out that  $\{\alpha_k\}_{k \geq k_0}$  is decreasing. Thus, the stopping criterion of Step 1.1 is the only one that can be satisfied. We prove the finite termination of the algorithm proceeding by contradiction. Assume that Algorithm 1 iterates infinitely. Thus,  $\{\alpha_k\}_{k \geq k_0}$  converges, because it is a bounded monotonic sequence. Let  $\alpha_* \geq 0$  be the limit of  $\{\alpha_k\}_{k \geq k_0}$ . We consider two cases:

- (a)  $\alpha_* = 0$ ;
- (b)  $\alpha_* > 0$ .

Assume case (a). By Lemma 1, for sufficiently large  $k$ , we have

$$\phi_i(\alpha_k) \leq \phi_i(0) + \alpha_k \rho \phi'_{\max}(0) \quad \text{for all } i \in \mathcal{I}.$$

Thus, since Step 2 is executed in iterations  $k \geq k_0$ , by (11) and the definition of  $i_k$ , we obtain

$$\phi'_{i_k}(\alpha_k) > -\sigma \phi'_{\max}(0), \quad (21)$$

for sufficiently large  $k$ . Since  $\mathcal{I}$  is a finite set of indexes, there are  $j \in \mathcal{I}$  and  $\mathbb{K}_1 \subset \mathbb{N}$ , such that  $i_k = j$  for all  $k \in \mathbb{K}_1$ . Hence, by (21) and (12), we have

$$\phi'_j(\alpha_k) > -\sigma \phi'_{\max}(0) \quad (22)$$

and

$$\left| \phi'_j(\alpha_{k+1}) \right| \leq -\bar{\sigma} \phi'_{\max}(0), \quad (23)$$

for all  $k \in \mathbb{K}_1$ . Taking limit on (22) and (23) for  $k \in \mathbb{K}_1$ , by the continuity of  $\phi_j$ , we obtain

$$-\sigma \phi'_{\max}(0) \leq \phi'_j(\alpha_*) \leq -\bar{\sigma} \phi'_{\max}(0),$$

concluding that  $\sigma \leq \bar{\sigma}$ . We are in contradiction, because  $\sigma > \bar{\sigma}$  by definition.

Now consider case (b). Since Algorithm 1 executes Step 2 for all  $k \geq k_0$ , by (11) and the definition of  $i_k$ , we obtain

$$\phi_{i_k}(\alpha_k) > \phi_{i_k}(0) + \alpha_k \rho \phi'_{\max}(0) \quad (24)$$

or

$$\phi'_{i_k}(\alpha_k) > -\sigma \phi'_{\max}(0). \quad (25)$$

Assume that (24) holds in infinitely many iterations  $k$ ,  $k \in \mathbb{K}_2 \subset \mathbb{N}$ . Since  $\mathcal{I}$  is a finite set of indexes, there are  $\ell \in \mathcal{I}$  and  $\mathbb{K}_3 \subset \mathbb{K}_2$ , such that  $i_k = \ell$  for all  $k \in \mathbb{K}_3$ . Hence, by (24) and (12), we have

$$\phi_\ell(\alpha_k) > \phi_\ell(0) + \alpha_k \rho \phi'_{\max}(0) \quad (26)$$

and

$$\phi_\ell(\alpha_{k+1}) \leq \phi_\ell(0) + \alpha_{k+1} \bar{\rho} \phi'_{\max}(0), \quad (27)$$

for all  $k \in \mathbb{K}_3$ . Taking limit on (26) and (27) for  $k \in \mathbb{K}_3$ , by the continuity of  $\phi_\ell$ , we obtain

$$\phi_\ell(0) + \alpha_* \rho \phi'_{\max}(0) \leq \phi_\ell(\alpha_*) \leq \phi_\ell(0) + \alpha_* \bar{\rho} \phi'_{\max}(0).$$

Thus  $\rho \geq \bar{\rho}$ , because  $\alpha_* > 0$  and  $\phi'_{\max}(0) < 0$ . Hence, we are in contradiction with the definition of  $\bar{\rho}$  and  $\rho$ . Finally, if (24) holds only in a finite number of iterations, then (25) is satisfied in infinitely many iterations. In this case, we can proceed as in case (a) to conclude that  $\sigma \leq \bar{\sigma}$ , generating a contradiction with the definition of  $\sigma$  and  $\bar{\sigma}$ . Therefore, the proof is complete.  $\square$

We can rule out the finite termination at  $\alpha_{\max}$  by assuming that

$$\phi_i(\alpha_{\max}) > \phi_i(0) + \alpha_{\max} \rho \phi'_{\max}(0) \quad \text{or} \quad \phi'_i(\alpha_{\max}) \geq \sigma \phi'_{\max}(0), \quad (28)$$

for some  $i \in \mathcal{I}$ . Under this assumption, Theorem 1 shows that the test at Step 2 holds for  $\alpha_{\max}$ . In this case, if  $\alpha_k = \alpha_{\max}$  for some  $k$ , then Theorem 2 implies that Algorithm 1 finds a step-size satisfying the strong Wolfe conditions. If functions  $\phi_i$  are bounded below for all  $i \in \mathcal{I}$ , then the first inequality of (28) holds, for example, for

$$\alpha_{\max} := \frac{\phi_{\max}(0) - \phi_{\min}}{-\rho \phi'_{\max}(0)},$$

where  $\phi_{\max}(0) := \max\{\phi_i(0) : i \in \mathcal{I}\}$  and  $\phi_{\min}$  is a strict lower bound of  $\phi_i$  for all  $i \in \mathcal{I}$ . Moreover, the second inequality of (28) trivially holds if  $\phi'_i(\alpha_{\max}) \geq 0$  for some  $i \in \mathcal{I}$ . A similar discussion of this matter for the scalar optimization case can be found in [37].

Algorithm 1 can be used to derive some theoretical results. In this case, at Step 2.3, the existence of  $\alpha_{k+1} \in (0, \alpha_{\max})$  satisfying (12), guaranteed by Lemma 2, is sufficient for our analysis, and we can rule out the discussion about the inner solver. Corollary 1 justifies our claim that, if the test at Step 2 holds for  $\alpha_k$ , then the interval  $(0, \alpha_k)$  brackets desired stepsizes.

**Corollary 1.** *Let  $\beta > 0$ . If, for some  $i \in \mathcal{I}$ ,*

$$\phi_i(\beta) > \phi_i(0) + \beta \rho \phi'_{\max}(0) \quad \text{or} \quad \phi'_i(\beta) > -\sigma \phi'_{\max}(0), \quad (29)$$

*then there exists an interval of stepsizes contained in  $(0, \beta)$  for which conditions (9) hold, i.e., the interval  $(0, \beta)$  brackets desired stepsizes.*

*Proof.* Set  $\alpha_{\max} > \beta$ , and  $\alpha_0 = \beta$ . Let  $\hat{\rho}$ ,  $\bar{\rho}$ ,  $\hat{\sigma}$ , and  $\bar{\sigma}$  be constants such that

$$0 < \rho < \hat{\rho} < \bar{\rho} < \bar{\sigma} < \hat{\sigma} < \sigma < 1.$$

Apply Algorithm 1 with the auxiliary constants  $0 < \hat{\rho} < \bar{\rho} < \bar{\sigma} < \hat{\sigma} < 1$ . By (29), we obtain

$$\phi_i(\beta) > \phi_i(0) + \beta \hat{\rho} \phi'_{\max}(0) \quad \text{or} \quad \phi'_i(\beta) > -\hat{\sigma} \phi'_{\max}(0),$$

because  $\rho < \hat{\rho}$ ,  $\hat{\sigma} < \sigma$ , and  $\phi'_{\max}(0) < 0$ . Thus, the test at Step 2 holds for  $\alpha_0$ . Then, by Theorem 2, the generated sequence  $\{\alpha_k\}_{k \geq 0}$  is decreasing and Algorithm 1 finds  $\alpha \in (0, \beta)$  satisfying conditions (9a) with  $\hat{\rho}$ , and (9b) with  $\hat{\sigma}$ . Hence,

$$\phi_i(\alpha) < \phi_i(0) + \alpha \rho \phi'_{\max}(0) \quad \text{for all } i \in \mathcal{I},$$

and

$$|\max\{\phi'_i(\alpha) : i \in \mathcal{I}\}| < -\sigma\phi'_{\max}(0),$$

because  $\rho < \hat{\rho}$ ,  $\hat{\sigma} < \sigma$ , and  $\phi'_{\max}(0) < 0$ . Therefore, by continuity arguments, there is a neighborhood of  $\alpha$  contained in  $(0, \beta)$  for which conditions (9) hold.  $\square$

In [34, Proposition 3.2], given  $d$  a  $K$ -descent direction for  $F$  at  $x$ , the authors showed that there is an interval of positive stepsizes satisfying the standard Wolfe conditions (4) and the strong Wolfe conditions (6), provided that  $F(x + \alpha d)$  is bounded below for  $\alpha > 0$ . For each  $i \in \mathcal{I}$ , this assumption implies that  $\phi_i$  is bounded below for  $\alpha > 0$ . Proposition 1 improves this result. We show that the presence of a single bounded below  $\phi_i$  is sufficient to guarantee the existence of stepsizes satisfying the Wolfe conditions. In the multiobjective optimization case, the benefits of this new result are clear: it is sufficient to assume the boundedness of only one objective, in contrast to the boundedness of all objectives required in [34].

**Proposition 1.** *Assume that  $F$  is of class  $\mathcal{C}^1$ ,  $C = \{w_1, \dots, w_p\}$ ,  $d$  is a  $K$ -descent direction for  $F$  at  $x$ , and there exists  $i \in \mathcal{I}$  such that  $\langle w_i, F(x + \alpha d) \rangle$  is bounded below for  $\alpha > 0$ . Then, there is an interval of positive stepsizes satisfying the standard Wolfe conditions (4) and the strong Wolfe conditions (6).*

*Proof.* Define  $l_i: \mathbb{R} \rightarrow \mathbb{R}$  by

$$l_i(\alpha) := \phi_i(0) + \alpha\rho\phi'_{\max}(0).$$

Since  $\phi'_{\max}(0) < 0$ , the line  $l_i(\alpha)$  is unbounded below for  $\alpha > 0$ . Thus, by the boundedness of  $\phi_i(\alpha)$ , there is a positive  $\beta$  such that  $\phi_i(\beta) > l_i(\beta)$ , i.e.,

$$\phi_i(\beta) > \phi_i(0) + \beta\rho\phi'_{\max}(0).$$

Hence, by Corollary 1, there exists an interval of stepsizes contained in  $(0, \beta)$  for which conditions (9) hold. Hence, the stepsizes belonging to this interval satisfy the strong Wolfe conditions (6) and, *a fortiori*, the standard Wolfe conditions (4).  $\square$

## 4 Implementation details

We implemented the Wolfe line search algorithm described in Section 3. In the following we discuss some practical aspects which support our implementation. The codes are written in Fortran 90, and are freely available at <https://lfprudente.mat.ufg.br/>.

### 4.1 Initial parameters

Parameters  $\rho$  and  $\sigma$  related to the Wolfe conditions must be set by the user. Usually  $\rho$  is chosen to be small, implying that condition (9a) requires a little more than a simple decrease in each function  $\phi_i$ ,  $i \in \mathcal{I}$ . A “traditional” recommended value is  $\rho = 10^{-4}$ . As we will see in Section 4.4, when we are dealing with multi-objective optimization problems involving quadratic functions,  $\rho$  has to be at least 0.5. As discussed in [34, Section 3], we say that the step-size  $\alpha$  is obtained by means of an exact line search whether

$$\max\{\phi'_i(\alpha) : i \in \mathcal{I}\} = 0.$$

Hence, by (9b), the parameter  $\sigma$  is related to the accuracy of the linear search. The trade-off between the computational cost and the accuracy of the search should be considered. Thus, the choice of  $\sigma$  is more tricky and may depend on the method used to solve problem (1). Values between 0.1 and 0.9 for  $\sigma$  should be appropriate. In [34], for non-linear conjugate gradient methods, the authors used  $\sigma = 0.1$ , performing a reasonably accurate line search. The

maximum allowed step-size  $\alpha_{\max}$  is also a customizable parameter. Its value should be large in practice, say  $\alpha_{\max} = 10^{10}$ .

Given  $\rho$  and  $\sigma$  such that  $0 < \rho < \sigma < 1$ , the auxiliary parameters  $\bar{\rho}$  and  $\bar{\sigma}$  used in the inner solver are set by

$$\bar{\rho} := \min\{1.1\rho, 0.75\rho + 0.25\sigma\},$$

and

$$\bar{\sigma} := \max\{0.9\sigma, 0.25\rho + 0.75\sigma\},$$

ensuring that  $0 < \rho < \bar{\rho} < \bar{\sigma} < \sigma < 1$ . In our implementation, we do not explicitly set the parameter  $\delta$ . However, the step-size updating scheme at Step 3 holds with  $\delta = 1.1$ , see Section 4.3 below.

The initial trial step-size  $\alpha_0$  is a user-supplied value. Its smart choice can directly benefit the performance of the line search algorithm. Based on the scalar minimization case, for a Newton-type method, the step  $\alpha_0 = 1$  should be used. On the other hand, for a gradient-type method, a wise choice is to use the vector extension of the Shanno and Phua [41] recommended choice, i.e., at iteration  $\ell$  of the underlying method (7), take

$$\alpha_0 = \alpha_{\ell-1} \frac{f(x^{\ell-1}, d^{\ell-1})}{f(x^\ell, d^\ell)}.$$

The special case where there exists a convex quadratic objective in a multiobjective problem will be discussed separately in Section 4.4.

## 4.2 Inner solver

In principle, any solver designed to find a step-size satisfying the strong Wolfe conditions for scalar optimization can be used to solve the subproblems at Step 2.3 of Algorithm 1. On the other hand, the finite termination of the inner solver is essential for the convergence results showed in the the last section. In our implementation we used the well-known algorithm of Moré and Thuente [37] as the inner solver. It is worth mentioning that the algorithm of Moré and Thuente is incorporated in some notable works, such as the L-BFGS-B code, see [47]. As we shall see below, this algorithm terminates in a finite number of steps, except for pathological cases.

Given a continuously differentiable function  $\phi: \mathbb{R} \rightarrow \mathbb{R}$  with  $\phi'(0) < 0$ , and constants  $\bar{\rho}$  and  $\bar{\sigma}$  satisfying  $0 < \bar{\rho} < \bar{\sigma} < 1$ , the algorithm of Moré and Thuente is designed to find  $\alpha > 0$  such that

$$\phi(\alpha) \leq \phi(0) + \bar{\rho}\alpha\phi'(0), \quad \text{and} \quad |\phi'(\alpha)| \leq -\bar{\sigma}\phi'(0). \quad (30)$$

The algorithm generates a sequence  $\{\alpha_\ell\}_{\ell \geq 0}$  belonging to the bounds

$$0 < \alpha_\ell \leq \alpha_{\max} \quad \text{for all} \quad \ell \geq 0,$$

where  $\alpha_{\max} > 0$  is a given algorithmic parameter. Assume that  $\alpha_{\max} > 0$  is such that

$$\phi(\alpha_{\max}) > \phi(0) + \alpha_{\max}\bar{\rho}\phi'(0) \quad \text{or} \quad \phi'(\alpha_{\max}) > -\bar{\sigma}\phi'(0). \quad (31)$$

Therefore, the interval  $(0, \alpha_{\max})$  contains a step-size  $\alpha$  satisfying (30), see [37, Theorem 2.1]. Consider the application of the algorithm of Moré and Thuente with an initial iterate  $\alpha_0 \in [0, \alpha_{\max}]$ . Theorem 2.3 of [37] implies that:

1. the algorithm terminates in a finite number of steps with an  $\alpha_\ell \in (0, \alpha_{\max})$  satisfying (30);
2. the algorithm iterates infinitely generating a sequence  $\{\alpha_\ell\}_{\ell \geq 0}$  that converges to some  $\alpha_* \in (0, \alpha_{\max})$  such that

$$\phi(\alpha_*) \leq \phi(0) + \bar{\rho}\alpha_*\phi'(0), \quad \text{and} \quad \phi'(\alpha_*) = \bar{\rho}\phi'(0).$$

Hence, conditions (30) holds at  $\alpha_*$ , because  $\bar{\rho} < \bar{\sigma}$  and  $\phi'(0) < 0$ . Moreover, if

$$\psi(\alpha) := \phi'(\alpha) - \bar{\rho}\phi'(0),$$

then  $\psi(\alpha_*) = 0$  and  $\psi$  changes sign an infinite number of times, in the sense that there exists a monotone sequence  $\{\beta_\ell\}$  that converges to  $\alpha_*$  and such that  $\psi(\beta_\ell)\psi(\beta_{\ell+1}) < 0$ .

Therefore, as claimed by the authors in [37], the algorithm produces a sequence of iterates that converge to a step-size satisfying (30) and, except for pathological cases, terminates in a finite number of steps.

Now consider the problem of finding  $\alpha > 0$  satisfying

$$\phi(\alpha) \leq \phi(0) + \bar{\rho}\alpha\Theta, \quad \text{and} \quad |\phi'(\alpha)| \leq -\bar{\sigma}\Theta, \quad (32)$$

where  $\Theta$  is a given constant such that  $\phi'(0) \leq \Theta < 0$ . We claim that the algorithm of Moré and Thuente can be easily adapted to solve this problem. Indeed, taking

$$\hat{\rho} := \bar{\rho} \frac{\Theta}{\phi'(0)} \quad \text{and} \quad \hat{\sigma} := \bar{\sigma} \frac{\Theta}{\phi'(0)},$$

we obtain  $0 < \hat{\rho} < \hat{\sigma} < 1$ , and (32) can be equivalently rewritten as (30) with constants  $\hat{\rho}$  and  $\hat{\sigma}$ . Furthermore, if  $\alpha_{\max} > 0$  is such that

$$\phi(\alpha_{\max}) > \phi(0) + \alpha_{\max}\bar{\rho}\Theta \quad \text{or} \quad \phi'(\alpha_{\max}) > -\bar{\sigma}\Theta, \quad (33)$$

then it is possible to show that  $(0, \alpha_{\max})$  contains a step-size  $\alpha$  satisfying (32), because (33) is equivalent to (31) with  $\hat{\rho}$  and  $\hat{\sigma}$ .

In the context of Algorithm 1, conditions (12) at Step 2.3 can be viewed as (32) with  $\phi \equiv \phi_{i_k}$  and  $\Theta = \phi'_{\max}(0)$ . Hence, the algorithm of Moré and Thuente can be used to solve the subproblems of Algorithm 1. Note that, at Step 2.3,  $i_k \in \mathcal{I}$  is an index such that (11) holds. Then, it follows that

$$\phi_{i_k}(\alpha_k) > \phi_{i_k}(0) + \alpha_k\bar{\rho}\phi'_{\max}(0) \quad \text{or} \quad \phi'_{i_k}(\alpha_k) > -\bar{\sigma}\phi'_{\max}(0),$$

because  $\rho < \bar{\rho} < \bar{\sigma} < \sigma$  and  $\phi'_{\max}(0) < 0$ . Thus, the algorithm of Moré and Thuente with an initial (inner) iterate belonging to  $[0, \alpha_k]$  finds, except for pathological cases, a step-size  $\alpha_{k+1} \in (0, \alpha_k)$  satisfying (12) in a finite number of (inner) steps. This allow us to conclude that it is to be expected that the algorithm of Moré and Thuente has the finite termination property, as assumed in Theorems 1 and 2.

### 4.3 Bracketing phase

In the bracketing phase, variable `brackt` is `false` and Algorithm 1 attempts to identify whether the interval  $(0, \alpha_k)$  contains desirable stepsizes by checking conditions (11). If conditions (11) are not fulfilled at iteration  $k$ , then Step 3 is executed by choosing

$$\alpha_{k+1} \in [\min\{\delta\alpha_k, \alpha_{\max}\}, \alpha_{\max}], \quad (34)$$

where  $\delta > 1$  is an algorithmic parameter.

In our implementation, Step 3 is executed by the inner algorithm of Moré and Thuente. Let  $\hat{i} \in \mathcal{I}$  be such that

$$\hat{i} = \arg \min\{\phi'_i(0) : i \in \mathcal{I}\}.$$

In the bracketing phase, if the test of Step 2 does not hold at iteration  $k$ , we call the algorithm of Moré and Thuente for  $\phi_{\hat{i}}$  with (inner) initial trial step-size equal to  $\alpha_k$ . Then, based on interpolations of known function and derivative values of  $\phi_{\hat{i}}$ , the new trial step-size  $\alpha_{k+1}$  is computed, satisfying (34) with  $\delta = 1.1$ , see [37]. Hence, as mentioned earlier, we do not explicitly set the algorithmic parameter  $\delta$ .

#### 4.4 Convex quadratic functions in multiobjective problems

In this section, we discuss the special case where there exists a convex quadratic objective in a multiobjective problem. Assume that  $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $F(x) = (F_1(x), \dots, F_m(x))$ ,  $K = \mathbb{R}_+^m$ ,  $C$  is the (ordered) canonical basis of  $\mathbb{R}^m$ ,  $e = [1, \dots, 1]^T \in \mathbb{R}^m$ ,  $\mathcal{I} = \{1, \dots, m\}$ ,  $\rho \leq 1/2$ , and there is  $i \in \mathcal{I}$  such that

$$F_i(x) = \frac{1}{2}x^T A_i x + b_i^T x + c_i,$$

where  $A_i \in \mathbb{R}^{n \times n}$  is a symmetric definite positive matrix,  $b \in \mathbb{R}^n$ , and  $c \in \mathbb{R}$ . Let  $d$  a  $K$ -descent direction for  $F$  at  $x$ . Note that, in this case, we have

$$\phi_i(\alpha) = F_i(x + \alpha d).$$

Its well known that the exact minimizer  $\alpha_i^*$  of  $\phi_i(\alpha)$  is given by

$$\alpha_i^* = -\frac{(A_i x + b_i)^T d}{d^T A_i d},$$

see, for example, [5, 38]. It is easy to see that

$$\phi_i(\alpha) \leq \phi_i(0) + \alpha \rho \phi'_{\max}(0) \quad \text{and} \quad \phi'_i(\alpha) \leq 0, \quad (35)$$

for all  $\alpha \in (0, \alpha_i^*]$ , because  $\phi'_i(0) \leq \phi'_{\max}(0) < 0$ . Define  $\mathcal{I}_Q \subset \mathcal{I}$  by

$$\mathcal{I}_Q := \{i \in \mathcal{I}: F_i \text{ is a convex quadratic function}\},$$

and  $\alpha_Q > 0$  by

$$\alpha_Q := \min\{\alpha_i^*: i \in \mathcal{I}_Q\}.$$

Now assume that  $\alpha_Q < \alpha_{\max}$ , and consider the application of Algorithm 1 with  $\alpha_0 = \alpha_Q$ . We claim that, in this case, Algorithm 1 needs to work only with the non-quadratic objectives. Indeed, since  $\alpha_0 \in (0, \alpha_i^*]$  for all  $i \in \mathcal{I}_Q$ , by (35), we obtain

$$\phi_i(\alpha_0) \leq \phi_i(0) + \alpha_0 \rho \phi'_{\max}(0) \quad \text{for all } i \in \mathcal{I}_Q,$$

and

$$0 = |\max\{\phi'_i(\alpha_0): i \in \mathcal{I}_Q\}| \leq -\sigma \phi'_{\max}(0).$$

Therefore, either  $\alpha_0$  satisfies conditions (9) and Algorithm 1 stops at Step 1.1 with  $\alpha = \alpha_0$  declaring convergence, or there is  $i_0 \in \mathcal{I} \setminus \mathcal{I}_Q$  such that (11) holds for  $\alpha_0$ . Note that the first case necessarily occurs when  $\mathcal{I}_Q = \mathcal{I}$ , i.e., all the objectives are convex quadratic functions. Consider the second case. Assuming that the inner solver has the finite termination property, by Theorems 1 and 2, the interval  $(0, \alpha_0)$  brackets a desirable step-size, Step 2 is always executed, and the generated sequence is decreasing. Since  $\alpha_k \in (0, \alpha_0)$  for all  $k \geq 1$ , by (35), the index  $i_k$  at Step 2.3 is such that  $i_k \in \mathcal{I} \setminus \mathcal{I}_Q$ , that is,  $\phi_{i_k}$  corresponds to a non-quadratic objective. Thus, the test at Step 2 should be performed only for the indexes in  $\mathcal{I} \setminus \mathcal{I}_Q$ . Moreover, by (12) and (35), it follows that, for all  $k \geq 0$ ,

$$\phi_i(\alpha_{k+1}) \leq \phi_i(0) + \alpha_{k+1} \rho \phi'_{\max}(0) \quad \text{for all } i \in \{i_k\} \cup \mathcal{I}_Q,$$

and

$$|\max\{\phi'_i(\alpha_{k+1}): i \in \{i_k\} \cup \mathcal{I}_Q\}| \leq -\sigma \phi'_{\max}(0),$$

because  $\rho < \bar{\rho} < \bar{\sigma} < \sigma$ . This implies that it is sufficient to test the convergence criterion at Step 1.1 only for the indexes in  $\mathcal{I} \setminus \mathcal{I}_Q$ . Hence, the quadratic functions can be neglected, as we claimed.

In our implementation, we require the user to inform which objectives are convex quadratics. If  $\mathcal{I}_Q = \mathcal{I}$ , then we take  $\alpha_0 = \alpha_Q$ , and Algorithm 1 stops returning  $\alpha = \alpha_0$ . Otherwise, we set  $\alpha_{\max} \leftarrow \alpha_Q$  and

$$\alpha_0 \leftarrow \min\{\alpha_0, \alpha_Q\},$$

where  $\alpha_0$  in the right hand side of the above expression is the user-supplied initial trial step-size. Then, Algorithm 1 is applied, neglecting the convex quadratic objectives.

## 5 Numerical experiments

We define the test problems by directly considering the functions  $\phi_i$  given by (8). For simplicity, we omit index  $i$  in the description of the functions. The first two functions given by

$$\phi(\alpha) = -\frac{\alpha}{\alpha^2 + \beta}, \quad (36)$$

where  $\beta > 0$ , and

$$\phi(\alpha) = (\alpha + \beta)^5 - 2(\alpha + \beta)^4, \quad (37)$$

where  $0 < \beta < 1.6$ , were proposed in [37]. These functions have unique minimizers at  $\alpha = \sqrt{\beta}$  and  $\alpha = 1.6 - \beta$ , respectively. The first function is concave to the right of the minimizer, while the region of concavity of the second function is to the left of the minimizer. Parameter  $\beta$  also controls the size of  $\phi'(0)$  because  $\phi'(0) = -1/\beta$  for function (36), and  $\phi'(0) = (5\beta - 8)\beta^3$  for function (37). In our experiments we set  $\beta = 0.16$  and  $\beta = 0.004$ , respectively. This choice for function (37) implies that condition (9b) is quite restrictive for any parameter  $\sigma > 1$ , because  $\phi'(0) \approx -5 \times 10^{-7}$ . Plots for these functions are in Figures 2(a) and 2(b).

The next function is defined by

$$\phi(\alpha) = \begin{cases} -\beta\alpha + \beta^2\alpha^2 & \text{if } \alpha < 0, \\ -\log(1 + \beta\alpha) & \text{if } 0 \leq \alpha \leq 1, \\ -\log(1 + \beta) - \frac{\beta}{1 + \beta}(\alpha - 1) + \frac{\beta^2}{(1 + \beta)^2}(\alpha - 1)^2 & \text{if } \alpha > 1, \end{cases} \quad (38)$$

where  $\beta > 0$ . This function is convex and has a unique minimizer at  $\alpha = 1.5 + 1/(2\beta)$ . Parameter  $\beta$  controls the size of  $\phi'(0) = -1/\beta$ . We take  $\beta = 100$ . This function appeared in [34], and its plot can be viewed in Figure 2(c).

The function defined by

$$\phi(\alpha) = \left(\sqrt{1 + \beta_1} - \beta_1\right) \sqrt{(1 - \alpha)^2 + \beta_2^2} + \left(\sqrt{1 + \beta_2} - \beta_2\right) \sqrt{\alpha^2 + \beta_1^2}, \quad (39)$$

where  $\beta_1, \beta_2 > 0$ , was suggested in [46]. This function is convex, but different combinations of parameters  $\beta_1$  and  $\beta_2$  lead to quite different characteristics. Plots of (39) with  $\beta_1 = 10^{-2}$  and  $\beta_2 = 10^{-3}$ , and with  $\beta_1 = 10^{-3}$  and  $\beta_2 = 10^{-2}$  are in Figures 2(d) and 2(e), respectively. Function (39) with these parameters was also used in the numerical tests of [37].

The function given by

$$\phi(\alpha) = 2 - 0.8e^{-\left(\frac{\alpha-0.6}{0.4}\right)^2} - e^{-\left(\frac{\alpha-0.2}{0.04}\right)} \quad (40)$$

is from [35]. This function has a global minimizer at  $\alpha \approx 0.2$  and a local minimizer at  $\alpha \approx 0.6$ . As highlighted in [35], the global minimizer of function (40) has a narrow attraction region when compared with the attraction region of its local minimizer. Plot of function (40) appears in Figure 2(f).

The next two functions are defined by

$$\phi(\alpha) = e^{-\beta\alpha}, \quad (41)$$

where  $\beta > 0$ , and

$$\phi(\alpha) = \begin{cases} 0 & \text{if } \alpha = 0, \\ -\alpha + 10^3\alpha^3 \sin(1/\alpha) & \text{if } \alpha \neq 0. \end{cases} \quad (42)$$

We have  $\phi'(0) = -1/\beta$  and  $\phi'(0) = -1$ , respectively. Although function (41) is bounded below, it has no minimizer. In contrast, function (42) has an infinite number of local minimizers. In

our numerical tests, we set  $\beta = 10$  for function (41). Figures 2(g) and 2(h) contain the plots of these functions.

Finally, we consider a simple convex quadratic function given by

$$\phi(\alpha) = \beta_1 \alpha^2 + \beta_2 \alpha, \quad (43)$$

where  $\beta_1 > 0$ , and  $\beta_2 < 0$ . In our experiments we take  $\beta_1 = 0.1$  and  $\beta_2 = -1$ . A plot of function (43) with these parameters appears in Figure 2(i).

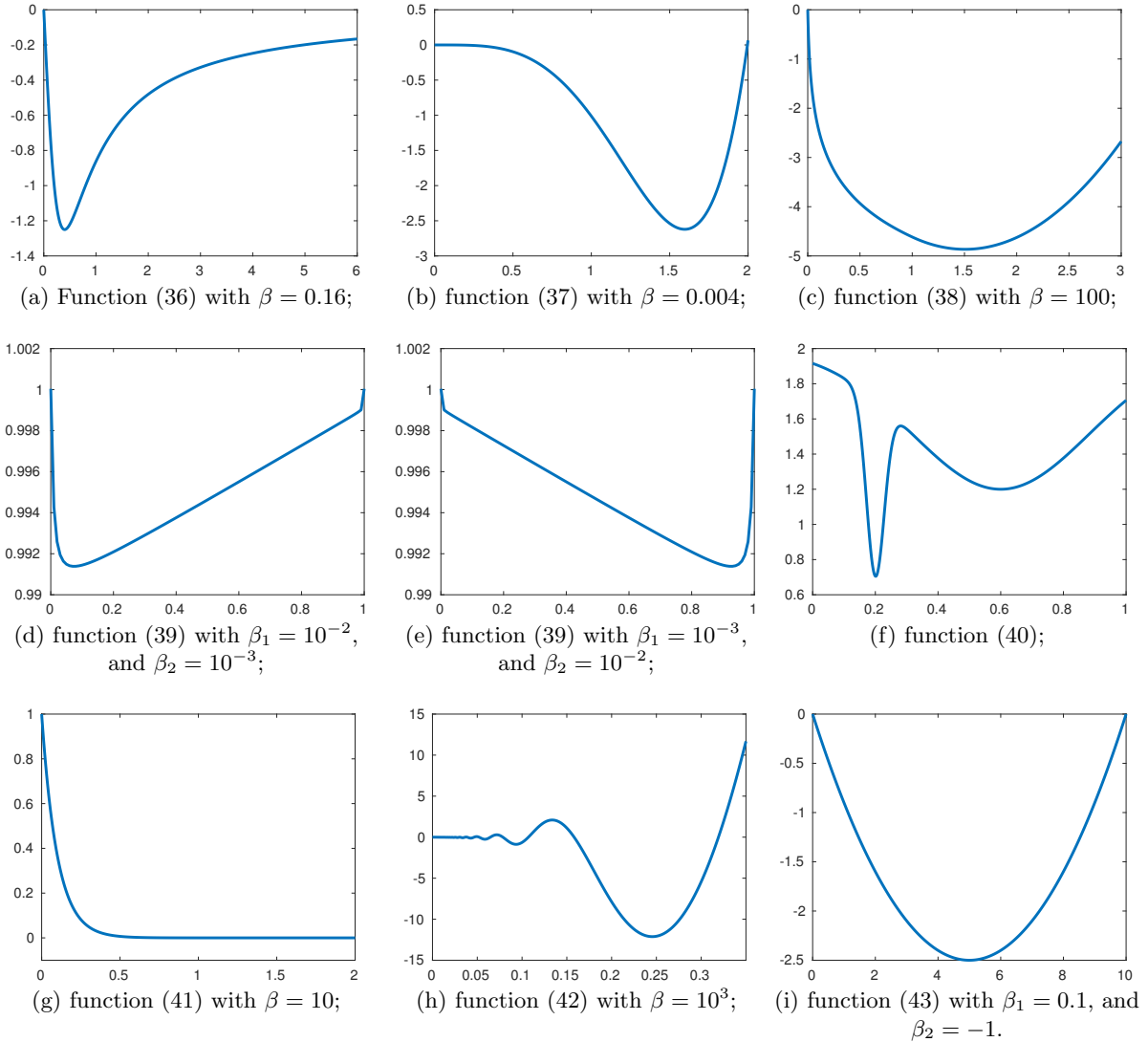


Figure 2: Plot of functions used in the numerical tests.

The set of test problems was defined combining different functions of Figure 2. Except for one problem, we defined  $\rho = 10^{-4}$  and  $\sigma = 0.1$ . This choice for  $\sigma$  guarantees a reasonably accurate line search. Parameter  $\alpha_{\max}$  was set to  $10^{10}$ . For each problem, we consider different values for the initial trial step-size, including particularly the remote values  $\alpha_0 = 10^{\pm 3}$ . The results are in Table 1. In the table, the first two columns identify the problem and the number  $m$  of functions involved. Columns “ $\alpha_0$ ” and “ $\alpha$ ” inform the initial trial step-size and the final iterate, respectively. “SC” informs the satisfied stopping criterion, where  $C$  denotes *convergence* meaning that the algorithm found a step-size satisfying conditions (9). Column “outiter” contains the number of outer iterations of Algorithm 1 given in the form: total number of outer iterations (number of iterations corresponding to the bracketing phase / number of iterations



corresponding to the selection phase). Recall that, in the bracketing phase, Algorithm 1 executes Step 3, whereas in the selection phase, it executes Step 2. Still in the table, “initer” is the total number of iterations used by the algorithm of Moré and Thuente, “nfev” is the number of evaluations of the functions, and “ngev” is the number of evaluations of the derivatives.

#	Problem	$\alpha_0$	$\alpha$	SC	outiter	initer	nfev	ngev
1	Functions in Figures 2(a) and 2(b) $m = 2$	$10^{-3}$	0.4	C	5(4/1)	11	22	21
		3	0.4	C	2(0/2)	17	21	19
		10	0.4	C	2(0/2)	17	21	19
		$10^3$	0.4	C	2(0/2)	20	24	22
2	Functions in Figures 2(c) and 2(i) $m = 2$	$10^{-3}$	1.50	C	6(5/1)	9	15	15
		0.5	1.48	C	2(1/1)	5	7	7
		2.5	1.49	C	1(0/1)	4	5	5
		$10^3$	1.53	C	1(0/1)	2	3	2
3	Functions in Figures 2(d) and 2(e) $m = 2$	$10^{-3}$	0.0737	C	4(3/1)	7	16	15
		0.4	0.0743	C	1(0/1)	5	8	7
		0.6	0.0743	C	1(0/1)	5	8	7
		$10^3$	0.0762	C	1(0/1)	8	10	9
4	Functions in Figures 2(a), (f) and 2(i) $m = 3$	$10^{-3}$	0.400	C	5(4/1)	8	19	18
		0.25	0.201	C	1(0/1)	8	11	11
		0.5	0.400	C	1(0/1)	4	7	6
		$10^3$	0.399	C	2(0/2)	9	13	11
5	Functions in Figures 2(b) and 2(g) $m = 2$	$10^{-3}$	1.60	C	6(5/1)	13	25	24
		0.3	1.60	C	3(2/1)	10	16	15
		1.5	1.60	C	2(1/1)	9	13	12
		$10^3$	1.60	C	1(0/1)	11	13	12
6	Functions in Figures 2(g) and 2(h) $m = 2$	$10^{-3}$	0.0236	C	3(2/1)	10	17	17
		0.2	0.245	C	2(1/1)	11	16	14
		10	0.00115	C	1(0/1)	10	13	11
		$10^3$	0.00195	C	1(0/1)	9	12	10
7	Functions in Figures 2(a)–2(h) $m = 8$	$10^{-3}$	0.0206	C	3(2/1)	13	44	44
		0.1	0.0164	C	2(0/2)	20	42	31
		10	0.0164	C	3(0/3)	31	50	38
		$10^3$	0.0164	C	3(0/3)	34	53	41

Table 1: Results of the numerical tests. We used  $\rho = 10^{-4}$  and  $\sigma = 0.1$  for Problems 1, 2, 4, 5 and 7, and  $\rho = 10^{-4}$  and  $\sigma = 10^{-3}$  for Problem 3.

As it can be seen in Table 1, Algorithm 1 found an step-size satisfying conditions (9) in all considered instances. Overall, regardless of the initial point used, few outer iterations and a moderate number of evaluations of functions/derivatives were needed to find an acceptable step-size. Typically, one or two iterations of the selection phase were sufficient for convergence. For each problem, the number of outer iterations was the largest when  $\alpha_0 = 10^{-3}$ . In these cases, Algorithm 1 performed some (cheap) iterations of the bracketing phase until it identified an interval containing an acceptable step-size and, consequently, entered in the selection phase. For the cases where  $\alpha_0 = 10^3$ , as it has to be expected, the algorithm identified that the interval  $(0, \alpha_0)$  brackets desirable stepsizes. This can be seen by noting that no bracketing iterations were performed.

We proceed by taking a detailed look at some test problems and the corresponding results. In Problem 1, except in the first choice for  $\alpha_0$  where the function in Figure 2(a) was used in the bracketing and selection phases, the algorithm identified that  $(0, \alpha_0)$  brackets acceptable stepsizes based on the function in Figure 2(b). In these cases, Algorithm 1 used this function to find  $\alpha_1 \approx 1.6$ . Since  $\alpha_1$  does not satisfy conditions (9), the second iteration of the selection phase was based on the function in Figure 2(a), resulting in the final iterate  $\alpha_2 \approx 0.4$ .

Problem 2 was proposed in [34] to show that it is not possible to replace  $\phi'_{\max}(0)$  by  $\phi'_i(0)$  in condition (9a) in the following sense: there is no step-size for this problem satisfying simultaneously conditions (9a) with  $\phi'_i(0)$  instead of  $\phi'_{\max}(0)$ , and (9b). Considering the Wolfe

conditions (9) in its own format, Problem 2 poses no challenges to the method proposed here.

For reasons of scale, we set  $\sigma = 10^{-3}$  for Problem 3. This value of parameter  $\sigma$  forces the set of acceptable stepsizes to be *close* to the minimizer of the function in Figure 2(d). For all initial trial stepsizes used, only one iteration of the selection phase based on this function was sufficient to found a desired step-size.

Problem 4 has two regions of acceptable stepsizes: around the minimizer of the function in Figure 2(a) and around the global minimizer of the function in Figure 2(f). As discussed in Section 4.4, the convex quadratic function in Figure 2(i) was neglected. For  $\alpha_0 = 10^{-3}$ , Algorithm 1 performed five bracketing iterations based on the function in Figure 2(a) until its minimizer was bracketed. For  $\alpha_0 = 0.25$  and  $\alpha_0 = 0.5$ , the algorithm identified that interval  $(0, \alpha_0)$  brackets the minimizer of the function in Figure 2(a) and the global minimizer of the function in Figure 2(f), respectively. As consequence, the final iterate for each of these choices was  $\alpha \approx 0.2$  and  $\alpha \approx 0.4$ , respectively. Two selection iterations were performed for  $\alpha_0 = 10^3$ . In this case, the initial trial step-size was modified as  $\alpha_0 \leftarrow \alpha_Q$ , where  $\alpha_Q = 5$  is the minimizer of the quadratic function in Figure 2(i). Using the updated interval  $(0, \alpha_0)$ , the first selection iteration based on the function in Figure 2(f) produced  $\alpha_1 \approx 0.59$ , which is around its local minimizer. The second selection iteration was based on the function in Figure 2(a), returning the acceptable step-size  $\alpha_2 \approx 0.4$ .

Problem 5 combines the function in Figure 2(b), which forces condition (9b) to be quite restrictive, with the function in Figure 2(g), that has no minimizer. Although there is a large region where  $\phi'(\alpha) \approx 0$  for the function in Figure 2(g), the acceptable stepsizes are *close* to the minimizer of the function in Figure 2(b). Typically, the algorithm uses the function in Figure 2 (g) in the bracketing phase until it generates a step-size to the right of the minimizer of the function in Figure 2(b). Then, using the function in Figure 2(b) in the selection phase, Algorithm 1 finds an acceptable step-size  $\alpha \approx 1.6$ .

Problem 6 combines two functions with quite different characteristics. While the function in Figure 2(g) has no minimizer, the function in Figure 2(h) has an infinite number of positive local minimizers *close* to zero. Table 1 shows that Algorithm 1 found different acceptable stepsizes for each  $\alpha_0$  considered.

Finally, the eight non-quadratic functions in Figure 2 compose Problem 7. All the final iterates were found based on the function in Figure 2(h). For  $\alpha_0 = 10^{-3}$ , the function in Figure 2(c) was used in the two iterations of the bracketing phase, returning  $\alpha_2 \approx 0.025$ . Then, the function in Figure 2(h) was used to find the final iterate  $\alpha_3 \approx 0.0206$ . In contrast, Algorithm 1 detects that  $(0, \alpha_0)$  brackets desirable stepsizes for the other choices of  $\alpha_0$ . For  $\alpha_0 = 0.1$ , the first selection iteration was based on the function in Figure 2(d). For  $\alpha_0 = 10$  and  $\alpha_0 = 10^3$ , the functions in Figures 2(b) and 2(d) were used in the first and second selection iterations, respectively. Then, in these last three cases, the acceptable step-size  $\alpha \approx 0.0164$  was found using the function in Figure 2(h).

## 6 Final remarks

In this work, an algorithm for finding a step-size satisfying the strong vector-valued Wolfe conditions has been proposed. In our implementation, we used the well-known algorithm of Moré and Thuente as the inner solver. As a consequence, except in pathological cases, the main algorithm stops in a finite number of iterations. First inequality of (28) is quite natural for practical problems. Hence, by Theorem 2, it is expected that the algorithm converges to an acceptable step-size.

Algorithm 1 is designed for vector-value problems (1) where the positive polar cone  $K^*$  given by (3) is finitely generated, i.e., when  $C$  is a finite set. Although this case covers most of the real applications, there are practical problems that require partial orders induced by cones with infinitely many extremal vectors. Applications involving such cones can be found, for

example, in [2] where problems of portfolio selection in security markets are studied. When  $F$  is  $K$ -convex and  $C$  is an infinite set, it is possible to show the existence of stepsizes satisfying the vector-value Wolfe conditions, see [34]. For the general case, even the existence of acceptable stepsizes is unknown. The study of the Wolfe conditions for infinitely generated cones as well as the extension of Algorithm 1 for this challenging problems are left as open problems for future research.

## References

- [1] AL-BAALI, M., AND FLETCHER, R. An efficient line search for nonlinear least squares. *Journal of Optimization Theory and Applications* 48, 3 (Mar 1986), 359–377.
- [2] ALIPRANTIS, C. D., FLORENZANO, M., DA ROCHA, V. F. M., AND TOURKY, R. Equilibrium analysis in financial markets with countably many securities. *Journal of Mathematical Economics* 40, 6 (2004), 683 – 699.
- [3] ANSARY, M. A., AND PANDA, G. A modified Quasi-Newton method for vector optimization problem. *Optimization* 64, 11 (2015), 2289–2306.
- [4] BELLO CRUZ, J. Y. A Subgradient Method for Vector Optimization Problems. *SIAM Journal on Optimization* 23, 4 (2013), 2169–2182.
- [5] BERTSEKAS, D. P. *Nonlinear programming*. Athena scientific Belmont, 1999.
- [6] BONNEL, H., IUSEM, A. N., AND SVAITER, B. F. Proximal methods in vector optimization. *SIAM Journal on Optimization* 15, 4 (2005), 953–970.
- [7] CARRIZO, G. A., LOTITO, P. A., AND MACIEL, M. C. Trust region globalization strategy for the nonconvex unconstrained multiobjective optimization problem. *Mathematical Programming* 159, 1 (Sep 2016), 339–369.
- [8] CENG, L. C., MORDUKHOVICH, B. S., AND YAO, J. C. Hybrid approximate proximal method with auxiliary variational inequality for vector optimization. *Journal of optimization theory and applications* 146, 2 (2010), 267–303.
- [9] CENG, L. C., AND YAO, J. C. Approximate proximal methods in vector optimization. *European Journal of Operational Research* 183, 1 (2007), 1–19.
- [10] CHUONG, T. D. Generalized proximal method for efficient solutions in vector optimization. *Numerical functional analysis and optimization* 32, 8 (2011), 843–857.
- [11] CHUONG, T. D. Newton-like methods for efficient solutions in vector optimization. *Computational Optimization and Applications* 54, 3 (2013), 495–516.
- [12] CHUONG, T. D., MORDUKHOVICH, B. S., AND YAO, J. C. Hybrid approximate proximal algorithms for efficient solutions in vector optimization. *Journal of Nonlinear and Convex Analysis* 12, 2 (8 2011), 257–285.
- [13] CHUONG, T. D., AND YAO, J. C. Steepest descent methods for critical points in vector optimization problems. *Applicable Analysis* 91, 10 (2012), 1811–1829.
- [14] DE, P., GHOSH, J. B., AND WELLS, C. E. On the minimization of completion time variance with a bicriteria extension. *Operations Research* 40, 6 (1992), 1148–1155.
- [15] DENNIS, J. E., AND SCHNABEL, R. B. *Numerical methods for unconstrained optimization and nonlinear equations*, vol. 16. Siam, 1996.

- [16] FLETCHER, R. *Practical methods of optimization*. John Wiley & Sons, 2013.
- [17] FLIEGE, J., GRAÑA DRUMMOND, L. M., AND SVAITER, B. F. Newton’s Method for Multiobjective Optimization. *SIAM Journal on Optimization* 20, 2 (2009), 602–626.
- [18] FLIEGE, J., AND SVAITER, B. F. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research* 51, 3 (2000), 479–494.
- [19] FLIEGE, J., AND VICENTE, L. N. Multicriteria approach to bilevel optimization. *Journal of optimization theory and applications* 131, 2 (2006), 209–225.
- [20] FLIEGE, J., AND WERNER, R. Robust multiobjective optimization & applications in portfolio optimization. *European Journal of Operational Research* 234, 2 (2014), 422 – 433.
- [21] FUKUDA, E. H., AND GRAÑA DRUMMOND, L. M. On the convergence of the projected gradient method for vector optimization. *Optimization* 60, 8-9 (2011), 1009–1021.
- [22] FUKUDA, E. H., AND GRAÑA DRUMMOND, L. M. Inexact projected gradient method for vector optimization. *Computational Optimization and Applications* 54, 3 (2013), 473–493.
- [23] GRAÑA DRUMMOND, L. M., AND IUSEM, A. N. A Projected Gradient Method for Vector Optimization Problems. *Computational Optimization and Applications* 28, 1 (2004), 5–29.
- [24] GRAÑA DRUMMOND, L. M., RAUPP, F. M. P., AND SVAITER, B. F. A quadratically convergent Newton method for vector optimization. *Optimization* 63, 5 (2014), 661–677.
- [25] GRAÑA DRUMMOND, L. M., AND SVAITER, B. F. A steepest descent method for vector optimization. *Journal of Computational and Applied Mathematics* 175, 2 (2005), 395 – 414.
- [26] GRAVEL, M., MARTEL, J. M., NADEAU, R., PRICE, W., AND TREMBLAY, R. A multi-criterion view of optimal resource allocation in job-shop production. *European journal of operational research* 61, 1-2 (1992), 230–244.
- [27] HAGER, W. W. A derivative-based bracketing scheme for univariate minimization and the conjugate gradient method. *Computers & Mathematics with Applications* 18, 9 (1989), 779–795.
- [28] HAGER, W. W., AND ZHANG, H. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization* 16, 1 (2005), 170–192.
- [29] HONG, T. S., CRAFT, D. L., CARLSSON, F., AND BORTFELD, T. R. Multicriteria optimization in intensity-modulated radiation therapy treatment planning for locally advanced cancer of the pancreatic head. *International Journal of Radiation Oncology\*Biophysics\*Physics* 72, 4 (2008), 1208–1214.
- [30] LEMARÉCHAL, C. A view of line-searches. In *Optimization and Optimal Control*. Springer, 1981, pp. 59–78.
- [31] LESCHINE, T. M., WALLENIUS, H., AND VERDINI, W. A. Interactive multiobjective analysis and assimilative capacity-based ocean disposal decisions. *European Journal of Operational Research* 56, 2 (1992), 278–289.
- [32] LU, F., AND CHEN, C.-R. Newton-like methods for solving vector optimization problems. *Applicable Analysis* 93, 8 (2014), 1567–1586.
- [33] LUC, D. T. Theory of vector optimization, Lectures Notes in economics and mathematical systems, vol. 319, 1989.

- [34] LUCAMBIO PÉREZ, L. R., AND PRUDENTE, L. F. Non-linear conjugate gradient methods for vector optimization. *submitted*.
- [35] MIGLIERINA, E., MOLHO, E., AND RECCHIONI, M. Box-constrained multi-objective optimization: A gradient-like method without a priori scalarization. *European Journal of Operational Research* 188, 3 (2008), 662–682.
- [36] MORÉ, J. J., AND SORENSEN, D. C. Newton’s method. Tech. rep., Argonne National Lab., IL (USA), 1982.
- [37] MORÉ, J. J., AND THUENTE, D. J. Line Search Algorithms with Guaranteed Sufficient Decrease. *ACM Trans. Math. Softw.* 20, 3 (Sept. 1994), 286–307.
- [38] NOCEDAL, J., AND WRIGHT, S. *Numerical optimization*. Springer Science & Business Media, 2006.
- [39] QU, S., GOH, M., AND CHAN, F. T. Quasi-Newton methods for solving multiobjective optimization. *Operations Research Letters* 39, 5 (2011), 397–399.
- [40] QU, S., GOH, M., AND LIANG, B. Trust region methods for solving multiobjective optimisation. *Optimization Methods and Software* 28, 4 (2013), 796–811.
- [41] SHANNO, D. F., AND PHUA, K. H. Remark on algorithm 500: Minimization of unconstrained multivariate functions. *ACM Transactions on Mathematical Software* 6, 4 (1980), 618–622.
- [42] TAVANA, M. A subjective assessment of alternative mission architectures for the human exploration of mars at nasa using multicriteria decision making. *Computers & Operations Research* 31, 7 (2004), 1147–1164.
- [43] TAVANA, M., SODENKAMP, M. A., AND SUHL, L. A soft multi-criteria decision analysis model with application to the European Union enlargement. *Annals of Operations Research* 181, 1 (2010), 393–421.
- [44] VILLACORTA, K. D., AND OLIVEIRA, P. R. An interior proximal method in vector optimization. *European Journal of Operational Research* 214, 3 (2011), 485–492.
- [45] WHITE, D. Epsilon-dominating solutions in mean-variance portfolio analysis. *European Journal of Operational Research* 105, 3 (1998), 457–466.
- [46] YANAI, H., OZAWA, M., AND KANEKO, S. Interpolation methods in one dimensional optimization. *Computing* 27, 2 (1981), 155–163.
- [47] ZHU, C., BYRD, R. H., LU, P., AND NOCEDAL, J. Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-scale Bound-constrained Optimization. *ACM Trans. Math. Softw.* 23, 4 (Dec. 1997), 550–560.